Assurance Activity Report for
Fortra's GoAnywhere Managed File Transfer
Version v6.8

Fortra's GoAnywhere Managed File Transfer v6.8 Security Target
Version 1.1

AAR Version 1.1

Protection Profile for Application Software, Version 1.3

Functional Package for Transport Layer Security (TLS), Version 1.1

Extended Package for Secure Shell (SSH), Version 1.0

Evaluated by:

2400 Research Blvd, Suite 395
Rockville, MD 20850

Prepared for:

National Information Assurance Partnership
Common Criteria Evaluation and Validation Scheme

The Developer of the TOE:
Fortra, LLC

The Author of the Security Target:
Acumen Security, LLC.

The TOE Evaluation was Sponsored by:
Fortra, LLC

Evaluation Personnel:
George Kumi
Sai Sandeep Yanamandra
Shaunak Shah

Common Criteria Version
Common Criteria Version 3.1 Revision 5

Common Evaluation Methodology Version
CEM Version 3.1 Revision 5

Revision History

VERSION	DATE	CHANGES
0.1	03/2021	Initial Draft
0.2	10/2021	Revised Draft
0.3	12/2022	Updated Draft
1.0	01/2023	Initial Release
1.1	03/2023	Minor update

Contents

`

1 TOE Overview

The Target of Evaluation (TOE) is the Fortra's GoAnywhere Managed File Transfer v6.8 (MFT). The TOE is a software application that provides secure file transfer services over HTTPS, TLS, and SSH. GoAnywhere MFT is a secure managed file transfer solution that streamlines the exchange of data between systems, employees, customers, and trading partners. It provides centralized control with extensive security settings, detailed audit trails, and helps process information from files into XML, CSV, and JSON databases.

2 Assurance Activities Identification

The TOE assurance requirements are taken directly from the Protection Profile for Application Software Version 1.3 which are derived from Common Criteria Version 3.1, Revision 5.

3 Test Equivalency Justification

All TOE platforms are tested by the lab.

The TOE implements twelve different cryptographic channels with trusted IT products:

- HTTPS/TLSv1.2 Web Server with or without TLS client authentication – Remote Administration
- TLSv1.2 client with or without TLS client authentication – Database server
- TLSv1.2 client with or without TLS client authentication – LDAP/AD server
- TLSv1.2 client without TLS client authentication – Mail server
- HTTPS/TLSv1.2 client with or without TLS client authentication – AS2, AS4, or WebDAV file servers
- SSHv2 client – SFTP or SCP file servers
- TLSv1.2 client – FTP/s file servers
- HTTPS/TLSv1.2 client – Amazon S3 or Azure Blob Storage servers
- HTTPS/TLSv1.2 client – REST, SOAP, or generic HTTPS servers
- HTTPS/TLSv1.2 server – AS2 or AS4 clients
- SSHv2 server – SFTP or SCP clients
- TLSv1.2 server – FTP/s clients

The TOE uses a single TLS implementation in the GoAnywhere MFT Bouncy Castle FIPS Java API cryptographic library version 1.0.2. The TOE maintains a global TLS configuration that is enforced for all TLS connections; so all TLS server connections are considered equivalent, and all TLS client connections are considered equivalent. Additionally, HTTPS/TLS is TLS transporting HTTP. There is no difference in the cryptography from stand-alone TLS.

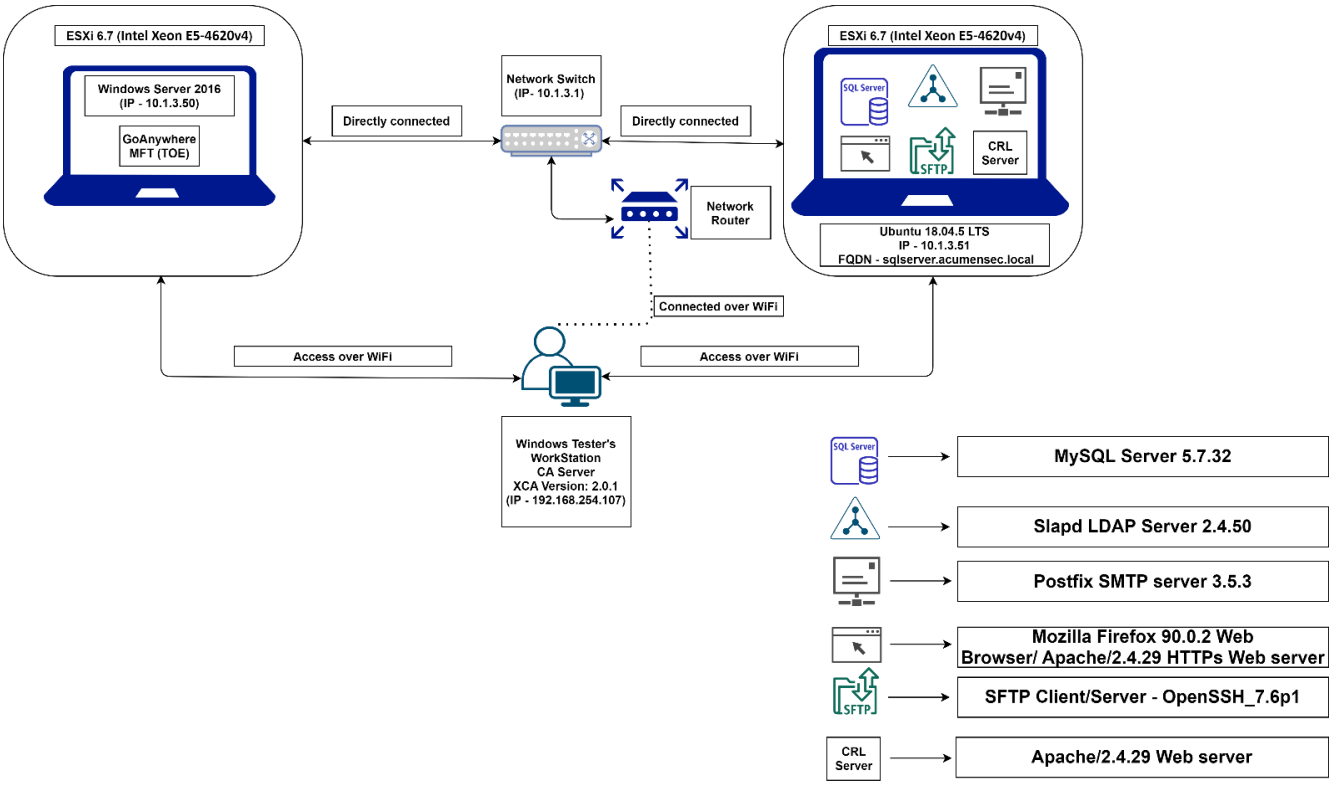
The following cryptographic channels were tested fully tested according to the evaluation activities:

- HTTPS/TLSv1.2 Web Server with TLS client authentication – Remote Administration
- TLSv1.2 client with TLS client authentication – Database server
- HTTPS/TLSv1.2 client with TLS client authentication– REST, SOAP, or generic HTTPS servers[1]
- SSHv2 client – SFTP or SCP file servers
- SSHv2 server – SFTP or SCP clients

[1] TLS client authentication tests only.

4 Test Bed Descriptions

Below is a visual representation of the components included in the test bed:



Location: Acumen Security, 2400 Research Boulevard Rockville Maryland, 20850

Packet Capture Location: Takes place on Remote Server (Ubuntu 18.04.5 LTS VM)

4.1 Configuration Information (Linux Platform)

The following provides configuration information about each device on the test network.

4.1.1 TOE Platform

- IP Address: 10.1.3.253
- Platform: CentOS 7

4.1.2 TOE Environment

- The CentOS 7 is the TOE's underlying platform
- TOE version: GoAnywhere MFT 6.8
- CPU: Intel Xeon E5-4620v4 (Broadwell)

4.1.3 Remote Server VM

- IP Address: 10.1.3.51
- OS: Ubuntu 18.04.5 LTS
- CPU: Intel Xeon E5-4620v4 (Broadwell)

The VM is configured as following servers:

- Remote DataBase server: MySQL Server 5.7.32
- Remote LDAP server: Slapd LDAP Server 2.4.50
- Remote SMTP server: Postfix SMTP server 3.5.3
- Remote HTTPS Web server: Mozilla Firefox 90.0.2 Web Browser/ Apache/2.4.29 HTTPs Web server
- Remote SFTP server/client: OpenSSH_7.6p1
- Remote CRL server: Apache/2.4.29 Web server

4.1.4 Tester's Workstation

- OS: Windows 10 Pro
- IP Address: 192.168.254.107

4.1.5 Switch

- Software Version: Build 12.2(53)SG1-IP-BASE
- IP Address: 10.1.3.1
- Switch MAC: 68:ef:bd:0a:61:7f

4.2 Configuration Information (Windows Platform)

The following provides configuration information about each device on the test network.

4.2.1 TOE Platform

- IP Address: 10.1.3.50
- Platform: Windows Server 2016

4.2.2 TOE Environment

- The Windows server 2016 is the TOE's underlying platform
- TOE version: GoAnywhere MFT 6.8
- CPU: Intel Xeon E5-4620v4 (Broadwell)

4.2.3 Remote Server VM

- IP Address: 10.1.3.51
- OS: Ubuntu 18.04.5 LTS
- CPU: Intel Xeon E5-4620v4 (Broadwell)

Configured on the following servers:

- Remote DataBase server: MySQL Server 5.7.32
- Remote LDAP server: Slapd LDAP Server 2.4.50
- Remote SMTP server: Postfix SMTP server 3.5.3
- Remote HTTPS Web server: Mozilla Firefox 90.0.2 Web Browser/ Apache/2.4.29 HTTPs Web server
- Remote SFTP server/client: OpenSSH_7.6p1
- Remote CRL server: Apache/2.4.29 Web server

4.2.4 Tester's Workstation

- OS: Windows 10 Pro

- IP Address: 192.168.254.107

4.2.5 Switch

- Software Version: Build 12.2(53)SG1-IP-BASE
- IP Address: 10.1.3.1
- Switch MAC: 68:ef:bd:0a:61:7f

4.3 Labgram

Name	OS	Version	Function	Protocols	IP address	MAC Address	Time	Tools (version)
GoAnywhere MFT on Esxi 6.7 server with Intel Xeon E5-4620v4 (Broadwell)	CentOS Linux	7	TOE and the Platform	TLS 1.2/ SSHv2	10.1.3.253	00:0C:29:9F:37:BD	Manually set and verified	Mozilla Firefox 90.0.2 Web Browser OpenSSH server(7.6p1) Nmap 7.70 Strace 4.24 GNU binutils 2.35 Canary detector(no version mentioned)
GoAnywhere MFT on Esxi 6.7 server with Intel Xeon E5-4620v4 (Broadwell)	Windows server	2016	TOE and the Platform	TLS 1.2/ SSHv2	10.1.3.50	00:0C:29:AF:C4:AD	Manually set and verified	Internet Explorer(11.576.14393.0) OpenSSH server(7.6p1) Nmap 7.70 HashMyFiles v2.41 Accesschk v6.14 Process Monitor v3.83 EMET 5.5 Process Hacker v2 VMMAP v3.26 Microsoft BinScope v2014 Notepad++ 8.1.2
Virtual Machine on Esxi 6.7 with Intel Xeon E5-4620v4 (Broadwell)	Ubuntu	18.04.5 LTS	Remote Database server. Remote Web client. Remote LDAP server. CRL server Remote .	TLSv1.2/SSHv2	10.1.3.51 FQDN: sqlserver.acumensec.local	00:0C:29:E9:C6:24	Manually set and verified	Mozilla Firefox 90.0.2 Web Browser Apache/2.4.29 Web server MySQL 5.7.32 server Wireshark 3.2.5 Zenmap 7.70 sftp (7.6) openssl 1.1.1 ssh-keygen (7.6) Filezilla client(3.47.2.1) OpenSSH server(7.6) Slapd OpenLDAP server(2.4.50) Postfix 3.5.3 Acumen tools:

			File Transfer Client. Remote File server. Remote Mail server					acumen-sshsfix (06/28/2020) acumen-sshs (10/19/2020) acumen-sshc (10/19/2020) acumen-tlsc-pkg (10/19/2020) acumen-tlsc-mysql [MySQL] (01/21/21) acumen-tlss-pkg (10/19/2020)
Tester's WorkStation	Windows	Windows 10	Test Workstation. CA Server.	SSHv2	192.168.254.107	74:E5:F9:D8:F5:0E	Manually set and verified	WinSCP (5.17) OpenSSH(7.6) XCA(2.0.1) Wireshark(3.2.5) EMET 5.5 WinMerge 2.16.6
Network Switch (Cisco WS-C2960L)	Build 12.2(53)SG1-IP-BASE	15.2(6)E1	Lab Switch	N/A	10.1.3.1	68:EF:BD:0A:61:7F	Manually set and verified	N/A
Network Lab Router (monsoon network)	IOS	15.2(6)E1	Lab Router	N/A	N/A	N/A	N/A	N/A

4.4 Testing Time and Location

All testing were conducted out at the Acumen Security offices located at 2400 Research Blvd Suite #395, Rockville, MD 20850. Testing and regression testing occurred from November 2020 through December 2022. The TOE was in a physically protected, access-controlled, designated test lab with no unattended entry/exit access points. At the start of each day the test bed was verified to ensure that it was not compromised.

Testing was conducted on GoAnywhere MFT 6.4.0, 6.8.0, 6.8.3 and the TOE was updated to the latest version GoAnywhere MFT 6.8.3.
Changes made from version 6.4.0 through 6.8.0:

- Updated client X.509 certificate SAN email address validation for Common Criteria.
- Added a new Strict Hostname Verification option which enforces all SSL/TLS connections to a remote server to properly validate the CN or SAN/DN values of the certificate regardless of the communication protocol.
- Added support for ECDSA host keys on the SFTP server.
- Fixed an issue where admin users were unable to login to the admin client using client authentication in FIPS 140-2 mode.
- Fixed issues with enforcing Strict Hostname Verification policy. This was introduced in 6.6.0.
- Added Allow Implicit Trust (SSH) setting to globally allow or deny implicit trust in SSH connections.

- Added support for 384-bit and 521-bit ECDSA key sizes in the SFTP/SCP/SSH client.

The above security-relevant changes were implemented to address the security functional requirements. Therefore, Regression Testing was performed on the following test cases to address the above changes.

- Entire TLSC test cases
- Entire TLSS test cases
- Entire X509 test cases
- SSHC test cases
    - FCS_SSHC_EXT.1.1 Test #1
    - FCS_SSHC_EXT.1.4 Test #2
    - FCS_SSHC_EXT.1.7 Test #1
    - FCS_SSHC_EXT.1.8 Test #1
    - FCS_SSHC_EXT.1.8 Test #2
- SSHS test cases
    - FCS_SSHS_EXT.1.1 Test #1
    - FCS_SSHS_EXT.1.4 Test #2
    - FCS_SSHS_EXT.1.5 Test #2
    - FCS_SSHS_EXT.1.7 Test #1

Changes made from version 6.8.0 to 6.8.3.

- Upgraded the Apache Batik libraries version from 1.10 to 1.14.
- Upgraded the Jasper Reports libraries version from 6.7.0 to 6.16.0.
- Upgraded the XML Graphics library version from 2.2 to 2.6.
- Enhanced the efficiency of the process that applies file/folder permissions to avoid unnecessary lookups.
- Improved threading usage within Agent transfers.
- Enhanced SFTP transfer speeds by setting the default SFTP buffer size to 1MB.
- Updated the shutdown process to reduce the amount of time it takes for a node to leave the cluster.

Remote testing was performed on a strictly minimal list of the following SFRs as a spot check to ensure the above changes did not affect any security requirement or functionality tested as a part of the evaluation. The fixes listed above do not change or affect the results collected during testing. The CCTL believes there is no need to redo any of the tests since the outcome would not change.

- Entire Filesystem test cases
- Static Analysis test cases
    - FCS_STO_EXT.1.1 Test #1
    - FPT_API_EXT.1.1 Test #1
- Operation test cases
    - FMT_SMF.1.1 Test #1
    - FPT_TUD_EXT.1.1 Test #1
    - FPT_TUD_EXT.1.2 Test #1
- Network test cases
    - FTP_DIT_EXT.1.1 Test #1
    - FTP_DIT_EXT.1.1 Test #2

- FTP_DIT_EXT.1.1 Test #3

Changes made from version 6.8.3 through 6.8.5.

- Upgraded Apache Tomcat from version 9.0.41 to 9.0.52.
- Fixed an issue for FPT_TUD_EXT.2.2 Test #1 where an executable file was left behind after uninstalling the application. The latest version 6.8.5 meets the requirement where no files other than configuration, output, audit log files are left behind.

Remote testing was performed on a strictly minimal list of the following SFRs as a spot check to ensure the above changes did not affect any security requirement or functionality tested as a part of the evaluation.

- Filesystem test cases
  - FPT_TUD_EXT.2.2 Test #1
  - FPT_LIB_EXT.1.1 Test #1
  - FPT_IDV_EXT.1.1 Test #1
- Operation test cases
  - FPT_TUD_EXT.1.1 Test #1
  - FPT_TUD_EXT.1.2 Test #1
- Static Analysis
  - FPT_TUD_EXT.2.1 Test #1

Changes made from version 6.8.5 through 6.8.7:

- Added configuration for amount of entropy required by GoAnywhere MFT from the Operating System for cryptographic random number generation.
- Upgraded the Postgres JDBC Driver from version 42.2.14 to version 42.3.3.
- Updated Spring Framework from 5.2.9 to 5.3.18.
- Updated Apache Log4j from version 2.16.0 to 2.17.1.
- Fixed an issue where the toolbar in Secure Folders would not properly refresh when navigating to virtual folders with different permissions. Breadcrumb navigation was also updated to refresh the toolbar.

The following improvements (made from 6.8.5 to 6.8.7) were not related to security features or evaluated features and with no impact on any SFRs.
Based on the above finding, sample regression testing was performed on 6.8.7 for the following test cases:

- FPT_IDV_EXT.1.1 Test #1,
- FPT_TUD_EXT.1.1 Test #1,
- FPT_TUD_EXT.1.2 Test #1,
- FPT_TUD_EXT.2.1 Test #1,
- FPT_TUD_EXT.2.2 Test #1,
- FCS_TLSC_EXT.1.2 Test #2,
- FCS_TLSC_EXT.5.1 Test #1,
- FIA_X509_EXT.1.1 Test #2,
- FIA_X509_EXT.1.1 Test #8a,
- FPT_LIB_EXT.1.1 Test #1

5 Detailed Test Cases (TSS and Guidance Activities)

5.1 TSS and Guidance Activities (Cryptographic Support)

5.1.1 FCS_CKM_EXT.1

5.1.1.1 FCS_CKM_EXT.1 TSS 1

Objective	The evaluator shall inspect the application and its developer documentation to determine if the application needs asymmetric key generation services. If not, the evaluator shall verify the generate no asymmetric cryptographic keys selection is present in the ST. Otherwise, the evaluation activities shall be performed as stated in the selection-based requirements.
Evaluator Findings	The evaluator examined the SFR section in the Security Target and determined that the application needs asymmetric key generation services. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.2 FCS_CKM.1(1)

5.1.2.1 FCS_CKM.1(1) TSS 1

Objective	The evaluator shall ensure that the TSS identifies the key sizes supported by the TOE. If the ST specifies more than one scheme, the evaluator shall examine the TSS to verify that it identifies the usage for each scheme.
Evaluator Findings	The evaluator examined the section titled **'TOE Summary Specification'** in the Security Target to verify that the TSS identifies the key sizes supported by the TOE, and if more than one scheme is specified, the usage for each scheme. Upon investigation, the evaluator found that the TSS states that the TSF generates RSA 2048-bit, 3072-bit, and 4096-bit keys. These keys are used for digital signature and key agreement services in TLS or digital signature services in SSH. The TSF generates ECDSA P-256, P-384, and P-521 keys. P-256 and P-384 keys are used for key agreement services in TLS and digital signature service in SSH. P-256, P-384, and P-521 keys are used for digital signature services in TLS and key agreement services in SSH. The TSF also generates Diffie-Hellman Group 14 keys for key agreement services in SSH. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.2.2 FCS_CKM.1(1) TSS 2

Objective	If the application invokes platform-provided functionality for asymmetric key generation, then the evaluator shall examine the TSS to verify that it describes how the key generation functionality is invoked.

Evaluator Findings	The evaluator examined the section titled **'TOE Summary Specification'** in the Security Target to verify that the TSS describes how the key generation functionality is implemented. Upon investigation, the evaluator found that the TSS states that the TOE DRBG is seeded with at least 256 bits of entropy from the platform DRBG, CryptGenRandom on Windows and /dev/random on CentOS. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.2.3 FCS_CKM.1(1) Guidance 1

Objective	The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected key generation scheme(s) and key size(s) for all uses defined in this PP.
Evaluator Findings	The evaluator examined the section titled **'Enabling FIPS 140-2 mode'** in the AGD to verify that it instructs the administrator how to configure the TOE to use the selected key generation scheme(s) and key size(s) for all uses defined in this PP. Upon investigation, the evaluator found that the AGD states that the GoAnywhere provides a FIPS 140-2 Compliance Mode and when enabled, it only permits the use of FIPS 140-2 compliant ciphers for encrypting the data. The Administrator must ensure that the FIPS 140-2 mode is always enabled to implement only evaluated encryption algorithms as other cryptographic engines were not evaluated or tested during the Common Criteria evaluation of the product. The AGD also states that the GoAnywhere MFT automatically generates and performs RSAES-PKCS1-v1_5 key transport with RSA 2048-bit, 3072-bit, and 4096-bit keys that are used for digital signature and key agreement services in TLS when the application is set to FIPS 140-2 mode. The TOE automatically generates and performs Elliptic Curve Diffie-Hellman with curves ECDSA P-256, P-384 which are used for key agreement services in TLS in FIPS 140-2 Compliance mode. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.2.4 FCS_CKM.1(1) Test/CAVP 1

Objective	If the application implements asymmetric key generation, then the following test activities shall be carried out.
Evaluator Findings	CAVP DRBG Certs: #C1876. For additional details, please refer to the CAVP Table 1. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.3 FCS_CKM.1(2)

5.1.3.1 FCS_CKM.1(2) TSS 1

Objective	The evaluator shall review the TSS to determine that it describes how the functionality described by FCS_RBG_EXT.1 is invoked.
Evaluator Findings	The evaluator examined the section titled **'TOE Summary Specification'** in the Security Target to verify that the TSS describes how the functionality described by FCS_RBG_EXT.1 is invoked. Upon investigation, the evaluator found that the TSS states that the TOE DRBG is seeded with at least 256 bits of entropy from the platform DRBG, CryptGenRandom on Windows and /dev/random on CentOS. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.3.2 FCS_CKM.1(2) TSS 2

Objective	If the application is relying on random bit generation from the host platform, the evaluator shall verify the TSS includes the name/manufacturer of the external RBG and describes the function call and parameters used when calling the external DRBG function. If different external RBGs are used for different platforms, the evaluator shall verify the TSS identifies each RBG for each platform. Also, the evaluator shall verify the TSS includes a short description of the vendor's assumption for the amount of entropy seeding the external DRBG. The evaluator uses the description of the RBG functionality in FCS_RBG_EXT or documentation available for the operational environment to determine that the key size being requested is identical to the key size and mode to be used for the encryption/decryption of the user data.
Evaluator Findings	The evaluator examined the SFR in the Security Target and determined that the TOE is not relying on random bit generation from the host platform. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.4 FCS_CKM.2

5.1.4.1 FCS_CKM.2 TSS 1

Objective	The evaluator shall ensure that the supported key establishment schemes correspond to the key generation schemes identified in FCS_CKM.1.1.
Evaluator Findings	The evaluator examined the section titled **'TOE Summary Specification'** in the Security Target to verify that the supported key establishment schemes correspond to the key generation schemes identified in FCS_CKM.1.1. Upon investigation, the evaluator found that the TSS states that the TSF performs RSAES-PKCS1-v1_5 key transport with 2048-bit, 3072-bit, and 4096-bit keys in TLS. The TSF performs Elliptic Curve Diffie-Hellman with curves P-256 and P-384 in TLS and SSH. SSH also supports P-521. The TSF performs Diffie-Helman with Group 14 in SSH.

	All of these supported key establishment schemes and sizes correspond to key generation schemes identified in FCS_CKM.1.1. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.4.2 FCS_CKM.2 TSS 2

Objective	If the ST specifies more than one scheme, the evaluator shall examine the TSS to verify that it identifies the usage for each scheme.
Evaluator Findings	The evaluator examined the section titled **'TOE Summary Specification'** in the Security Target to verify that the TSS identifies the usage for each scheme. Upon investigation, the evaluator found that the TSS states that: • TSF performs RSAES-PKCS1-v1_5 key transport in TLS. • The TSF performs Elliptic Curve Diffie-Hellman in TLS and SSH. • The TSF performs Diffie-Helman with Group 14 in SSH. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.4.3 FCS_CKM.2 Guidance 1

Objective	The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected key establishment scheme(s).
Evaluator Findings	The evaluator examined the section titled **'SFTP Server Configuration'** under the subsection **'a. Enabled Public Key Algorithms'** in the AGD to verify that it instructs the administrator how to configure the TOE to use the selected key establishment scheme(s). Upon investigation, the evaluator found that the AGD states that the TOE can be configured to select the following Key Exchange Algorithms that are compliant as per the evaluation: diffie-hellman-group14-sha1, ecdh-sha2-nistp256, ecdh-sha2-nistp384, and ecdh-sha2-nistp521. The instructions are available in the AGD. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.4.4 FCS_CKM.2 Test/CAVP 1

Objective	The evaluator shall verify the implementation of the key establishment schemes supported by the TOE using the applicable tests below.
Evaluator Findings	CAVP Certs: C1876 For additional details, please refer to the CAVP Table 1. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.5 FCS_COP.1(1)

5.1.5.1 FCS_COP.1(1) Guidance 1

Objective	The evaluator checks the AGD documents to determine that any configuration that is required to be done to configure the functionality for the required modes and key sizes is present.
Evaluator Findings	The evaluator examined the section titled **'Enabling FIPS 140-2 mode'** in the AGD to verify that any configuration that is required to be done to configure the functionality for the required modes and key sizes is present. Upon investigation, the evaluator found that the AGD states that the GoAnywhere provides a FIPS 140-2 Compliance Mode and when enabled, it only permits the use of FIPS 140-2 compliant ciphers for encrypting the data. The Administrator must ensure that the FIPS 140-2 mode is always enabled to implement only evaluated encryption algorithms as other cryptographic engines were not evaluated or tested during the Common Criteria evaluation of the product. The AGD also states that the GoAnywhere MFT automatically generates and performs RSAES-PKCS1-v1_5 key transport with RSA 2048-bit, 3072-bit, and 4096-bit keys that are used for digital signature and key agreement services in TLS when the application is set to FIPS 140-2 mode. The TOE automatically generates and performs Elliptic Curve Diffie-Hellman with curves ECDSA P-256, P-384 which are used for key agreement services in TLS in FIPS 140-2 Compliance mode. There is no specific configuration required for generating these keys apart from selecting the common criteria compliant cipher-suites that must be supported by the application as seen in SSL/TLS Configuration. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.5.2 FCS_COP.1(1) Test/CAVP 1

Objective	The evaluator shall perform all of the [tests in the PP] for each algorithm implemented by the TSF and used to satisfy the requirements of this PP. **TD0598 has been applied.**
Evaluator Findings	CAVP Certs: #C1876 For additional details, please refer to the CAVP Table 1. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.6 FCS_COP.1(2)

5.1.6.1 FCS_COP.1(2) TSS 1

Objective	The evaluator shall check that the association of the hash function with other application cryptographic functions (for example, the digital signature verification function) is documented in the TSS.

Evaluator Findings	The evaluator examined the section titled **'TOE Summary Specification'** in the Security Target to verify that the TSS documents the association of the hash function with other application cryptographic functions. Upon investigation, the evaluator found that the TSS states that the TSF provides cryptographic hashing services using SHA-1, SHA-256, SHA-384, and SHA-512 as specified in FIPS Pub 180-4 "Secure Hash Standard." The TSF uses all of the hashes for RSA SigGen & SigVer, ECDSA SigGen & SigVer, HMAC, and SSH KDF; with the exception SHA-1 which is not used with ECDSA SigGen & SigVer. Further validation details can be viewed in Table 4 of the ST. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.6.2 FCS_COP.1(2) Test/CAVP 1

Objective	The evaluator shall perform all of the following tests for each hash algorithm implemented by the TSF and used to satisfy the requirements of this PP.
Evaluator Findings	CAVP Certs: #C1876 For additional details, please refer to the CAVP Table 1. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.7 FCS_COP.1(3)

5.1.7.1 FCS_COP.1(3) Test/CAVP 1

Objective	Algorithm Tests
Evaluator Findings	CAVP Certs: #C1876 For additional details, please refer to the CAVP Table 1. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.8 FCS_COP.1(4)

5.1.8.1 FCS_COP.1(4) Test/CAVP 1

Objective	For each of the supported parameter sets, the evaluator shall compose 15 sets of test data. Each set shall consist of a key and message data. The evaluator shall have the TSF generate HMAC tags for these sets of test data. The resulting MAC tags shall be compared to the result of generating HMAC tags with the same key and IV using a known-good implementation.
Evaluator Findings	CAVP Certs: #C1876 For additional details, please refer to the CAVP Table 1. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.9 FCS_HTTPS_EXT.1/Client

5.1.9.1 FCS_HTTPS_EXT.1.1/Client TSS 1

Objective	The evaluator shall examine the TSS and determine that enough detail is provided to explain how the implementation complies with RFC 2818. **TD0668 has been applied.**
Evaluator Findings	The evaluator examined the section titled **'TOE Summary Specification'** in the Security Target to verify that the TSS provides enough detail to explain how the implementation complies with RFC 2818. Upon investigation, the evaluator found that the TSS states that the TSF implements HTTPS as specified in RFC 2818 using TLSv1.2 as the secure transport protocol. TLSv1.2 is specified in FCS_TLSC_EXT.1 and FCS_TLSS_EXT.1. The TSF does not establish the connection (client or server) if the peer certificate is invalid. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.10 FCS_HTTPS_EXT.1/Server

5.1.10.1 FCS_HTTPS_EXT.1.1/Server TSS 1

Objective	The evaluator shall examine the TSS and determine that enough detail is provided to explain how the implementation complies with RFC 2818. **TD0668 has been applied.**
Evaluator Findings	The evaluator examined the section titled **'TOE Summary Specification'** in the Security Target to verify that the TSS provides enough detail to explain how the implementation complies with RFC 2818. Upon investigation, the evaluator found that the TSS states that when acting as an HTTPS server, the TSF supports TLS client authentication for remote administration, HTTPS file access, and AS2 connections. If the client does not present a certificate or the certificate is not authorized, the TSF falls back to application layer authentication (e.g. username and password). The TSF does not establish the connection (client or server) if the peer certificate is invalid. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.11 FCS_RBG_EXT.1

5.1.11.1 FCS_RBG_EXT.1 TSS 1

Objective	If use no DRBG functionality is selected, the evaluator shall inspect the application and its developer documentation and verify that the application needs no random bit generation services.

Evaluator Findings	The evaluator examined the SFR section in the Security Target and determined that "use no DRBG functionality" is not selected. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.11.2 FCS_RBG_EXT.1 TSS 2

Objective	If implement DRBG functionality is selected, the evaluator shall ensure that additional FCS_RBG_EXT.2 elements are included in the ST.
Evaluator Findings	The evaluator examined the section titled **'TOE Summary Specification'** in the Security Target to verify that additional FCS_RBG_EXT.2 elements are included in the ST. Upon investigation, the evaluator found that the TSS includes additional information for FCS_RBG_EXT.2. The TSS states that the TOE provides random bit generation services using an SP 800-90A CTR_DRBG using AES-256. The TOE DRBG is seeded with at least 256 bits of entropy from the platform DRBG, CryptGenRandom on Windows and /dev/random on CentOS. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.11.3 FCS_RBG_EXT.1 TSS 3

Objective	If invoke platform-provided DRBG functionality is selected, the evaluator performs the following activities. The evaluator shall examine the TSS to confirm that it identifies all functions (as described by the SFRs included in the ST) that obtain random numbers from the platform RBG. The evaluator shall determine that for each of these functions, the TSS states which platform interface (API) is used to obtain the random numbers. The evaluator shall confirm that each of these interfaces corresponds to the acceptable interfaces listed for each platform below [in the PP].
Evaluator Findings	The evaluator examined the SFR section in the Security Target and determined that "invoke platform-provided DRBG functionality" is not selected. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.12 FCS_RBG_EXT.2

Objective	Documentation shall be produced - and the evaluator shall perform the activities - in FCS_CKM.1.1(1) accordance with Appendix D - Entropy Documentation and Assessment and the Clarification to the Entropy Documentation and Assessment Annex.
Evaluator Findings	An Entropy Assessment Report for entropy source details was provided. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.13 FCS_STO_EXT.1

5.1.13.1 FCS_STO_EXT.1 TSS 1

Objective	The evaluator shall check the TSS to ensure that it lists all persistent credentials (secret keys, PKI private keys, or passwords) needed to meet the requirements in the ST. For each of these items, the evaluator shall confirm that the TSS lists for what purpose it is used, and how it is stored.
Evaluator Findings	The evaluator examined the section titled **'TOE Summary Specification'** in the Security Target to verify that the TSS lists all persistent credentials (secret keys, PKI private keys, or passwords) needed to meet the requirements in the ST. For each of these items, the evaluator shall confirm that the TSS lists for what purpose it is used, and how it is stored. Upon investigation, the evaluator found that the TSS states that the TSF stores the database password in non-volatile memory encrypted using AES-GCM-256 implemented as specified in FCS_COP.1(1). The TSF does not store any other credentials in non-volatile memory. All other sensitive data (e.g. user passwords, private keys) is stored in the remote database. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.14 FCS_TLS_EXT.1

5.1.14.1 FCS_TLS_EXT.1 Guidance 1

Objective	The evaluator shall ensure that the selections indicated in the ST are consistent with selections in the dependent components.
Evaluator Findings	The evaluator examined the section titled **'Operational Environment'** in the AGD to verify that the selections indicated in the ST are consistent with selections in the dependent components. Upon investigation, the evaluator found that the AGD provides a list of required IT Environment Components when the TOE is configured in its evaluated configuration. The TOE supports (sometimes optionally) secure connectivity with several other IT environment devices which are also described in the AGD. These are consistent with findings in the ST. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.15 FCS_TLSC_EXT.1

5.1.15.1 FCS_TLSC_EXT.1.1 TSS 1

Objective	The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that the cipher suites supported are specified. The evaluator shall check the TSS to ensure that the cipher suites specified include those listed for this component.

Evaluator Findings	The evaluator examined the section titled **'TOE Summary Specification'** in the Security Target to verify that the TSS specifies the cipher suites supported and that the cipher suites specified include those listed for this component. Upon investigation, the evaluator found that the TSS states that the TSF is a TLSv1.2 client supporting the following ciphersuites: • TLS_RSA_WITH_AES_128_CBC_SHA • TLS_RSA_WITH_AES_256_CBC_SHA • TLS_RSA_WITH_AES_128_CBC_SHA256 • TLS_RSA_WITH_AES_256_CBC_SHA256 • TLS_RSA_WITH_AES_128_GCM_SHA256 • TLS_RSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 • TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 • TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.15.2 FCS_TLSC_EXT.1.1 Guidance 1

Objective	The evaluator shall also check the operational guidance to ensure that it contains instructions on configuring the product so that TLS conforms to the description in the TSS.
Evaluator Findings	The evaluator examined the section titled **'SSL/TLS Configuration'** in the AGD to verify that it contains instructions on configuring the product so that TLS conforms to the description in the TSS. Upon investigation, the evaluator found that the AGD states that the GoAnywhere MFT Application automatically configures references identifiers based on the FQDN or IP address configured used to specify the TLS server. The Application compares the FQDN, or IP address configured to specify the TLS server against the identifiers in the presented X.509 certificates for certificate validation. The AGD also states that after the FIPS 140-2 Compliance Mode is enabled on the TOE, the Administrator must manually configure the allowed Protocols (TLSv1.2) and allowed cipher suites in the Algorithms section. The configured Algorithm settings are applicable for FTPS, HTTPS, AS2, SSL, SMTPS, GoFast, and Agent communications, as well as database connections and User authentication over SSL (LDAPS). The following steps must be followed to configure the Algorithms. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.15.3 FCS_TLSC_EXT.1.2 TSS 1

Objective	The evaluator shall ensure that the TSS describes the client's method of establishing all reference identifiers from the application-configured reference identifier, including which types of reference identifiers are supported (e.g. Common Name, DNS Name, URI Name, Service Name, or other application-specific Subject Alternative Names) and whether IP addresses and wildcards are supported.
Evaluator Findings	The evaluator examined the section titled **'TOE Summary Specification'** in the Security Target to verify that the TSS describes the client's method of establishing all reference identifiers from the application-configured reference identifier, including which types of reference identifiers are supported and whether IP addresses and wildcards are supported. Upon investigation, the evaluator found that the TSS states that the TSF automatically configures references identifiers based on the FQDN or IP address configured used to specify the TLS server. When an FQDN has been configured, the TOE establishes reference identifiers of DNS-ID and CN-ID. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.15.4 FCS_TLSC_EXT.1.2 TSS 2

Objective	The evaluator shall ensure that this description identifies whether and the manner in which certificate pinning is supported or used by the product.
Evaluator Findings	The evaluator examined the section titled **'TOE Summary Specification'** in the Security Target to verify that the TSS identifies whether and the manner in which certificate pinning is supported or used by the product. Upon investigation, the evaluator found that the TSS states that the TSF does not support certificate pinning. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.15.5 FCS_TLSC_EXT.1.2 Guidance 1

Objective	The evaluator shall verify that the AGD guidance includes instructions for setting the reference identifier to be used for the purposes of certificate validation in TLS.
Evaluator Findings	The evaluator examined the section titled **'Database Configuration'** in the AGD to verify that it includes instructions for setting the reference identifier to be used for the purposes of certificate validation in TLS. Upon investigation, the evaluator found that the AGD states that for the TLS connections with the database server, the reference identifier is configured in the JDBC URL. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.15.6 FCS_TLSC_EXT.1.3 TSS 1

Objective	If the selection for authorizing override of invalid certificates is made, then the evaluator shall ensure that the TSS includes a description of how and when user or administrator authorization is obtained. The evaluator shall also ensure that the TSS describes any mechanism for storing such authorizations, such that future presentation of such otherwise-invalid certificates permits establishment of a trusted channel without user or administrator action.
Evaluator Findings	The evaluator examined the SFR in the Security Target and determined that "except when override is authorized" is not selected. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.16 FCS_TLSC_EXT.2

5.1.16.1 FCS_TLSC_EXT.2 TSS 1

Objective	The evaluator shall ensure that the TSS description required per FIA_X509_EXT.2.1 describes the use of client-side certificates for TLS mutual authentication. The evaluator shall also ensure that the TSS describes any factors beyond configuration that are necessary in order for the client to engage in mutual authentication using X.509v3 certificates.
Evaluator Findings	The evaluator examined the section titled **'TOE Summary Specification'** in the Security Target to verify that the TSS includes the use of client-side certificates for TLS mutual authentication and any factors beyond configuration that are necessary in order for the client to engage in mutual authentication using X.509v3 certificates. Upon investigation, the evaluator found that the TSS states that the TSF supports TLS mutual authentication for FTP/s, AS2, and HTTPS connections. The TSF is capable of presenting a certificate in response to a CertificateRequest message from the server. The TSF only sends a certificate if the administrator has configured a client certificate for the specific server in question. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.16.2 FCS_TLSC_EXT.2 Guidance 1

Objective	The evaluator shall ensure that the AGD guidance includes any instructions necessary to configure the TOE to perform mutual authentication. The evaluator also shall verify that the AGD guidance required per FIA_X509_EXT.2.1 includes instructions for configuring the client-side certificates for TLS mutual authentication.
Evaluator Findings	The evaluator examined the section titled **'Admin Server Configuration'** in the AGD to verify that it includes any instructions necessary to configure the TOE to perform mutual authentication and for configuring the client-side certificates. Upon investigation, the evaluator found that the AGD provides information for both non mutual connections and for mutual authentication. When client authentication is set

	to mutual authentication it is required that the SSL connection will not connect or authenticate a User unless a valid certificate is available. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.17 FCS_TLSC_EXT.5

5.1.17.1 FCS_TLSC_EXT.5 TSS 1

Objective	The evaluator shall verify that TSS describes the Supported Groups Extension.
Evaluator Findings	The evaluator examined the section titled **'TOE Summary Specification'** in the Security Target to verify that the TSS describes the Supported Groups Extension. Upon investigation, the evaluator found that the TSS states that the TSF presents the Supported Groups (formerly named Supported Elliptic Curves) extension in the ClientHello message with support for groups secp256r1 and secp384r1. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.18 FCS_TLSS_EXT.1

5.1.18.1 FCS_TLSS_EXT.1.1 TSS 1

Objective	The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that the cipher suites supported are specified. The evaluator shall check the TSS to ensure that the cipher suites specified include those listed for this component.
Evaluator Findings	The evaluator examined the section titled **'TOE Summary Specification'** in the Security Target to verify that the TSS specifies the cipher suites supported and that the cipher suites specified include those listed for this component. Upon investigation, the evaluator found that the TSS states that The TSF is a TLSv1.2 server supporting the following ciphersuites: • TLS_RSA_WITH_AES_128_CBC_SHA • TLS_RSA_WITH_AES_256_CBC_SHA • TLS_RSA_WITH_AES_128_CBC_SHA256 • TLS_RSA_WITH_AES_256_CBC_SHA256 • TLS_RSA_WITH_AES_128_GCM_SHA256 • TLS_RSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384

	• TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 • TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.18.2 FCS_TLSS_EXT.1.1 Guidance 1

Objective	The evaluator shall also check the operational guidance to ensure that it contains instructions on configuring the TOE so that TLS conforms to the description in the TSS.
Evaluator Findings	The evaluator examined the section titled **'Enabling FIPS 140-2 mode'** and **'SSL/TLS configuration'** in the AGD to verify that it contains instructions on configuring the TOE so that TLS conforms to the description in the TSS. Upon investigation, the evaluator found that the AGD states the GoAnywhere MFT uses TLS encryption to communicate securely with various IT environment devices. The algorithms settings are useful in setting the TLS protocol versions and cipher suites to be used globally by the application. The configuration instructions are provided in the AGD. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.18.3 FCS_TLSS_EXT.1.2 TSS 1

Objective	The evaluator shall verify that the TSS contains a description of the denial of old SSL and TLS versions consistent relative to selections in FCS_TLSS_EXT.1.2.
Evaluator Findings	The evaluator examined the section titled '**TOE Summary Specification'** in the Security Target to verify that the TSS describes the denial of old SSL and TLS versions consistent relative to selections in FCS_TLSS_EXT.1.2. Upon investigation, the evaluator found that the TSS states that the TSF sends a Fatal protocol_version alert message if it receives a ClientHello requesting SSLv2.0, SSLv3.0, TLSv1.0, or TLSv1.1. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.18.4 FCS_TLSS_EXT.1.2 Guidance 1

Objective	The evaluator shall verify that the AGD guidance includes any configuration necessary to meet this requirement.
Evaluator Findings	The evaluator examined the section titled **'SSL/TLS configuration'** in the AGD to verify that it includes any configuration necessary to meet this requirement. Upon investigation, the evaluator found that the AGD states that the GoAnywhere MFT Application automatically configures references identifiers based on the FQDN or IP address configured used to specify the TLS server. The Application compares the

	FQDN, or IP address configured to specify the TLS server against the identifiers in the presented X.509 certificates for certificate validation. The AGD also states that after the FIPS 140-2 Compliance Mode is enabled on the TOE, the Administrator must manually configure the allowed Protocols (TLSv1.2) and allowed cipher suites in the Algorithms section. The configured Algorithm settings are applicable for FTPS, HTTPS, AS2, SSL, SMTPS, GoFast, and Agent communications, as well as database connections and User authentication over SSL (LDAPS). The following steps must be followed to configure the Algorithms. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.18.5 FCS_TLSS_EXT.1.3 TSS 1

Objective	The evaluator shall verify that the TSS describes the key agreement parameters of the server's Key Exchange message.
Evaluator Findings	The evaluator examined the section titled '**TOE Summary Specification'** in the Security Target to verify that the TSS describes the key agreement parameters of the server's Key Exchange message. Upon investigation, the evaluator found that the TSS states that the TSF supports RSA key agreement with RSA key sizes of 2048 bits, 3072 bits, or 4096 bits (i.e. the size of the RSA key in the TLS server certificate). The TSF supports ECDHE key agreement using curves P-256 and P-384. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.18.6 FCS_TLSS_EXT.1.3 Guidance 1

Objective	The evaluator shall verify that any configuration guidance necessary to meet the requirement must be contained in the AGD guidance.
Evaluator Findings	The evaluator examined the section titled **'SSL/TLS configuration'** in the AGD to verify that it contains any configuration guidance necessary to meet the requirement. Upon investigation, the evaluator found that after the FIPS 140-2 Compliance Mode is enabled on the TOE, the Administrator must manually configure the allowed Protocols (TLSv1.2) and allowed cipher suites in the Algorithms section. The configured Algorithm settings are applicable for FTPS, HTTPS, AS2, SSL, SMTPS, GoFast, and Agent communications, as well as database connections and User authentication over SSL (LDAPS). The following steps must be followed to configure the Algorithms. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.19 FCS_TLSS_EXT.2

5.1.19.1 FCS_TLSS_EXT.2.2 TSS 1

Objective	The evaluator shall ensure that the TSS description required per FIA_X509_EXT.2.1 includes the use of client-side certificates for TLS mutual authentication.

Evaluator Findings	The evaluator examined the section titled '**TOE Summary Specification'** in the Security Target to verify that the TSS describes the use of client-side certificates for TLS mutual authentication. Upon investigation, the evaluator found that the TSS states that the TSF can be configured to request a client certificate when establishing TLS connections for remote administration, HTTPS file access, AS2, and FTP/s. When the TSF is configured to authenticate TLS clients using certificates, the TSF sends a Certificate Request message in the TLS handshake. The TSF requires the DN in the presented certificate to match a DN authorized for the services the client is connecting to. The TSF then verifies the certificate matches the certificate pinned to the user's account using a SHA-1 hash and validates the certificate chain as described in FIA_X509_EXT.1. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.19.2 FCS_TLSS_EXT.2.2 Guidance 1

Objective	The evaluator shall verify that the AGD guidance required per FIA_X509_EXT.2.1 includes instructions for configuring the client-side certificates for TLS mutual authentication. The evaluator shall ensure that the AGD guidance includes instructions for configuring the server to require mutual authentication of clients using these certificates.
Evaluator Findings	The evaluator examined the section titled **'Admin Server Configuration'** in the AGD to verify that it includes instructions for configuring the client-side certificates for TLS mutual authentication and for configuring the server to require mutual authentication of clients using these certificates. Upon investigation, the evaluator found that the AGD states that the GoAnywhere MFT is configured as an Administration server in order to allow the users access the Application UI over TLSv1.2. The Admin Server page provides the ability to modify the GoAnywhere Admin Server connection and listener. To manage the Admin Server, log in as an Admin User with the Product Administrator role. If your user account is assigned to a custom Admin User Role, your ability to view, modify, or execute actions on this page are based on the permissions specified for that role. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.19.3 FCS_TLSS_EXT.2.3 TSS 1

Objective	If the product implements mutual authentication, the evaluator shall verify that the TSS describes how the DN and SAN in the certificate is compared to the expected identifier.
Evaluator Findings	The evaluator examined the section titled **'TOE Summary Specification'** in the Security Target to verify that the TSS describes how the DN and SAN in the certificate is compared to the expected identifier. Upon investigation, the evaluator found that the TSS states when the TSF is configured to authenticate TLS clients using certificates, the TSF sends a Certificate Request message in the TLS handshake. The TSF requires the

	DN in the presented certificate to match a DN that has been configured as authorized for the services the client is connecting to. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.19.4 FCS_TLSS_EXT.2.3 Guidance 1

Objective	If the DN is not compared automatically to the domain name, IP address, username, or email address, the evaluator shall ensure that the AGD guidance includes configuration of the expected identifier or the directory server for the connection.
Evaluator Findings	The evaluator examined the section titled **'Management Functions'** in the AGD to verify that, if the DN is not compared automatically to the domain name, IP address, username, or email address, the guidance includes configuration of the expected identifier or the directory server for the connection. Upon investigation, the evaluator found that the AGD states that the GoAnywhere MFT allows identifiers to be configured for admin user authentication from a remote web browser. Configuration instructions are available in the AGD. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.2 TSS and Guidance Activities (User Data Protection)

5.2.1 FDP_DAR_EXT.1

5.2.1.1 FDP_DAR_EXT.1 TSS 1

Objective	The evaluator shall examine the TSS to ensure that it describes the sensitive data processed by the application. The evaluator shall then ensure that the [test] activities [in the PP] cover all of the sensitive data identified in the TSS.
Evaluator Findings	The evaluator examined the section titled **'TOE Summary Specification'** in the Security Target to verify that the TSS describes the sensitive data processed by the application. Upon investigation, the evaluator found that the TSS states that the TSF relies on platform provided functionality to encrypt data at rest. The TSF requires the user to enable full drive encryption. The evaluator also examined the section titled **'Other Assumptions'** in the AGD to verify that the stated sensitive data is covered by the results obtained from the test assurance activities. Upon investigation, the evaluator found that the sensitive data information is covered by the test results. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.2.2 FDP_DEC_EXT.1

5.2.2.1 FDP_DEC_EXT.1.1 Guidance 1

Objective	The evaluator shall perform the platform-specific [test] actions [in the PP] and inspect user documentation to determine the application's access to hardware resources. The evaluator shall ensure that this is consistent with the selections indicated.
Evaluator Findings	The evaluator examined the section titled **'TOE access to platform resources'** in the AGD to verify that the stated hardware access is consistent with the SFR selections. Upon investigation, the evaluator found that the AGD states that network connectivity is the only hardware platform resource accessed by the TOE. The evaluator also examined the section titled **'TOE access to platform resources'** in the AGD to verify that the stated hardware access is consistent with the results obtained from the test assurance activities. Upon investigation, the evaluator found that the hardware access information is consistent. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.2.2.2 FDP_DEC_EXT.1.1 Guidance 2

Objective	The evaluator shall review documentation provided by the application developer and for each resource which it accesses, identify the justification as to why access is required.
Evaluator Findings	The evaluator examined the section titled **'TOE access to platform resources'** in the AGD to identify, for each resource which it accesses, the justification as to why access is required. Upon investigation, the evaluator found that the AGD states that network connectivity is the only hardware platform resource accessed by the TOE. The TOE communicates with several IT environment for the following reasons. Database server for Remote database for storing settings and private keys, Authentication server for Remote authentication server for user authentication, File server for Remote file server for storing user files, Mail server for supporting SMTP to send notifications and Remote browser for Remote administration and User file access. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.2.2.3 FDP_DEC_EXT.1.2 Guidance 1

Objective	The evaluator shall perform the platform-specific [test] actions [in the PP] and inspect user documentation to determine the application's access to sensitive information repositories. The evaluator shall ensure that this is consistent with the selections indicated.
Evaluator Findings	The evaluator examined the section titled **'TOE access to platform resources'** in the AGD to verify that the stated hardware access is consistent with the SFR selections. Upon investigation, the evaluator found that the AGD states that network connectivity is the only hardware platform resource accessed by the TOE. This is consistent with

	selections made in the SFR in which the application shall restrict its access to network connectivity and system logs. The evaluator also examined the section titled **'TOE access to platform resources'** in the AGD to verify that the stated repository access is consistent with the results obtained from the test assurance activities. Upon investigation, the evaluator found that the repository access information is consistent. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.2.2.4 FDP_DEC_EXT.1.2 Guidance 2

Objective	The evaluator shall review documentation provided by the application developer and for each sensitive information repository which it accesses, identify the justification as to why access is required.
Evaluator Findings	The evaluator examined the section titled **'TOE access to platform resources'** in the AGD to identify, for each sensitive information repository which it accesses, the justification as to why access is required. Upon investigation, the evaluator found that the AGD states that system logs are the only sensitive information repository accessed by the TOE. The TOE accesses system logs (i.e., Windows Event log) for the purpose of writing events to the logs. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.3 TSS and Guidance Activities (Identification and Authentication)

5.3.1 FIA_X509_EXT.1

5.3.1.1 FIA_X509_EXT.1.1 TSS 1

Objective	The evaluator shall ensure the TSS describes where the check of validity of the certificates takes place. The evaluator ensures the TSS also provides a description of the certificate path validation algorithm. **TD0668 has been applied.**
Evaluator Findings	The evaluator examined the section titled **'TOE Summary Specification'** in the Security Target to verify that the TSS describes where the check of validity of the certificates takes place and the certificate path validation algorithm. Upon investigation, the evaluator found that the TSS states that the TSF performs certificate validation during the TLS handshake when the non-TOE entity presents a certificate to the TSF. The X.509 certificates are validated using the certificate path validation algorithm defined in RFC 5280, which can be summarized as follows: • the public key algorithm and parameters are checked • the current date/time is checked against the validity period

	• revocation status is checked • issuer name of X matches the subject name of X+1 • name constraints are checked • policy OIDs are checked • policy constraints are checked; issuers are ensured to have CA signing bits • path length is checked Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.3.2 FIA_X509_EXT.2

5.3.2.1 FIA_X509_EXT.2.1 TSS 1

Objective	The evaluator shall check the TSS to ensure that it describes how the TOE chooses which certificates to use, and any necessary instructions in the administrative guidance for configuring the operating environment so that the TOE can use the certificates.
Evaluator Findings	The evaluator examined the section titled **'TOE Summary Specification'** in the Security Target to verify that the TSS describes how the TOE chooses which certificates to use, and any necessary instructions in the administrative guidance for configuring the operating environment so that the TOE can use the certificates. Upon investigation, the evaluator found that the TSS states that during the TLS handshake, the TSF uses the certificate presented by the TLS client or server to authenticate the remote endpoint of the connection. If the TSF cannot establish a connection to fetch a CRL, the TSF considers the certificate invalid and rejects the certificate. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.3.2.2 FIA_X509_EXT.2.1 TSS 2

Objective	The evaluator shall examine the TSS to confirm that it describes the behavior of the TOE when a connection cannot be established during the validity check of a certificate used in establishing a trusted channel. The evaluator shall verify that any distinctions between trusted channels are described. If the requirement that the administrator is able to specify the default action, then the evaluator shall ensure that the operational guidance contains instructions on how this configuration action is performed.
Evaluator Findings	The evaluator examined the section titled **'TOE Summary Specification'** in the Security Target to verify that the TSS describes the behavior of the TOE when a connection cannot be established during the validity check of a certificate used in establishing a trusted channel. Upon investigation, the evaluator found that the TSS states that if the

	TSF cannot establish a connection to fetch a CRL, the TSF considers the certificate invalid and rejects the certificate. The evaluator also examined the section titled **'TOE Summary Specification'** in the Security Target to verify that the TSS describes any distinctions between trusted channels. Upon investigation, the evaluator found that the TSS notes no distinctions between trusted channels. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.3.2.3 FIA_X509_EXT.2.1 Guidance 1

Objective	The evaluator shall check the administrative guidance to ensure that it describes configuring the operating environment so that the TOE can use the certificates.
Evaluator Findings	The evaluator examined the section titled **'Operating System'** and **'Configuring Various System Users'** in the AGD to verify that it describes configuring the operating environment so that the TOE can use the certificates. Upon investigation, the evaluator found that the TOE generally authenticates users or client who present an external X509 Client certificate while the Authentication type can be selected as Certificate with a SHA1 fingerprint of the Client's certificate. Please refer to Section 6.1 in the AGD for additional details. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

TSS and Guidance Activities (Security Management)

5.3.3 FMT_CFG_EXT.1

5.3.3.1 FMT_CFG_EXT.1.1 TSS 1

Objective	The evaluator shall check the TSS to determine if the application requires any type of credentials and if the application installs with default credentials.
Evaluator Findings	The evaluator examined the section titled **'TOE Summary Specification'** in the Security Target to determine if the application requires any type of credentials and if the application installs with default credentials. Upon investigation, the evaluator found that the TSS states that the TOE is not installed with any default credential. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.3.4 FMT_MEC_EXT.1

5.3.4.1 FMT_MEC_EXT.1 TSS 1 **[TD0437]**

Objective	The evaluator shall review the TSS to identify the application's configuration options (e.g. settings) and determine whether these are stored and set using the mechanisms

	supported by the platform or implemented by the application in accordance with the PP-Module for File Encryption. At a minimum the TSS shall list settings related to any SFRs and any settings that are mandated in the operational guidance in response to an SFR. Conditional: If "implement functionality to encrypt and store configuration options as defined by FDP_PRT_EXT.1 in the PP-Module for File Encryption" is selected, the evaluator shall ensure that the TSS identifies those options, as well as indicates where the encrypted representation of these options is stored.
Evaluator Findings	The evaluator examined the section titled **'TOE Summary Specification'** in the Security Target to verify that the TSS identifies the application's configuration options (e.g. settings) and determine whether these are stored and set using the mechanisms supported by the platform or implemented by the application in accordance with the PP-Module for File Encryption. Upon investigation, the evaluator found that the TSS states that the TOE stores settings and configuration options in C:\ProgramData for Windows and /etc for CentOS. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.3.5 FMT_SMF.1

5.3.5.1 FMT_SMF.1 Guidance 1

Objective	The evaluator shall verify that every management function mandated by the PP is described in the operational guidance and that the description contains the information required to perform the management duties associated with the management function.
Evaluator Findings	The evaluator examined the section titled **'Security Management'** in the AGD to verify that every management function mandated by the PP is described in the operational guidance and that the description contains the information required to perform the management duties associated with the management function. Upon investigation, the evaluator found that the AGD states that the TOE allows the configuration of users, database server, authentication server, mail server, file servers, file transfer services, keys and certificates, and cryptographic protocols. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.4 TSS and Guidance Activities (Privacy)

5.4.1 FPR_ANO_EXT.1

5.4.1.1 FPR_ANO_EXT.1 TSS 1

Objective	The evaluator shall inspect the TSS documentation to identify functionality in the application where PII can be transmitted.

Evaluator Findings	The evaluator examined the section titled **'TOE Summary Specification'** in the Security Target to verify that the TSS identifies functionality in the application where PII can be transmitted. Upon investigation, the evaluator found that the TSS states that the TOE does not transmit PII over the network. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.5 TSS and Guidance Activities (Protection of the TSF)

5.5.1 FPT_AEX_EXT.1

5.5.1.1 FPT_AEX_EXT.1.1 TSS 1

Objective	The evaluator shall ensure that the TSS describes the compiler flags used to enable ASLR when the application is compiled.
Evaluator Findings	The evaluator examined the section titled **'TOE Summary Specification'** in the Security Target to verify that the TSS describes the compiler flags used to enable ASLR when the application is compiled. Upon investigation, the evaluator found that the TSS states that: **Windows** The TOE does not request that any memory is mapped to an explicit address. The TOE is compiled without any specific flags on Windows to enable ASLR (/DYNAMICBASE is enabled by default). **Linux** The TOE does not request that any memory is mapped to an explicit address. The TOE is composed of Java code. ASLR is provided by the platform-provided JRE. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.5.2 FPT_API_EXT.1

5.5.2.1 FPT_API_EXT.1TSS 1

Objective	The evaluator shall verify that the TSS lists the platform APIs used in the application.
Evaluator Findings	The evaluator examined the section titled **'TOE Summary Specification'** in the Security Target to verify that the TSS lists the platform APIs used in the application. Upon investigation, the evaluator found that the TSS states that the TOE uses the Windows platform APIs listed in Section 6.1 of the ST. The TOE uses the Linux (CentOS) platform APIs listed in Section 6.2 of the ST. Based on these findings, this assurance activity is considered satisfied.

Verdict	Pass

5.5.3 FPT_IDV_EXT.1

5.5.3.1 FPT_IDV_EXT.1 TSS 1

Objective	If "other version information" is selected the evaluator shall verify that the TSS contains an explaination of the versioning methodology.
Evaluator Findings	The evaluator examined the section titled **'TOE Summary Specification'** in the Security Target to verify that the TSS contains an explanation of the versioning methodology. Upon investigation, the evaluator found that the TSS states that the TSF is installed with a SWID tag containing a SoftwareIdentity element and an Entity element. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.5.4 FPT_TUD_EXT.1

5.5.4.1 FPT_TUD_EXT.1.1 Guidance 1

Objective	The evaluator shall check to ensure the guidance includes a description of how updates are performed.
Evaluator Findings	The evaluator examined the section titled '**Secure Updates'** in the AGD to verify that it includes a description of how updates are performed. Upon investigation, the evaluator found that the AGD provides the instructions for a secure update. This includes the process of checking for the current version and installation of the new version. Updates to the TOE are digitally signed and verified by the platform (Windows Installer or RPM Package manager) prior to installation. The TOE does not update itself, but rather relies on the platform package manager to install updates. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.5.4.2 FPT_TUD_EXT.1.2 Guidance 1

Objective	The evaluator shall verify guidance includes a description of how to query the current version of the application.
Evaluator Findings	The evaluator examined the section titled **'Secure Updates'** in the AGD to verify that it includes a description of how to query the current version of the application. Upon investigation, the evaluator found that the AGD states that the steps the operator may follow to query the system for its currently running version are: 1. On the MFT dashboard click the Help button and go to About. 2. The current version of the TOE is displayed. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.5.4.3 FPT_TUD_EXT.1.4 TSS 1

Objective	The evaluator shall verify that the TSS identifies how updates to the application are signed by an authorized source. The definition of an authorized source must be contained in the TSS. The evaluator shall also ensure that the TSS (or the operational guidance) describes how candidate updates are obtained. **TD0561 has been applied.**
Evaluator Findings	The evaluator examined the section titled **'TOE Summary Specification'** in the Security Target to verify that the TSS identifies how the application installation package and updates to it are signed by an authorized source. Upon investigation, the evaluator found that the TSS states that updates to the TOE are digitally signed and verified by the platform (Windows Installer or RPM Package manager) prior to installation. The evaluator also examined the section titled **'TOE Summary Specification'** in the Security Target to verify that the TSS describes how candidate updates are obtained. Upon investigation, the evaluator found that the TSS states that the TSF does not update itself, but rather relies on the platform package manager to install updates. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.5.4.4 FPT_TUD_EXT.1.5 TSS 1

Objective	The evaluator shall verify that the TSS identifies how the application is distributed. If "with the platform" is selected the evaluated shall perform a clean installation or factory reset to confirm that TOE software is included as part of the platform OS. If "as an additional package" is selected the evaluator shall perform the tests in FPT_TUD_EXT.2
Evaluator Findings	The evaluator examined the section titled **'TOE Summary Specification'** in the Security Target to verify that the TSS identifies how the application is distributed. Upon investigation, the evaluator found that the TSS states that the TOE is not distributed with the platform OS. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.5.4.5 FPT_TUD_EXT.2 TSS 3

Objective	The evaluator shall verify that the TSS identifies how the application installation package is signed by an authorized source. The definition of an authorized source must be contained in the TSS. **TD0561 has been applied.**
Evaluator Findings	The evaluator examined the section titled **'TOE Summary Specification'** in the Security Target to verify that the TSS identifies how the application installation package is signed by an authorized source. Upon investigation, the evaluator found that the TSS states that the TOE Updates to the TOE are digitally signed and verified by the platform (Windows Installer or RPM Package manager) prior to installation.

	Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.5.5

5.6 TSS and Guidance Activities (Trusted Path/Channels)

5.6.1 FTP_DIT_EXT.1

5.6.1.1 FTP_DIT_EXT.1 TSS 1

Objective	For platform-provided functionality, the evaluator shall verify the TSS contains the calls to the platform that TOE is leveraging to invoke the functionality. **TD0601 has been applied.** **TD0668 has been applied.**
Evaluator Findings	The evaluator examined the section titled **'TOE Summary Specification'** in the Security Target to verify that the TSS contains the calls to the platform that TOE is leveraging to invoke the functionality. Upon investigation, the evaluator found that the TOE does not support platform-provided functionality. The TSS states that the TSF encrypts all transmitted data using TLSv1.2 or SSHv2. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

6 Detailed Test Cases (Test Activities)

6.1 Filesystem (Linux)

6.1.1 FMT_CFG_EXT.1.2 Test #2

Item	Data
Test Assurance Activity	The evaluator shall install and run the application. The evaluator shall inspect the filesystem of the platform (to the extent possible) for any files created by the application and ensure that their permissions are adequate to protect them. The method of doing so varies per platform. **For Linux:** The evaluator shall run the command find -L . -perm /002 inside the application's data directories to ensure that all files are not world-writable. The command should not print any files. **TD0519 has been applied.**
Test Steps	• The evaluator installed and run the application. During installation, the TOE added two directories: /etc/HelpSystems/GoAnywhere and /opt/HelpSystems/GoAnywhere. • The evaluator ran the command find – L . -perm /002 inside the application's data directories and ensured that all the files are not world-writable as the command did not print any files.
Pass/Fail with Explanation	Pass. The permissions of the filesystem for any files created by the TOE are adequate to protect them.

6.1.2 FMT_MEC_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	If "**invoke the mechanisms recommended by the platform vendor for storing and setting configuration options**" is chosen, the method of testing varies per platform as follows: **For Linux:** The evaluator shall run the application while monitoring it with the utility strace. The evaluator shall make security-related changes to its configuration. The evaluator shall verify that strace logs corresponding changes to configuration files that reside in /etc (for system-specific configuration), in the user's home directory (for user-specific configuration), or /var/lib/ (for configurations controlled by UI and not intended to be directly modified by an administrator). **Non-applicable platforms removed.** **TD0437 has been applied.**
Test Steps	• The evaluator ran the command 'sudo ps -ef' to determine the process ID (9188) while the application is running. • The evaluator started the utility strace to observe the strace logs while making security-related changes to TOE's configuration. • The evaluator ran the application and observed the configuration at System -> Security Settings. • The evaluator updated the security settings by making the following highlighted changes in the Security Settings. • The evaluator ensured that the security settings updated successfully.

	• The evaluator verified that the strace logs indicate corresponding changes to the configuration files that reside in /etc while the security-related changes were made in its TOE's configuration. • The evaluator then observed the SFTP configuration at Services -> Service Manager while monitoring it with the strace utility. • The evaluator updated the SFTP server configuration by updating the name from default to helpsystems and ensured that the security settings updated successfully. • The evaluator verified that the strace logs indicate corresponding changes to the configuration files that reside in /etc while the security-related changes were made in its TOE's configuration. • The evaluator then observed the HTTPS configuration at Services -> Service Manager. • The evaluator updated the HTTPS server configuration by updating the name from default to helpsystems and ensured that the security settings updated successfully. • The evaluator verified that the strace logs indicate corresponding changes to the configuration files that reside in /etc while the security-related changes were made in its TOE's configuration.
Pass/Fail with Explanation	Pass. The TOE writes security related configuration changes to /etc folder as expected.

6.1.3 FPT_AEX_EXT.1.4 Test #1

Item	Data
Test Assurance Activity	The evaluator shall run the application and determine where it writes its files. For files where the user does not choose the destination, the evaluator shall check whether the destination directory contains executable files. This varies per platform: **For Linux:** The evaluator shall run the program, mimicking normal usage, and note where all user-modifiable files are written. The evaluator shall ensure that there are no executable files stored in the same directories to which the application wrote user-modifiable files. **Non-applicable platforms removed.** **TD0445 has been applied.**
Test Steps	• The evaluator ran the command 'sudo ps -ef' to determine the process ID (9188) while the application is running. • The evaluator started the utility strace to observe the strace logs while making security-related changes to TOE's configuration. • The evaluator ran the application and observed the configuration at System -> Security Settings. • The evaluator updated the security settings by making the following highlighted changes in the Security Settings. • The evaluator ensured that the security settings updated successfully. • The evaluator verified that the strace logs indicate corresponding changes to the configuration files that reside in /etc while the security-related changes were made in its TOE's configuration.

	• The evaluator then observed the SFTP configuration at Services -> Service Manager while monitoring it with the strace utility. • The evaluator updated the SFTP server configuration by updating the name from default to helpsystems and ensured that the security settings updated successfully. • The evaluator verified that the strace logs indicate corresponding changes to the configuration files that reside in /etc while the security-related changes were made in its TOE's configuration. • The evaluator then observed the HTTPS configuration at Services -> Service Manager. • The evaluator updated the HTTPS server configuration by updating the name from default to helpsystems and ensured that the security settings updated successfully. • The evaluator verified that the strace logs indicate corresponding changes to the configuration files that reside in /etc while the security-related changes were made in its TOE's configuration. • The evaluator ensured that there are no executable files stored in the same directories to which the application wrote user-modifiable files as the command did not print any output.
Pass/Fail with Explanation	Pass. The user modifiable files are not written to the same directory containing executable files.

6.1.4 FPT_IDV_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	The evaluator shall install the application, then check for the existence of version information. If SWID tags is selected the evaluator shall check for a .swidtag file. The evaluator shall open the file and verify that is contains at least a SoftwareIdentity element and an Entity element.
Test Steps	• The evaluator ensured that the application is fully installed and then checked for version information. • The evaluator navigated to the swidtag file and ensured that it contains a SoftwareIdentity and an Entity element.
Pass/Fail with Explanation	Pass. The TOE versioning is consistent with the SWID versioning methodology.

6.1.5 FPT_LIB_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	The evaluator shall install the application and survey its installation directory for dynamic libraries. The evaluator shall verify that libraries found to be packaged with or employed by the application are limited to those in the assignment.
Test Steps	• The evaluator surveyed the installation directory for dynamic libraries and found the libraries at /opt/HelpSystems/GoAnywhere/lib directory. • The evaluator further checked the tomcat/bin and tomcat/lib folders of the installation directory to ensure that the third-party libraries listed in section 6.3 of the Security Target were found.

	• The evaluator verified that libraries found to be packaged with or employed by the application are limited to those in the assignment i.e., the third-party libraries listed in section 6.3 of the Security Target.
Pass/Fail with Explanation	Pass. The evaluator Verified that libraries found to be packaged with or employed by the application are limited to those in the assignment. This meets the test requirements.

6.1.6 FPT_TUD_EXT.1.3 Test #1

Item	Data
Test Assurance Activity	The evaluator shall verify that the application's executable files are not changed by the application. The evaluator shall complete the following test: **For all other platforms**: The evaluator shall install the application and then locate all of its executable files. The evaluator shall then, for each file, save off either a hash of the file or a copy of the file itself. The evaluator shall then run the application and exercise all features of the application as described in the ST. The evaluator shall then compare each executable file with the either the saved hash or the saved copy of the file. The evaluator shall verify that these are identical. **TD0548 has been applied.**
Test Steps	• The evaluator installed the application and located all of its executable files using the find command in the GoAnywhere Directory. • The evaluator shall then, for each file, saved off the hash of the executable files. The entire list of files and their hashes are provided in the form of a .txt file below. • The evaluator then ran the application and exercised all features of the application as described in the ST. • The evaluator then generated hash for each executable after exercising all the features on the TOE. The entire list of files after the TOE was exercised is provided in the form of a .txt file below. • The evaluator then compared each executable file with the saved hash files obtained using the diff utility and ensured that these are identical as the utility did not provide any output indicating differences.
Pass/Fail with Explanation	Pass. The Hash of the executable files of the application are not changed by the application. This meets testing requirements.

6.1.7 FPT_TUD_EXT.2.2 Test #1

Item	Data
Test Assurance Activity	**For All Other Platforms:** The evaluator shall record the path of every file on the entire filesystem prior to installation of the application, and then install and run the application. Afterwards, the evaluator shall then uninstall the application, and compare the resulting filesystem to the initial record to verify that no files, other than configuration, output, and audit/log files, have been added to the filesystem.
Test Steps	• The evaluator recorded the path of every file on the entire filesystem prior to installation of the application using the command “find / -type f > before_install.txt”. • The evaluator then started installing the application.

	• The evaluator observed during the installation process that the application will be installed at /opt/HelpSystems/GoAnywhere and the configuration files will be stored at /etc/HelpSystems/GoAnywhere. • The evaluator ensured that the application was successfully installed. • The evaluator ran the application and updated the security settings successfully. • The evaluator ensured that the TOE was able to communicate with the DB server successfully. • The evaluator inspected the filesystem after installation and ensured that the installation files were present at /opt/HelpSystems/GoAnywhere and the configuration files stored at /etc/HelpSystems/GoAnywhere. • The evaluator then started uninstalling the application. • The evaluator ensured that the application was uninstalled successfully. • The evaluator then recorded the path of every file on the entire filesystem after uninstalling the application. • The evaluator verified that no files, other than configuration, output, and audit/log files, have been added to the filesystem. • The evaluator ensured that the configuration files related to Helpsystems were removed from /etc folder after uninstalling the application. • The evaluator further compared the resulting filesystem to the initial record to verify that no files, other than configuration, output, and audit/log files, have been added to the filesystem.
Pass/Fail with Explanation	Pass. The evaluator verified that no other files, other than configuration, output, and audit/log files, have been added to the filesystem by the TOE. This meets the test requirements.

6.2 Network (Linux)

6.2.1 FCS_CKM.2.1 – RSA

Item	Data
Test Assurance Activity	The evaluator shall verify the correctness of the TSF's implementation of RSAES-PKCS1-v1_5 by using a known good implementation for each protocol selected in FTP_DIT_EXT.1 that uses RSAES-PKCS1-v1_5.
Pass/Fail with Explanation	Pass. The evaluator verified the correctness of the TSF's implementation of RSAES-PKCS1-v1_5 as a part of FCS_TLSS_EXT.1.3 Test #1.

6.2.2 FCS_CKM.2.1 – DH14

Item	Data
Test Assurance Activity	The evaluator shall verify the correctness of the TSF's implementation of Diffie-Hellman group 14 by using a known good implementation for each protocol selected in FTP_DIT_EXT.1 that uses Diffie-Hellman group 14.
Pass/Fail with Explanation	Pass. The evaluator verified the correctness of the TSF's implementation of Diffie-Hellman group 14 as a part of FCS_SSHS_EXT1.6 Test #1 and FCS_SSHC_EXT1.6 Test #1.

6.2.3 FCS_HTTPS_EXT.1.1/Client Test #1

Item	Data
Test Assurance Activity	The evaluator shall attempt to establish an HTTPS connection with a webserver, observe the traffic with a packet analyzer, and verify that the connection succeeds and that the traffic is identified as TLS or HTTPS. **TD0668 has been applied.**
Test Steps	• Configure the TOE and the HTTPS server for HTTPS POST function. • Create the RootCA and the server certificates using XCA. • Upload RootCA onto the TOE's trust store. • Configure the server to accept SSL. • Establish an HTTPS POST connection from the TOE to a webserver. • Verify that the connection succeeds, and traffic is encrypted with TLS.
Pass/Fail with Explanation	Pass. The TOE attempted to establish a TLS connection with a web server, observed the traffic with packet analyzer(Wireshark) and verified successful connection. The evaluator also identified the traffic as TLS.

6.2.4 FCS_HTTPS_EXT.1.1/Server Test #1

Item	Data
Test Assurance Activity	The evaluator shall attempt to establish an HTTPS connection to the TOE using a client, observe the traffic with a packet analyzer, and verify that the connection succeeds and that the traffic is identified as TLS or HTTPS. **TD0668 has been applied.**
Test Steps	• The evaluator ensured that the TOE's default https web server runs on port 8001. • Establish an HTTPS connection from a web browser (client) to the TOE. • Verify that the connection succeeds, and traffic is encrypted with TLS.
Pass/Fail with Explanation	Pass. The evaluator attempted to establish a TLS connection to the TOE using a client (web browser), observed the traffic with a packet analyzer(Wireshark) and verified that the connection succeeds. The evaluator also identified the traffic as TLS.

6.2.5 FDP_NET_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	**Test 1:** The evaluator shall run the application. While the application is running, the evaluator shall sniff network traffic ignoring all non-application associated traffic and verify that any network communications witnessed are documented in the TSS or are user-initiated.
Test Steps	• While the application is running, capture the packets using Wireshark and filter out all non-network related traffic. • The evaluator only observed the following traffic in the above packet capture: • The traffic with server 192.168.254.107 port 3389 is non application related and is the traffic associated between the evaluator's workstation and the VM using Remote Desktop connection (RDP). • Traffic observed throughout the packet capture between the TOE (10.1.3.253) and the MySQL server (10.1.3.51 port 3306) is the User Configured database.

	• Traffic observed throughout the packet capture between the TOE (10.1.3.253 port 636) and the LDAP server (10.1.3.51) is the User Configured Authentication server. • Traffic observed throughout the packet capture between the TOE (10.1.3.253 port 8001) and the HTTPS (10.1.3.51) is the remote HTTPS administration of the TOE and is recorded in the TSS. • Traffic observed throughout the packet capture between the TOE (10.1.3.253) and the SSH server (10.1.3.51 port 22) is the user-initiated connection from the TOE to the SSH server. • The NTP protocol packets shown below denotes the communication of the VM hosting the TOE and the Network Time protocol server to synchronize system time and is non application related. • Traffic observed throughout the packet capture between the TOE (10.1.3.253 port 1214) and the SSH client (10.1.3.51) is the user-initiated connection from the TOE to the SSH server.
Pass/Fail with Explanation	Pass. All the network communication witnessed when the TOE is running are user initiated.

6.2.6 FDP_NET_EXT.1.1 Test #2

Item	Data
Test Assurance Activity	**Test 2:** The evaluator shall run the application. After the application initializes, the evaluator shall run network port scans to verify that any ports opened by the application have been captured in the ST for the third selection and its assignment. This includes connection-based protocols (e.g. TCP, DCCP) as well as connectionless protocols (e.g. UDP).
Test Steps	• Start the application and wait for the application to initialize. • Run network port scan to verify all open ports are listed in the ST. • The evaluator provided rationale regarding the open ports which are not related to the TOE. • The evaluator provided rationale about the ports that are kept open by the TOE. • The evaluator ran the UDP port scan to determine the udp ports that are kept open by the TOE. • The evaluator provided rationale regarding all open ports which are not related to the TOE.
Pass/Fail with Explanation	Pass. The TOE did not open any unexpected ports. This meets the testing requirements.

6.2.7 FTP_DIT_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	Test 1: The evaluator shall exercise the application (attempting to transmit data; for example, by connecting to remote systems or websites) while capturing packets from the application. The evaluator shall verify from the packet capture that the traffic is encrypted with HTTPS, TLS, DTLS, SSH, or IPsec in accordance with the selection in the ST.

	TD0601 has been applied. **TD0668 has been applied.**
Test Steps	TOE as HTTPS server: • Note: The TOE's default https web server runs on port 8001 • The evaluator configured the TOE to act as a HTTPS server in System -> Admin Server to serve the web clients who want to administrate the TOE remotely over HTTPS/TLS. • The evaluator attempted to establish an HTTPS connection from a web browser (client) to the TOE and was able to successfully access the TOE. • The evaluator verified that the connection succeeds, and traffic is encrypted with TLS. TOE as TLS Client to the Database server: • The evaluator imported the RootCA certificate into the keyvault of the TOE that signed the server certificate. • The evaluator loaded the server certificate (server.crt) and the server key (server_key.pem) to the database server. • The evaluator configured the server to leverage the loaded certificate and key along with TLS_RSA_WITH_AES_128_CBC_SHA as the cipher suite. • The evaluator ensured that the configuration was applied to the mysql server. • The evaluator attempted a connection from the TOE to the Data base server and verified the connection to be successful. • The evaluator observed the packet capture and ensured that the traffic is encrypted with TLS. TOE as SSH/SFTP server: • The evaluator set the user's authentication type as Password based. • The evaluator attempted to login to the TOE using a valid username/password combination • The evaluator ensured that the connection is successful. • The evaluator observed the packet capture and ensured that the traffic is encrypted with SSH when attempted to exercise the SSH service of the TOE. TOE as SSH Client: • The evaluator configured the SSH server to allow aes128-cbc algorithm. • The evaluator attempted a connection from the TOE to the server and verify the connection succeeds. • The evaluator verified through packet capture that the traffic encrypted with SSH when attempted to exercise the SSH client service on the TOE.
Pass/Fail with Explanation	Pass. All the traffic captured when the TOE is exercised is either TLS or SSH.

6.2.8 FTP_DIT_EXT.1.1 Test #2

Item	Data
Test Assurance Activity	Test 2: The evaluator shall exercise the application (attempting to transmit data; for example by connecting to remote systems or websites) while capturing packets from the application. The evaluator shall review the packet capture and verify that no sensitive data is transmitted in the clear. **TD0601 has been applied.**

	TD0668 has been applied.
Pass/Fail with Explanation	Pass. The evaluator reviewed the packet capture for each connection and verified that no sensitive data is transmitted in the clear.

6.2.9 FTP_DIT_EXT.1.1 Test #3

Item	Data
Test Assurance Activity	Test 3: The evaluator shall inspect the TSS to determine if user credentials are transmitted. If credentials are transmitted the evaluator shall set the credential to a known value. The evaluator shall capture packets from the application while causing credentials to be transmitted as described in the TSS. The evaluator shall perform a string search of the captured network packets and verify that the plaintext credential previously set by the evaluator is not found. **TD0601 has been applied.** **TD0668 has been applied.**
Test Steps	TOE as TLS Client to the Database server: • The evaluator imported the RootCA certificate into the keyvault of the TOE that signed the server certificate. • The evaluator loaded the server certificate (server.crt) and the server key (server_key.pem) to the database server. • The evaluator configured the server to leverage the loaded certificate and key along with TLS_RSA_WITH_AES_128_CBC_SHA as the cipher suite. • The evaluator ensured that the configuration was applied to the mysql server. • The evaluator attempted a connection from the TOE to the Data base server using credentials user:GADATA and password:123TesT321 and verified the connection to be successful. • The evaluator observed the packet capture and ensured that the traffic is encrypted with TLS. The evaluator also verified that sensitive data was not sent in plaintext and was sent as encrypted application data. • The credentials used to access the Database server were username: GADATA and password: 123TesT321. The evaluator performed a string search of the captured network packets and verify that the plaintext credentials previously set by the evaluator are not found. TOE as SSH Client: • The evaluator set the SSH server as 10.1.3.51 • The evaluator attempted a connection from the TOE to the SFTP server and verify the connection succeeds. • The evaluator verified through packet capture that the traffic encrypted with SSH when attempted to exercise the SSH client service on the TOE. • The credentials used to access the SFTP server were username: acumensec and password: 123TesT321. The evaluator performed a string search of the captured network packets and verify that the plaintext credentials previously set by the evaluator are not found.

Pass/Fail with Explanation	Pass. The evaluator performed a string search of the captured network packets and verified that the plaintext credential set by the evaluator is not found. This meets testing requirements.

6.3 Operation (Linux)

6.3.1 FMT_CFG_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	If the application uses any default credentials, the evaluator shall run the following tests. **Test 1:** The evaluator shall install and run the application without generating or loading new credentials and verify that only the minimal application functionality required to set new credentials is available.
Test Steps	The evaluator observed the TSS which states that the TOE does not come with default credentials. Therefore, this test case is not applicable.
Pass/Fail with Explanation	Pass. the TOE does not come with default credentials.

6.3.2 FMT_CFG_EXT.1.1 Test #2

Item	Data
Test Assurance Activity	If the application uses any default credentials, the evaluator shall run the following tests. **Test 2:** The evaluator shall attempt to clear all credentials and verify that only the minimal application functionality required to set new credentials is available
Test Steps	The evaluator observed the TSS which states that the TOE does not come with default credentials. Therefore, this test case is not applicable.
Pass/Fail with Explanation	Pass. the TOE does not come with default credentials.

6.3.3 FMT_CFG_EXT.1.1 Test #3

Item	Data
Test Assurance Activity	If the application uses any default credentials, the evaluator shall run the following tests. **Test 3:** The evaluator shall run the application, establish new credentials, and verify that the original default credentials no longer provide access to the application.
Test Steps	The evaluator observed the TSS which states that the TOE does not come with default credentials. Therefore, this test case is not applicable.
Pass/Fail with Explanation	Pass. the TOE does not come with default credentials.

6.3.4 FMT_SMF.1.1 Test #1

Item	Data
Test Assurance Activity	The evaluator shall test the application's ability to provide the management functions by configuring the application and testing each option selected from above. The evaluator is expected to test these functions in all the ways in which the ST and guidance documentation state the configuration can be managed.
Test Steps	• The Management functions of the TOE are as shown below Configure users • The evaluator created admin users by navigating to Users -> Add Admin User. • The evaluator configured the Admin User roles at Users -> Admin User Roles -> Add Role. • The evaluator set the username, authentication type and Roles and clicked Save. • The admin user 'testadmin' was created and was configured as an Agent manager and Log viewer. • The evaluator created web users at Users -> Add Web Users. • The evaluator configured the username of the web user as 'fmtsmf'. • The evaluator configured the protocol section of the web user as SFTP. • The evaluator successfully created the web user named 'fmtsmf' configured for SFTP services. Configure database server • The evaluator configured the database server at System -> DataBase Configuration. Configure authentication server • The evaluator configured the authentication server at System -> Admin Server. • The evaluator ensured that the configuration was successful. Configure mail server • The evaluator configured the mail server at Resources -> SMTP Servers. • The evaluator set the port to 465 and configured the user and connection type. • The evaluator ensured that the configuration was successful. Configure file servers • The evaluator configured the File server at Services -> Service Manager and set the automatically start service to Yes and configured the upload restrictions. • The evaluator configured the File server algorithm parameters. • The evaluator configured the port on which the server runs. • The evaluator configured the Host keys. • The evaluator ensured that the configuration was successful. File transfer services • The evaluator created the SSH server resource. • The evaluator ensured that the TOE has the ability to link server resource and create a file transfer project • The evaluator executed the project and ensured that the file transfer was successful. Configure keys and certificates • The evaluator ensured that the TOE has the ability to create and manage Key vaults.

	• The evaluator ensured that the TOE has the ability to create key vaults. • The evaluator ensured that the TOE has the ability to add certificates. • The evaluator ensured that the TOE has the ability to add key pairs. • The evaluator ensured that the TOE has the ability to add file-based certs and keys. Configure cryptographic protocols • The evaluator ensured that the TOE has the ability to configure cryptographic protocols. The evaluator ensured that the configuration was successful.
Pass/Fail with Explanation	The evaluator verified that the TOE can be configured as stated in the ST and Guidance documentation.

6.3.5 FPR_ANO_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	If **require user approval before executing is selected**, the evaluator shall run the application and exercise the functionality responsibly for transmitting PII and verify that user approval is required before transmission of the PII.
Test Steps	As stated in the ST, TOE does not expressly transmit any PII. Therefore, this test is considered satisfied.
Pass/Fail with Explanation	Pass. As stated in the ST, TOE does not expressly transmit any PII. Therefore, this test is considered satisfied.

6.3.6 FPT_AEX_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	The evaluator shall perform either a static or dynamic analysis to determine that no memory mappings are placed at an explicit and consistent address. The method of doing so varies per platform. For those platforms requiring the same application running on two different systems, the evaluator may alternatively use the same device. After collecting the first instance of mappings, the evaluator must uninstall the application, reboot the device, and reinstall the application to collect the second instance of mappings. **For Linux:** The evaluator shall run the same application on two different Linux systems. The evaluator shall then compare their memory maps using pmap -x PID to ensure the two different instances share no mapping locations. **TD0544 has been applied.** **Non-applicable platforms removed.**
Test Steps	• The evaluator ran the command 'netstat -tulnp' and 'ps auxZ | grep tomcat' to identify the process ID running on CentOS VM1 and CentOS VM2. • The TOE is run on two different CentOS VMs. Note the Process ID is different for both instances. The evaluator verified that there was no mapping of locations (i.e. memory address locations were different between both hosts for the TOE) when the TOE was run on two different instances. • Execute the command as root user pmap -x 16150 on CentOS VM 1(process id from the above command).

	• Execute the command as root user pmap -x 10797 on CentOS VM 2(process id from the above command).
Pass/Fail with Explanation	Pass. The evaluator has run the application on two different machines and has observed that the application running on two different machines share no memory mapping location. This meets the test requirements.

6.3.7 FPT_AEX_EXT.1.3 Test #1

Item	Data
Test Assurance Activity	The evaluator shall configure the platform in the ascribed manner and carry out one of the prescribed tests: **For Linux:** The evaluator shall ensure that the application can successfully run on a system with either SELinux or AppArmor enabled and enforcing in enforce mode. **Non-applicable platforms removed.** **TD0435 has been applied.**
Test Steps	• Go to the system on which the TOE is running and verify the status of SE linux and enforcing in enforce mode on the system
Pass/Fail with Explanation	Pass. The TOE can successfully run on a system that has SElinux enabled and enforcing in enforce mode.

6.3.8 FPT_TUD_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	The evaluator shall check for an update using procedures described in either the application documentation or the platform documentation and verify that the application does not issue an error. If it is updated or if it reports that no update is available this requirement is considered to be met.
Test Steps	• Start the TOE (Goanyhwere MFT), Click on Help and Click on Check for updates. • As seen in screenshot No update is available, and the latest version of TOE is 6.8.3 which also matches the documented and installed version.
Pass/Fail with Explanation	Pass. No update is available, and the latest version of TOE is being used. This meets testing requirements.

6.3.9 FPT_TUD_EXT.1.2 Test #1

Item	Data
Test Assurance Activity	The evaluator shall query the application for the current version of the software according to the operational user guidance. The evaluator shall then verify that the current version matches that of the documented and installed version.
Test Steps	• The evaluator queried the application for the current version of the software according to the operational user guidance at Help -> About. • The evaluator then verified that the current version 6.8.3 matches that of the documented and installed version.
Pass/Fail with Explanation	Pass. The evaluator queried the application for the current version of the software according to the operational user guidance and verified that the current version matches that of the documented and installed version.

6.4 PKG_TLSC (Linux)

6.4.1 FCS_TLSC_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	The evaluator shall establish a TLS connection using each of the cipher suites specified by the requirement. This connection may be established as part of the establishment of a higher-level protocol, e.g., as part of an EAP session. It is sufficient to observe the successful negotiation of a cipher suite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic in an attempt to discern the cipher suite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).
Test Steps	• The evaluator created a server certificate that was signed by the Root CA and the server key using the XCA tool. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator loaded the server certificate (server.crt) and the server key (server_key.pem) to the database server. • The evaluator configured the server to leverage the loaded certificate and key along with each of the cipher suite supported by the TOE. ▪ TLS_RSA_WITH_AES_128_CBC_SHA ▪ TLS_RSA_WITH_AES_256_CBC_SHA ▪ TLS_RSA_WITH_AES_128_CBC_SHA256 ▪ TLS_RSA_WITH_AES_256_CBC_SHA256 ▪ TLS_RSA_WITH_AES_128_GCM_SHA256 ▪ TLS_RSA_WITH_AES_256_GCM_SHA384 ▪ TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 ▪ TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 • The evaluator attempted a connection from the TOE to the Data base server and verified the connection to be successful. • The evaluator observed the packet capture and ensured that the TLS connection was successful using the configured cipher suite. • The evaluator created a server ec certificate that was signed by the Root CA_ec and the server key using the XCA tool. • The evaluator imported the RootCA_ec certificate into the key vault of the TOE that signed the ec server certificate. • The evaluator loaded the server certificate (server_ec.crt) and the server key (server_ec_key.pem) to the database server. • The evaluator configured the server to leverage the loaded certificate and key along with along with each of the cipher suite supported by the TOE. ▪ TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 ▪ TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 ▪ TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 ▪ TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 • The evaluator attempted a connection from the TOE to the Data base server and verified the connection to be successful.

	• The evaluator observed the packet capture and ensured that the TLS connection was successful using the configured cipher suite.
Pass/Fail with Explanation	Pass. The evaluator established a TLS connection using each of the cipher suites with the Data Base server and observed the successful negotiation using each of the claiming cipher suite specified. This meets the testing requirements.

6.4.2 FCS_TLSC_EXT.1.1 Test #2

Item	Data
Test Assurance Activity	The goal of the following test is to verify that the TOE accepts only certificates with appropriate values in the extendedKeyUsage extension, and implicitly that the TOE correctly parses the extendedKeyUsage extension as part of X.509v3 server certificate validation. The evaluator shall attempt to establish the connection using a server with a server certificate that contains the Server Authentication purpose in the extendedKeyUsage extension and verify that a connection is established. The evaluator shall repeat this test using a different, but otherwise valid and trusted, certificate that lacks the Server Authentication purpose in the extendedKeyUsage extension and ensure that a connection is not established. Ideally, the two certificates should be similar in structure, the types of identifiers used, and the chain of trust.
Test Steps	Note: The following test performed satisfies the FCS_TLSC_EXT.1.3 Test #4 requirement where the evaluator demonstrated that a server using a certificate which does not have a valid identifier (in the SAN) results in an authentication failure which was confirmed through wireshark packet capture where the client could not connect to the server as the server returned a certificate_unknown error to the client and also confirmed through TOE logs that the TOE could not connect to the server as there was no subject alternative name found matching IP address 10.1.3.51. • The evaluator imported the RootCA certificate into the keyvault of the TOE that signed the server certificate. • The evaluator ensured the server certificate that is being used for a TLS connection contain Server Authentication purpose in the extendedKeyUsage extension. • The evaluator loaded the server certificate (server.crt) and the server key (server_key.pem) to the database server and configured the server to leverage the loaded certificate and key to establish a TLS connection with the TOE. • The evaluator attempted a connection from the TOE to the Data base server and verified the connection to be successful. • The evaluator observed the packet capture and ensured that the connection using a server with a server certificate that contains the Server Authentication purpose in the extendedKeyUsage extension established successfully as seen in packet 2.

	• The evaluator created an identical certificate (server_nsa.crt) that is similar in structure, the types of identifiers used, and the chain of trust but lacks the Server Authentication purpose in the extendedKeyUsage extension. • The evaluator loaded the server certificate (server_nsa.crt) and the server key (server_key.pem) to the database server and configured the server to leverage the loaded certificate and key to establish a TLS connection with the TOE. • The evaluator attempted a connection from the TOE to the Data base server and verified that the connection did not establish. • The evaluator observed the packet capture and ensured that the connection using a server with a server certificate that lacks the Server Authentication purpose in the extendedKeyUsage extension did not establish. The packet 2 in the screen capture below shows the missing Extended key usage field in extensions. • The evaluator further verified the debug logs on the TOE to ensure that the connection did not establish due to invalid server extended key usage.
Pass/Fail with Explanation	Pass. The TOE established the connection using a server with a server certificate that contains the Server Authentication purpose in the extendedKeyUsage extension and did not establish a connection with a server certificate that lacks the Server Authentication purpose in the extendedKeyUsage extension. This meets the testing requirements.

6.4.3 FCS_TLSC_EXT.1.1 Test #3

Item	Data
Test Assurance Activity	The evaluator shall send a server certificate in the TLS connection that does not match the server-selected cipher suite (for example, send a ECDSA certificate while using the TLS_RSA_WITH_AES_128_CBC_SHA cipher suite or send a RSA certificate while using one of the ECDSA cipher suites.) The evaluator shall verify that the product disconnects after receiving the server's Certificate handshake message.
Test Steps	• The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator attempted a connection to the server and observed the output on the TOE that indicated that the TOE could not to the server. • The evaluator used the "acumen-tlsc-mysql" tool to send an RSA server certificate in the TLS connection while using the cipher suite TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 when the TOE attempted to connect to the server. The evaluator observed the tool output which indicated that the TLS connection did not establish due to FATAL alert returned by the server. • The evaluator observed the packet capture to ensure that the server sent an RSA server certificate (as seen in packet 3) in the TLS connection while using the cipher suite TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 (as seen in packet 2) when the TOE attempted to connect to the server. • The evaluator observed that the TOE disconnected after receiving the server's Certificate handshake message.

Pass/Fail with Explanation	Pass. The TOE disconnected with the remote server after receiving the server's Certificate handshake message as the server was using a cipher suite that did not match the certificate. This meets the testing requirements.

6.4.4 FCS_TLSC_EXT.1.1 Test #4

Item	Data
Test Assurance Activity	The evaluator shall configure the server to select the TLS_NULL_WITH_NULL_NULL cipher suite and verify that the client denies the connection.
Test Steps	• The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator attempted a connection to the server and observed the output on the TOE that indicated that the TOE could not to the server. • The evaluator used the "acumen-tlsc-mysql" tool to attempt a connection by the remote TLS server using the TLS_NULL_WITH_NULL_NULL cipher suite. The evaluator observed the tool output which indicated that the TLS connection did not establish due to FATAL alert returned by the server. • The evaluator observed the packet capture to ensure that the server attempted a connection with TLS_NULL_WITH_NULL_NULL cipher suite (as seen in packet 2) and verified that the client denies the connection (as seen in packet 3).
Pass/Fail with Explanation	Pass. The TOE denies a connection to a server using the TLS_NULL_WITH_NULL_NULL cipher suite. This meets the testing requirements.

6.4.5 FCS_TLSC_EXT.1.1 Test #5.1

Item	Data
Test Assurance Activity	Change the TLS version selected by the server in the Server Hello to an undefined TLS version (for example 1.5 represented by the two bytes 03 06) and verify that the client rejects the connection.
Test Steps	• The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator attempted a connection to the server and observed the output on the TOE that indicated that the TOE could not to the server. • The evaluator used the "acumen-tlsc-mysql" tool to attempt a connection by a remote TLS server using an undefined TLS version (0x0001) and observed the tool output which indicated that the TLS version was set to 0x0001. • The evaluator observed the packet capture and ensured that the TOE rejected the connection due to protocol version as the TLS version selected by the server in the Server Hello was set to an undefined TLS version (0x0001).
Pass/Fail with Explanation	Pass. The TOE rejected the connection when the server selects an undefined version of TLS and verified that the TOE rejected the connection. This meets the testing requirements.

6.4.6 FCS_TLSC_EXT.1.1 Test #5.2

Item	Data

Test Assurance Activity	Change the TLS version selected by the server in the Server Hello to the most recent unsupported TLS version (for example 1.1 represented by the two bytes 03 02) and verify that the client rejects the connection.
Test Steps	• The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator attempted a connection to the server and observed the output on the TOE that indicated that the TOE could not to the server. • The evaluator used the "acumen-tlsc-mysql" tool to send a Server Hello with an unsupported TLS version (TLS v1.1) in the TLS connection when the TOE attempted to connect to the server. The evaluator observed the tool output which indicated that the TLS connection did not establish due to FATAL alert: PROTOCOL_VERSION returned by the TOE. • The evaluator observed the packet capture to ensure that the server sent a Server Hello with an unsupported TLS version (TLS v1.1) in the TLS connection when the TOE attempted to connect to the server and verified that the TLS connection did not establish due to FATAL alert: PROTOCOL_VERSION returned by the server.
Pass/Fail with Explanation	Pass. The TOE rejected the connection when the server selects a recently unsupported version of TLS. This meets the testing requirements.

6.4.7 FCS_TLSC_EXT.1.1 Test #5.3

Item	Data
Test Assurance Activity	[conditional] If DHE or ECDHE cipher suites are supported, modify at least one byte in the server's nonce in the Server Hello handshake message, and verify that the client does not complete the handshake and no application data flows.
Test Steps	• The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator attempted a connection to the server and observed the output on the TOE that indicated that the TOE could not to the server. • The evaluator used the "acumen-tlsc-mysql" tool to attempt a connection with server that modifies the server's nonce in the server hello message. The evaluator observed the tool output which indicated that the TLS connection did not establish due to WARNING alert: DECRYPT_ERROR returned by the TOE. Note: Certificate modification is done by acumen-tlsc-mysql tool as shown below to meet the test requirements. server.crt is the server certificate and the server_key.pem is the server private key. • The evaluator observed the packet capture and ensured that the client does not complete the handshake and no application data flows.
Pass/Fail with Explanation	Pass. The evaluator ensured that the client does not complete the TLS handshake with a remote server due to an invalid server nonce in the Server Hello message and no application data flows.

6.4.8 FCS_TLSC_EXT.1.1 Test #5.4

Item	Data

Test Assurance Activity	Modify the server's selected cipher suite in the Server Hello handshake message to be a cipher suite not presented in the Client Hello handshake message. The evaluator shall verify that the client does not complete the handshake and no application data flows.
Test Steps	• The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator attempted a connection to the server and observed the output on the TOE which indicated that the TOE could not to the server due to "illegal_paramater". • The evaluator used the "acumen-tlsc-mysql" tool to attempt a connection with server that modifies the server's selected cipher suite in the Server Hello handshake message to be a cipher suite not presented in the Client Hello handshake message. The evaluator observed the tool output which indicated that the TLS connection did not establish due to TLS error: ILLEGAL_PARAMETER returned by the TOE. Certificate modification is done by acumen-tlsc-mysql tool as shown below to meet the test requirements. server.crt is the server certificate and the server_key.pem is the server private key. • The evaluator observed the packet capture and ensured that the client does not complete the handshake due to "Illegal parameter" and no application data flows.
Pass/Fail with Explanation	Pass. The evaluator verified that the client does not complete the handshake and no application data flows when an unsupported cipher suite is presented in the Server Hello handshake message .

6.4.9 FCS_TLSC_EXT.1.1 Test #5.5

Item	Data
Test Assurance Activity	[conditional] If DHE or ECDHE cipher suites are supported, modify the signature block in the server's Key Exchange handshake message, and verify that the client does not complete the handshake and no application data flows. This test does not apply to cipher suites using RSA key exchange. If a TOE only supports RSA key exchange in conjunction with TLS, then this test shall be omitted.
Test Steps	• The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator used the "acumen-tlsc-mysql" tool to attempt a connection with server that modifies the signature block in the server's Key Exchange handshake message. The evaluator observed the tool output which indicated that the TLS connection did not establish as the client did not complete the handshake due to TLS error: DECRYPT_ERROR returned by the TOE. Note: Certificate modification is done by acumen-tlsc-mysql tool as shown below to meet the test requirements. server.crt is the server certificate and the server_key.pem is the server private key. • The evaluator attempted a connection to the server and observed the output on the TOE that indicated that the TOE could not connect to the server due to "decrypt_error". • The evaluator observed the packet capture and ensured that the client does not complete the handshake due to "Decrypt error" and no application data flows.

Pass/Fail with Explanation	Pass. The evaluator verified that the client does not complete the handshake and no application data flows when a modified signature is presented by the server in the server's Key Exchange handshake message.

6.4.10 FCS_TLSC_EXT.1.1 Test #5.6

Item	Data
Test Assurance Activity	Modify a byte in the Server Finished handshake message and verify that the client does not complete the handshake and no application data flows.
Test Steps	• The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator used the "acumen-tlsc-mysql" tool to attempt a connection with server that modifies a byte in the Server Finished handshake message. The evaluator observed the tool output which indicated that the TLS connection did not establish as the client did not complete the handshake due to TLS error: DECRYPT_ERROR returned by the TOE. Note: Certificate modification is done by acumen-tlsc-mysql tool as shown below to meet the test requirements. server.crt is the server certificate and the server_key.pem is the server private key. • The evaluator attempted a connection to the server and observed the output on the TOE that indicated that the TOE could not connect to the server due to "decrypt_error". • The evaluator observed the packet capture and ensured that the client does not complete the handshake and no application data flows.
Pass/Fail with Explanation	Pass. The evaluator verified that the client does not complete the handshake and no application data flows when a modified server's finished handshake message is sent.

6.4.11 FCS_TLSC_EXT.1.1 Test #5.7

Item	Data
Test Assurance Activity	Send a message consisting of random bytes from the server after the server has issued the Change Cipher Spec message and verify that the client does not complete the handshake and no application data flows. The message must still have a valid 5-byte record header in order to ensure the message will be parsed as TLS.
Test Steps	• The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator used the "acumen-tlsc-mysql" tool to attempt a connection with server that sends a message consisting of random bytes from the server after the server has issued the Change Cipher Spec message. The evaluator observed the tool output which indicated that the TLS connection did not establish as the client did not complete the handshake due to TLS error: UNEXPECTED_MESSAGE returned by the TOE. Note: Certificate modification is done by acumen-tlsc-mysql tool as shown below to meet the test requirements. server.crt is the server certificate and the server_key.pem is the server private key. • The evaluator attempted a connection to the server and observed the output on the TOE that indicated that the TOE could not connect to the server due to "unexpected_message".

	• The evaluator observed the packet capture and ensured that the client does not complete the handshake due to “encrypted alert”. The application data shown in the below pcap is the garbled message that is sent before sending the finished message and must not be mistaken for application data that indicates a successful TLS connection. The evaluator ensured that the message still have a valid 5-byte record header in order to ensure the message will be parsed as TLS.
Pass/Fail with Explanation	Pass. The evaluator verified that the client did not complete the handshake and no application data flows when a message consisting of random bytes is sent from the server after the server has issued the Change Cipher Spec message.

6.4.12 FCS_TLSC_EXT.1.2 Test #1

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. Test 1: The evaluator shall present a server certificate that contains a CN that does not match the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection fails. Note that some systems might require the presence of the SAN extension. In this case the connection would still fail but for the reason of the missing SAN extension instead of the mismatch of CN and reference identifier. Both reasons are acceptable to pass Test 1. TD0499 has been applied.
Test Steps	• The evaluator ensured that the Hostname Verification is enabled on the TOE in Admin > Security Settings. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator created a server certificate that contains a CN (11.1.3.51) that does not match the reference identifier (10.1.3.51) and does not contain the SAN extension using XCA tool. • The evaluator uploaded the server key and the certificate on the server and ensured it to contain a CN (11.1.3.51) that does not match the reference identifier (10.1.3.51) and does not contain the SAN extension. • The evaluator configured the database server to leverage the uploaded server certificate and the key for TLS handshake. • The TOE automatically configures references identifiers based on the FQDN or IP address configured used to specify the TLS server. The evaluator attempted a connection to the server and ensured that the TOE could not connect to the server. • The evaluator observed the packet capture and ensured that the client does not complete the TLS handshake due to “certificate unknown” message

	• The evaluator further observed the logs on the TOE located at /opt/HelpSystems/GoAnywhere/userdata/logs to confirm that the TOE could not connect to the server as there was no subject alternative name found matching IP address 10.1.3.51.
Pass/Fail with Explanation	Pass. The evaluator verified that the TLS connection failed when the server presented certificate that contains a CN that does not match the reference identifier and does not contain the SAN extension.

6.4.13 FCS_TLSC_EXT.1.2 Test #2

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. Test 2: The evaluator shall present a server certificate that contains a CN that matches the reference identifier, contains the SAN extension, but does not contain an identifier in the SAN that matches the reference identifier. The evaluator shall verify that the connection fails. The evaluator shall repeat this test for each supported SAN type. TD0499 has been applied.
Test Steps	• The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator created a server certificate that contains a CN (10.1.3.51) that matches the reference identifier (10.1.3.51) contains the SAN extension but does not contain an identifier in the SAN (11.1.3.51) that matches the reference identifier (10.1.3.51). • The evaluator uploaded the server key and the certificate on the server certificate that contains a CN (10.1.3.51) that does matches the reference identifier (10.1.3.51) contains the SAN extension but does not contain an identifier in the SAN (11.1.3.51) that matches the reference identifier (10.1.3.51). • The evaluator configured the database server to leverage the uploaded server certificate and the key for TLS handshake. • The TOE automatically configures references identifiers based on the FQDN or IP address configured used to specify the TLS server. The evaluator attempted a connection to the server and ensured that the TOE could not connect to the server. • The evaluator observed the packet capture and ensured that the client does not complete the TLS handshake due to “certificate unknown” message. • The evaluator further observed the logs on the TOE located at /opt/HelpSystems/GoAnywhere/userdata/logs to confirm that the TOE could not connect to the server as there was no subject alternative name found matching IP address 10.1.3.51. DNS: • The evaluator created a server certificate that contains a CN (sqlserver.acumensec.local) that matches the reference identifier (sqlserver.acumensec.local) contains the SAN extension but does not contain

	an identifier in the SAN (wrong.acumensec.local) that matches the reference identifier (sqlserver.acumensec.local). • The evaluator uploaded the certificate on the server and ensured it contains a CN (sqlserver.acumensec.local) that matches the reference identifier (sqlserver.acumensec.local) contains the SAN extension but does not contain an identifier in the SAN (wrong.acumensec.local) that matches the reference identifier (sqlserver.acumensec.local). • The evaluator configured the database server to leverage the uploaded server certificate and the key for TLS handshake. • The TOE automatically configures references identifiers based on the FQDN or IP address configured used to specify the TLS server. The evaluator attempted a connection to the server and ensured that the TOE could not connect to the server. • The evaluator observed the packet capture and ensured that the client does not complete the TLS handshake due to "certificate unknown" message. • The evaluator further observed the logs on the TOE located at /opt/HelpSystems/GoAnywhere/userdata/logs to confirm that the TOE could not connect to the server as there was no subject alternative name found matching IP address 10.1.3.51.
Pass/Fail with Explanation	Pass. The evaluator verified with each supported identifier that the TLS connection fails when the server presented a certificate that contains a CN that matches the reference identifier, contains the SAN extension, but does not contain an identifier in the SAN that matches the reference identifier.

6.4.14 FCS_TLSC_EXT.1.2 Test #3

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. Test 3: [conditional] If the TOE does not mandate the presence of the SAN extension, the evaluator shall present a server certificate that contains a CN that matches the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection succeeds. If the TOE does mandate the presence of the SAN extension, this Test shall be omitted. TD0499 has been applied.
Test Steps	Note: As mentioned in the TSS, when an IP address is configured the TOE mandates the presence of a SAN. Hence this test is N/A when the reference identifiers are IP addresses. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator created a server certificate that contains a CN (sqlserver.acumensec.local) that matches the reference identifier (sqlserver.acumensec.local) and does not contain the SAN extension using the XCA tool.

	• The evaluator uploaded the server key and the certificate on the server and ensured it to contain a CN that matches the reference identifier(sqlserver.acumensec.local) and does not contain the SAN extension. • The evaluator configured the database server to leverage the uploaded server certificate and the key for TLS handshake. • The TOE automatically configures references identifiers based on the FQDN or IP address configured used to specify the TLS server. The evaluator attempted a connection to the server and ensured that the connection was successful. • The evaluator observed the packet capture and ensured that the TLS connection was successful.
Pass/Fail with Explanation	Pass. The evaluator confirmed that as per TSS, the TOE mandates the presence of SAN extension for IP address as reference identifier and omitted the test. The evaluator verified with FQDN as identifier that the TLS connection succeeds when the server presented a certificate that contains a CN that matches the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection succeeds.

6.4.15 FCS_TLSC_EXT.1.2 Test #4

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. Test 4: The evaluator shall present a server certificate that contains a CN that does not match the reference identifier but does contain an identifier in the SAN that matches. The evaluator shall verify that the connection succeeds. TD0499 has been applied.
Test Steps	• The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator created a server certificate that contains a CN (11.1.3.51) that does not match the reference identifier (10.1.3.51) but does contain an identifier in the SAN (10.1.3.51) that matches using XCA tool. • The evaluator uploaded the server key and the certificate on the server and ensured it contains a CN (11.1.3.51) that does not match the reference identifier (10.1.3.51) but does contain an identifier in the SAN (10.1.3.51) that matches. • The evaluator configured the database server to leverage the uploaded server certificate and the key for TLS handshake. • The TOE automatically configures references identifiers based on the FQDN or IP address configured used to specify the TLS server. The evaluator attempted a connection to the server and ensured that the connection was successful. • The evaluator observed the packet capture and ensured that the TLS connection was successful.

Pass/Fail with Explanation	Pass. The evaluator verified that the TLS connection succeeds when the server presented a certificate that contains a CN that does not match the reference identifier but does contain an identifier in the SAN that matches.

6.4.16 FCS_TLSC_EXT.1.2 Test #5.1

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. The evaluator shall perform the following wildcard tests with each supported type of reference identifier. Test 5.1: [conditional]: If wildcards are supported, the evaluator shall present a server certificate containing a wildcard that is not in the left-most label of the presented identifier (e.g. foo.*.example.com) and verify that the connection fails. TD0499 has been applied.
Test Steps	CN: • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator created a server certificate containing a wildcard that is not in the left-most label of the presented identifier (sqlserver.*.acumensec.local). • The evaluator uploaded the server key and the certificate on the server and ensured it contains a wildcard that is not in the left-most label of the presented identifier (sqlserver.*.acumensec.local) in the CN. • The evaluator configured the database server to leverage the uploaded server certificate and the key for TLS handshake. • The TOE automatically configures references identifiers based on the FQDN or IP address configured used to specify the TLS server. The evaluator attempted a connection to the server and ensured that the TOE could not connect to the server. • The evaluator observed the packet capture and ensured that the client does not complete the TLS handshake due to “certificate unknown” message. • The evaluator further observed the logs on the TOE located at /opt/HelpSystems/GoAnywhere/userdata/logs to confirm that the TOE could not connect to the server as the Common Name 'sqlserver.*.acumensec.local' is not allowed according to the strict hostname verification policy. SAN: • The evaluator created a server certificate containing a wildcard that is not in the left-most label of the presented identifier (sqlserver.*.acumensec.local). • The evaluator uploaded the server key and the certificate on the server and ensured it contains a wildcard that is not in the left-most label of the presented identifier (sqlserver.*.acumensec.local) in the SAN field. • The evaluator configured the database server to leverage the uploaded server certificate and the key for TLS handshake.

	• The TOE automatically configures references identifiers based on the FQDN or IP address configured used to specify the TLS server. The evaluator attempted a connection to the server and ensured that the TOE could not connect to the server. • The evaluator observed the packet capture and ensured that the client does not complete the TLS handshake due to "certificate unknown" message. • The evaluator further observed the logs on the TOE located at /opt/HelpSystems/GoAnywhere/userdata/logs to confirm that the TOE could not connect to the server as the Subject Alternative Name 'sqlserver.*.acumensec.local' is not allowed according to the strict hostname verification policy.
Pass/Fail with Explanation	Pass. The TOE rejects the connection when the server sends a certificate with wildcard not in the left-most label of the presented identifier.

6.4.17 FCS_TLSC_EXT.1.2 Test #5.2(a)

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. The evaluator shall perform the following wildcard tests with each supported type of reference identifier. Test 5.2: [conditional]: If wildcards are supported, the evaluator shall present a server certificate containing a wildcard in the left-most label but not preceding the public suffix (e.g. *.example.com). - The evaluator shall configure the reference identifier with a single left-most label (e.g. foo.example.com) and verify that the connection succeeds. TD0499 has been applied.
Test Steps	SAN: • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator created a server certificate that contains a wildcard in the left-most label but not preceding the public suffix (*.acumensec.local). • The evaluator uploaded the server key and the certificate on the server and ensured it contains a wildcard in the left-most label but not preceding the public suffix (*.acumensec.local) in the SAN field. • The evaluator configured the database server to leverage the uploaded server certificate and the key for TLS handshake. • The TOE automatically configures references identifiers based on the FQDN or IP address configured used to specify the TLS server. The evaluator attempted a connection to the server with identifier "sqlserver.acumensec.local" resolved to 10.1.3.51 and ensured that the connection was successful.

	• The evaluator observed the packet capture and ensured that the TLS handshake was successful. CN: • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator created a server certificate that contains a wildcard in the left-most label but not preceding the public suffix (*.acumensec.local). • The evaluator uploaded the server key and the certificate on the server and ensured it contains a wildcard in the left-most label but not preceding the public suffix (*.acumensec.local) in the CN field. • The evaluator configured the database server to leverage the uploaded server certificate and the key for TLS handshake. • The TOE automatically configures references identifiers based on the FQDN or IP address configured used to specify the TLS server. The evaluator attempted a connection to the server with identifier "sqlserver.acumensec.local" resolved to 10.1.3.51 and ensured that the connection was successful. • The evaluator observed the packet capture and ensured that the TLS handshake was successful.
Pass/Fail with Explanation	Pass. The evaluator verified that the TLS connection succeeds when the server sends a certificate with wildcard in the left-most label of the presented identifier.

6.4.18 FCS_TLSC_EXT.1.2 Test #5.2(b)

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. The evaluator shall perform the following wildcard tests with each supported type of reference identifier. Test 5.2: [conditional]: If wildcards are supported, the evaluator shall present a server certificate containing a wildcard in the left-most label but not preceding the public suffix (e.g. *.example.com). - The evaluator shall configure the reference identifier without a left-most label as in the certificate (e.g. example.com) and verify that the connection fails. TD0499 has been applied.
Test Steps	SAN: • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator created a server certificate that contains a wildcard in the left-most label but not preceding the public suffix (*.acumensec.local). • The evaluator uploaded the server key and the certificate on the server and ensured it to contain a server certificate that contains a wildcard in the left-most label but not preceding the public suffix (*.acumensec.local) in the SAN field.

	• The evaluator configured the database server to leverage the uploaded server certificate and the key for TLS handshake. • The TOE automatically configures references identifiers based on the FQDN or IP address configured used to specify the TLS server. The evaluator attempted a connection to the server with identifier "acumensec.local" resolved to 10.1.3.51 and ensured that the TOE could not connect to the server. • The evaluator observed the packet capture and ensured that the client does not complete the TLS handshake due to "certificate unknown" message. • The evaluator further observed the logs on the TOE located at /opt/HelpSystems/GoAnywhere/userdata/logs to confirm that the TOE could not connect to the server as there was no subject alternative name found matching IP address 10.1.3.51. CN: • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator created a server certificate that contains a wildcard in the left-most label but not preceding the public suffix (*.acumensec.local). • The evaluator uploaded the server key and the certificate on the server and ensured it to contain a server certificate that contains a wildcard in the left-most label but not preceding the public suffix (*.acumensec.local) in the CN field. • The evaluator configured the database server to leverage the uploaded server certificate and the key for TLS handshake. • The TOE automatically configures references identifiers based on the FQDN or IP address configured used to specify the TLS server. The evaluator attempted a connection to the server with identifier "acumensec.local" resolved to 10.1.3.51 and ensured that the TOE could not connect to the server. • The evaluator observed the packet capture and ensured that the client does not complete the TLS handshake due to "certificate unknown" message. • The evaluator further observed the logs on the TOE located at /opt/HelpSystems/GoAnywhere/userdata/logs to confirm that the TOE could not connect to the server as there was no subject alternative name found matching IP address 10.1.3.51.
Pass/Fail with Explanation	Pass. The evaluator verified that the TLS connection failed when the server presented a certificate containing a wildcard in the left-most label but not preceding the public suffix while the reference identifier did not contain a left most label.

6.4.19 FCS_TLSC_EXT.1.2 Test #5.2(c)

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. The evaluator shall perform the following wildcard tests with each supported type of reference identifier.

	Test 5.2: [conditional]: If wildcards are supported, the evaluator shall present a server certificate containing a wildcard in the left-most label but not preceding the public suffix (e.g., *.example.com). - The evaluator shall configure the reference identifier with two left-most labels (e.g. bar.foo.example.com) and verify that the connection fails. TD0499 has been applied.
Test Steps	SAN: • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator created a server certificate that contains a wildcard in the left-most label but not preceding the public suffix (*.acumensec.local). • The evaluator uploaded the server key and the certificate on the server and ensured it to contain a server certificate that contains a wildcard in the left-most label but not preceding the public suffix (*.acumensec.local) in the SAN field. • The evaluator configured the database server to leverage the uploaded server certificate and the key for TLS handshake. • The TOE automatically configures references identifiers based on the FQDN or IP address configured used to specify the TLS server. The evaluator attempted a connection to the server with identifier "random.sqlserver.acumensec.local" resolved to 10.1.3.51 and ensured that the TOE could not connect to the server. • The evaluator observed the packet capture and ensured that the client does not complete the TLS handshake due to "certificate unknown" message. • The evaluator further observed the logs on the TOE located at /opt/HelpSystems/GoAnywhere/userdata/logs to confirm that the TOE could not connect to the server as there was no subject alternative name found matching IP address 10.1.3.51. CN: • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator created a server certificate that contains a wildcard in the left-most label but not preceding the public suffix (*.acumensec.local). • The evaluator uploaded the server key and the certificate on the server and ensured it to contain a server certificate that contains a wildcard in the left-most label but not preceding the public suffix (*.acumensec.local) in the CN field. • The evaluator configured the database server to leverage the uploaded server certificate and the key for TLS handshake. • The TOE automatically configures references identifiers based on the FQDN or IP address configured used to specify the TLS server. The evaluator attempted a connection to the server with identifier "random.sqlserver.acumensec.local" resolved to 10.1.3.51 and ensured that the TOE could not connect to the server. • The evaluator observed the packet capture and ensured that the client does not complete the TLS handshake due to "certificate unknown" message. • The evaluator further observed the logs on the TOE located at /opt/HelpSystems/GoAnywhere/userdata/logs to confirm that the TOE could not connect to the server as there was no subject alternative name found matching IP address 10.1.3.51.

Pass/Fail with Explanation	Pass. The evaluator verified that the TLS connection failed when the reference identifier on the client was configured with two left-most labels while the server presented a certificate containing a wildcard in the left-most label but not preceding the public suffix.

6.4.20 FCS_TLSC_EXT.1.2 Test #5.3(a)

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. The evaluator shall perform the following wildcard tests with each supported type of reference identifier. Test 5.3: [conditional]: If wildcards are supported, the evaluator shall present a server certificate containing a wildcard in the left-most label immediately preceding the public suffix (e.g. *.com). - The evaluator shall configure the reference identifier with a single left-most label (e.g. foo.com) and verify that the connection fails. TD0499 has been applied.
Test Steps	SAN: • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator created a server certificate that contains a wildcard in the left-most label immediately preceding the public suffix (*.local). • The evaluator uploaded the server key and the certificate on the server and ensured it contains a wildcard in the left-most label immediately preceding the public suffix (*.local) in the SAN field. • The evaluator configured the database server to leverage the uploaded server certificate and the key for TLS handshake. • The TOE automatically configures references identifiers based on the FQDN or IP address configured used to specify the TLS server. The evaluator attempted a connection to the server with identifier “acumensec.local” resolved to 10.1.3.51 and ensured that the TOE could not connect to the server. • The evaluator observed the packet capture and ensured that the client does not complete the TLS handshake due to “certificate unknown” message. • The evaluator further observed the logs on the TOE located at /opt/HelpSystems/GoAnywhere/userdata/logs to confirm that the TOE could not connect to the server as the Subject Alternative Name '*.local' is not allowed according to the strict hostname verification policy. CN: • The evaluator created a server certificate that contains a wildcard in the left-most label immediately preceding the public suffix (*.local).

	• The evaluator uploaded the server key and the certificate on the server and ensured it contains a wildcard in the left-most label immediately preceding the public suffix (*.local) in the CN field. • The evaluator configured the database server to leverage the uploaded server certificate and the key for TLS handshake. • The TOE automatically configures references identifiers based on the FQDN or IP address configured used to specify the TLS server. The evaluator attempted a connection to the server with identifier “acumensec.local” resolved to 10.1.3.51 and ensured that the TOE could not connect to the server. • The evaluator observed the packet capture and ensured that the client does not complete the TLS handshake due to “certificate unknown” message. • The evaluator further observed the logs on the TOE located at /opt/HelpSystems/GoAnywhere/userdata/logs to confirm that the TOE could not connect to the server as the Common Name '*.local' is not allowed according to the strict hostname verification policy.
Pass/Fail with Explanation	Pass. The evaluator verified that the TLS connection fails when the server presented a certificate with wildcard in the left-most label of the presented identifier while the reference identifier configured on the TOE does not contain a left most label.

6.4.21 FCS_TLSC_EXT.1.2 Test #5.3(b)

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. The evaluator shall perform the following wildcard tests with each supported type of reference identifier. Test 5.3: [conditional]: If wildcards are supported, the evaluator shall present a server certificate containing a wildcard in the left-most label immediately preceding the public suffix (e.g. *.com). - The evaluator shall configure the reference identifier with two left-most labels (e.g. bar.foo.com) and verify that the connection fails. TD0499 has been applied.
Test Steps	SAN: • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator created a server certificate that contains a wildcard in the left-most label immediately preceding the public suffix (*.local). • The evaluator uploaded the server key and the certificate on the server and ensured it contains a wildcard in the left-most label immediately preceding the public suffix (*.local) in the SAN field. • The evaluator configured the database server to leverage the uploaded server certificate and the key for TLS handshake.

	• The TOE automatically configures references identifiers based on the FQDN or IP address configured used to specify the TLS server. The evaluator attempted a connection to the server with identifier “sqlserver.acumensec.local” resolved to 10.1.3.51 and ensured that the TOE could not connect to the server. • The evaluator observed the packet capture and ensured that the client does not complete the TLS handshake due to “certificate unknown” message. • The evaluator further observed the logs on the TOE located at /opt/HelpSystems/GoAnywhere/userdata/logs to confirm that the TOE could not connect to the server as The Subject Alternative Name '*.local' is not allowed according to the strict hostname verification policy. CN: • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator created a server certificate that contains a wildcard in the left-most label immediately preceding the public suffix (*.local). • The evaluator uploaded the server key and the certificate on the server and ensured it contains a wildcard in the left-most label immediately preceding the public suffix (*.local) in the CN field. • The evaluator configured the database server to leverage the uploaded server certificate and the key for TLS handshake. • The TOE automatically configures references identifiers based on the FQDN or IP address configured used to specify the TLS server. The evaluator attempted a connection to the server with identifier “sqlserver.acumensec.local” resolved to 10.1.3.51 and ensured that the TOE could not connect to the server. • The evaluator observed the packet capture and ensured that the client does not complete the TLS handshake due to “certificate unknown” message. • The evaluator further observed the logs on the TOE located at /opt/HelpSystems/GoAnywhere/userdata/logs to confirm that the TOE could not connect to the server as The Common Alternative Name '*.local' is not allowed according to the strict hostname verification policy.
Pass/Fail with Explanation	Pass. The evaluator verified that the TLS connection fails when the server presented a certificate with wildcard in the left-most label of the presented identifier immediately preceding the public suffix while the reference identifier configured on the TOE contain two left most labels.

6.4.22 FCS_TLSC_EXT.1.3 Test #1

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that a server using a certificate with a valid certification path successfully connects. TD0513 has been applied.
Expected Results	As a part of FIA_X509_EXT.1.1 Test #1, • The evaluator attempted a connection from the TOE (TLS client) to the MySQL server which is presenting a certificate with a valid certificate path and confirmed that the TLS connection successfully established as the resource test was successful.

	• The evaluator observed the packet capture on the server and confirmed that the TLS handshake was successful with server presenting a certificate with a valid certification path. (Added a Note in FIA_X509_EXT.1.1 Test #1 which satisfied the current requirement)
Pass/Fail with Explanation	Pass. This test is covered as a part of FIA_X509_EXT.1.1 Test #1 where the evaluator demonstrated that a server using a certificate with a valid certification path results in a successful connection.

6.4.23 FCS_TLSC_EXT.1.3 Test #1b

Item	Data
Test Assurance Activity	The evaluator shall modify the certificate chain used by the server in test 1a to be invalid and demonstrate that a server using a certificate without a valid certification path to a trust store element of the TOE results in an authentication failure. TD0513 has been applied.
Expected Results	As a part of FIA_X509_EXT.1.1 Test #1, • The evaluator observed the packet capture on the server and confirmed that the TLS handshake was not successful as one of the issuing certificates in the certificate path is not a CA certificate due to which the TOE was unable to construct a valid certificate path which resulted in Certificate Unknown error returned to the server. • The evaluator further observed the debug logs on the TOE at /opt/HelpSystems/GoAnywhere/userdata/logs to confirm that the TOE was unable to construct a valid chain and the certification path could not be validated as the certificate presented was not a CA certificate. (Added a Note in FIA_X509_EXT.1.1 Test #1 which satisfied the current requirement)
Pass/Fail with Explanation	Pass. This test is covered as a part of FIA_X509_EXT.1.1 Test #1 where the evaluator demonstrated that modifying the certificate chain used by the server to be invalid resulted in an authentication failure as the TOE was unable to construct a valid certificate path reason being one of the issuing certificates in the certificate path is not a CA certificate.

6.4.24 FCS_TLSC_EXT.1.3 Test #1c

Item	Data
Test Assurance Activity	[conditional]: If the TOE trust store can be managed, the evaluator shall modify the trust store element used in Test 1a to be untrusted and demonstrate that a connection attempt from the same server used in Test 1a results in an authentication failure. TD0513 has been applied.
Expected Test Results	As a part of FIA_X509_EXT.1.1 Test #1, • The evaluator observed the packet capture on the server and confirmed that the TLS handshake was not successful as one of the issuing certificates in the certificate path is not present in the trust store due to which the TOE was unable to construct a valid certificate path which resulted in Certificate Unknown error returned to the server. • The evaluator further observed the debug logs on the TOE at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs to confirm that the TOE was

	unable to construct a valid chain as no issuer certificate in the trust store or the certification path was found. (Added a Note in FIA_X509_EXT.1.1 Test #1 which satisfied the current requirement)
Pass/Fail with Explanation	Pass. This test is covered as a part of FIA_X509_EXT.1.1 Test #1 where the evaluator demonstrated that modifying the trust store element to be untrusted (by deleting the Intermediate certificate from the trust store) and attempting a connection from the server resulted in an authentication failure as the TOE was unable to construct a valid certificate path reason being one of the issuing certificates in the certificate path is not present in the TOE's trust store.

6.4.25 FCS_TLSC_EXT.1.3 Test #2

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that using an invalid certificate (unless excepted) results in the function failing as follows, unless excepted: Test 2: The evaluator shall demonstrate that a server using a certificate which has been revoked results in an authentication failure.
Expected Results	As a part of FIA_X509_EXT.1.1 Test #3, • The evaluator observed the packet capture on the server and verified that the client fetched the CRLs required to validate the certificates and ensured to return a fatal alert to the server as the server certificate was revoked. • The evaluator further observed the debug logs on the TOE at /opt/HelpSystems/GoAnywhere/userdata/logs to confirm that the Certificate 82:F7:34:04:5D:C4:24:C3:64:0E:A1:16:2E:16:2B:04:0D:32:CB:C2 has been revoked by CRL at 'http://10.1.3.51/ICA2.crl' which corresponds to the server certificate. (Added a Note in FIA_X509_EXT.1.1 Test #3 which satisfied the current requirement)
Pass/Fail with Explanation	Pass. This test is covered as a part of FIA_X509_EXT.1.1 Test #3 where the evaluator demonstrated that a server using a certificate that was revoked resulted in an authentication failure.

6.4.26 FCS_TLSC_EXT.1.3 Test #3

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that using an invalid certificate (unless excepted) results in the function failing as follows, unless excepted: Test 3: The evaluator shall demonstrate that a server using a certificate which has passed its expiration date results in an authentication failure.
Expected Results	As a part of FIA_X509_EXT.1.1 Test #2, • The evaluator observed the packet capture on the server and confirmed that the TLS handshake was not successful with certificate unknown alert returned by the TOE. • The evaluator further observed the debug logs on the TOE at /opt/HelpSystems/GoAnywhere/userdata/logs to confirm that the server certificate expired on 20210212050000GMT+00:00 which indicates the function failing. (Added a Note in FIA_X509_EXT.1.1 Test #2 which satisfied the current requirement)

Pass/Fail with Explanation	Pass. This test is covered as a part of FIA_X509_EXT.1.1 Test #2 where the evaluator demonstrated that a server using a certificate which has passed its expiration date results in an authentication failure.

6.4.27 FCS_TLSC_EXT.1.3 Test #4

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that using an invalid certificate (unless excepted) results in the function failing as follows, unless excepted: Test 4: The evaluator shall demonstrate that a server using a certificate which does not have a valid identifier results in an authentication failure.
Expected Results	As a part of FCS_TLSC_EXT.1.2 Test #2, • The evaluator observed the packet capture and ensured that the client does not complete the TLS handshake due to "certificate unknown" message returned by the TOE. • The evaluator further observed the logs on the TOE located at /opt/HelpSystems/GoAnywhere/userdata/logs to confirm that the TOE could not connect to the server as there was no subject alternative name found matching IP address 10.1.3.51. (Added a Note in FCS_TLSC_EXT.1.2 Test #2 which satisfied the current requirement)
Pass/Fail with Explanation	Pass. This test is covered as a part of FCS_TLSC_EXT.1.2 Test #2 where the evaluator demonstrated that a server using a certificate which does not have a valid identifier in the SAN results in an authentication failure.

6.4.28 FCS_TLSC_EXT.2.1 Test #1

Item	Data
Test Assurance Activity	The evaluator shall establish a connection to a server that is not configured for mutual authentication (i.e. does not send Server's Certificate Request (type 13) message). The evaluator observes negotiation of a TLS channel and confirms that the TOE did not send Client's Certificate message (type 11) during handshake.
Test Steps	Note: As per the TSS, The TOE supports TLS mutual authentication for FTP/s, AS2, and HTTPS connections. The evaluator configured the TOE as a TLS client to establish a connection with the HTTPS server used for Remote File transfers. User File transfers via HTTPS on the GoAnywhere MFT is configured as Projects. The HTTPS server for these projects is linked via the server resource shown below • The evaluator created a client certificate for the TOE that was signed by the RootCA. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator created a key vault named "HTTPS" in encryption > Key Management system > Add key Valult. • The evaluator uploaded the client certificate in the key vault on the TOE. • The evaluator configured the TOE to reach 10.1.3.51 as the HTTPS server on port 445 in Resources > HTTPS Servers. • The evaluator selected the uploaded client certificate to be used for client authentication which ensured that mutual authentication must be performed as these options must be selected if client authentication is required.

	Note: The evaluator has chosen the tool as a remote server hence the evidence is limited to showing the successfully connection between the client and server. It is important to note this HTTPS connection is used for Remote file transfers on the TOE as stated in the TSS. • The evaluator used the "acumen-tlsc-pkg" tool to ensure that the server does not perform mutual authentication (i.e. does not send Server's Certificate Request (type 13) message) to establish a TLS connection with the TOE. The evaluator observed that the TLS connection was successful with application data received as seen in the tool output. • The evaluator observed the packet capture between the TOE and the server and ensured that the server did not send Server's Certificate Request (type 13) message and the TOE did not send the Client's Certificate message (type 11) during handshake.
Pass/Fail with Explanation	Pass. The evaluator observed the negotiation of a TLS channel and confirmed that the TOE did not send Client's Certificate message (type 11) during handshake when the server is not configured for mutual authentication.

6.4.29 FCS_TLSC_EXT.2.1 Test #2

Item	Data
Test Assurance Activity	The evaluator shall establish a connection to a server with a shared trusted root that is configured for mutual authentication (i.e. it sends Server's Certificate Request (type 13) message). The evaluator observes negotiation of a TLS channel and confirms that the TOE responds with a non-empty Client's Certificate message (type 11) and Certificate Verify (type 15) message.
Test Steps	Note: As per the TSS, The TOE supports TLS mutual authentication for FTP/s, AS2, and HTTPS connections. The evaluator configured the TOE as a TLS client to establish a connection with the HTTPS server used for Remote File transfers. User File transfers via HTTPS on the GoAnywhere MFT is configured as Projects. The HTTPS server for these projects is linked via the server resource shown below • The evaluator created a client certificate for the TOE that was signed by the RootCA. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator created a key vault named "HTTPS" in encryption > Key Management system > Add key Valult. • The evaluator uploaded the client certificate in the key vault on the TOE. • The evaluator configured the TOE to reach 10.1.3.51 as the HTTPS server on port 445 in Resources > HTTPS Servers. • The evaluator selected the uploaded client certificate to be used for client authentication which ensured that mutual authentication must be performed as these options must be selected if client authentication is required. Note: The evaluator has chosen the tool as a remote server hence the evidence is limited to showing the successfully connection between the client and server. It is important to note this HTTPS connection is used for Remote file transfers on the TOE as stated in the TSS. • The evaluator used the "acumen-tlsc-pkg" tool to ensure that the server supports mutual authentication. The evaluator observed that the TLS connection

	was successful with mutual authentication and the application data received as seen in the tool output. • The evaluator observed the packet capture between the TOE and the server and ensured that the server sent a Server's Certificate Request (type 13) message confirms that the TOE responds with a non-empty Client's Certificate message (type 11) and Certificate Verify (type 15) message.
Pass/Fail with Explanation	Pass. The evaluator observed the negotiation of a TLS channel and confirmed that the TOE responds with a non-empty Client's Certificate message (type 11) and Certificate Verify (type 15) message when the server is configured for mutual authentication.

6.4.30 FCS_TLSC_EXT.5.1 Test #1

Item	Data
Test Assurance Activity	The evaluator shall configure a server to perform key exchange using each of the TOE's supported curves and/or groups. The evaluator shall verify that the TOE successfully connects to the server.
Test Steps	• The evaluator imported the RootCA_ec certificate into the keyvault of the TOE that signed the ec server certificate. • The evaluator created a server ec certificate using the XCA tool. • The evaluator uploaded the server certificate and the server key to the Database server. • The evaluator used the "acumen-tlsc-mysql" tool to attempt a TLS connection using each of the supported elliptic curves. The evaluator observed that the TLS connection was successful using secp256r1 and secp384r1 and the application data received as seen in the tool output. • The evaluator observed the packet capture and ensured that the TLS handshake was successful using secp256r1 curve. • The evaluator observed the packet capture and ensured that the TLS handshake was successful using secp384r1 curve.
Pass/Fail with Explanation	Pass. The evaluator verified that the TOE successfully connects to the server with each of its supported curves when the server is configured to perform key exchange using each of the TOE's supported curves and/or groups.

6.5 PKG_TLSS (Linux)

6.5.1 FCS_TLSS_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	The evaluator shall establish a TLS connection using each of the cipher suites specified by the requirement. This connection may be established as part of the establishment of a higher-level protocol, e.g., as part of an EAP session. It is sufficient to observe the successful negotiation of a cipher suite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic in an attempt to discern the cipher suite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).
Test Steps	• The evaluator created a server certificate that was signed by the Root CA using the XCA tool. • The evaluator ensured that the server certificate was signed by the RootCA.

	• The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate.
	• The evaluator loaded the server certificate (https_server_253.pem) along with the key in pem format to the TOE's system keyvault.
	• The evaluator configured the TOE's Administration server in System > Admin server where the port was set 8001 in General.
	• The evaluator selected the TLS protocol, supported cipher suites and the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and saved the configuration.
	• The evaluator used the openssl s_client resource to establish the TLS connection with each of the cipher suite supported by the TOE as server and verified the connection to be successful.
	  ▪ TLS_RSA_WITH_AES_128_CBC_SHA
	  ▪ TLS_RSA_WITH_AES_256_CBC_SHA
	  ▪ TLS_RSA_WITH_AES_128_CBC_SHA256
	  ▪ TLS_RSA_WITH_AES_256_CBC_SHA256
	  ▪ TLS_RSA_WITH_AES_128_GCM_SHA256
	  ▪ TLS_RSA_WITH_AES_256_GCM_SHA384
	  ▪ TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
	  ▪ TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
	• The evaluator observed the packet capture and ensured that the TLS connection was successful where the server responded with the cipher suite configured on the client side.
	
	• The evaluator created a server ec certificate that was signed by the Root CA_ec using the XCA tool.
	• The evaluator ensured that the server ec certificate was signed by the RootCA_ec.
	• The evaluator imported the RootCA_ec certificate into the key vault of the TOE that signed the server certificate.
	• The evaluator loaded the server certificate (https_server_ec_253.pem) along with the key in pem format to the TOE's system keyvault.
	• The evaluator configured the TOE's Administration server in System > Admin server where the port was set 8001 in General.
	• The evaluator selected the TLS protocol, supported cipher suites and the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and saved the configuration.
	• The evaluator used the openssl s_client resource to establish the TLS connection with each of the cipher suite supported by the TOE as server and verified the connection to be successful.
	  ▪ TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
	  ▪ TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
	  ▪ TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
	  ▪ TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384

	• The evaluator observed the packet capture and ensured that the TLS connection was successful where the server responded with the cipher suite configured on the client side.
Pass/Fail with Explanation	Pass. The TOE successfully connected with all claimed algorithms. This meets testing requirements.

6.5.2 FCS_TLSS_EXT.1.1 Test #2

Item	Data
Test Assurance Activity	The evaluator shall send a Client Hello to the server with a list of cipher suites that does not contain any of the cipher suites in the server's ST and verify that the server denies the connection. Additionally, the evaluator shall send a Client Hello to the server containing only the TLS_NULL_WITH_NULL_NULL cipher suite and verify that the server denies the connection.
Test Steps	• The evaluator created a server certificate that was signed by the Root CA using the XCA tool. • The evaluator ensured that the server certificate was signed by the RootCA. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator loaded the server certificate (https_server_253.pem) along with the key to the TOE's system keyvault. • The evaluator configured the TOE's Administration server in System > Admin server where the port was set 8001 in General. • The evaluator selected the TLS protocol, supported cipher suites and the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and saved the configuration. • The evaluator used the "acumen-tlss-pkg" tool to send a Client Hello to the server with a cipher suite that is not present in the list of the cipher suites claimed in the server's ST and verified that the server denies the connection. Additionally, the evaluator sent a Client Hello to the server using the tool containing only the TLS_NULL_WITH_NULL_NULL cipher suite and verified that the server denied the connection. • The evaluator observed the packet capture (packet 1-2) and ensured that when the client hello was sent to the server with a cipher suite (TLS_RSA_WITH_NULL_MD5) that is not present in the list of the cipher suites claimed in the server's ST, the server denied the connection due to Handshake failure. • The evaluator observed the packet capture (packet 3-4) and ensured that when the client hello was sent to the server with a TLS_NULL_WITH_NULL_NULL cipher suite, the server denied the connection due to Handshake failure.
Pass/Fail with Explanation	Pass. The TOE rejected a connection when the Client Hello consists of a cipher not claimed in the ST or a NULL cipher. This meets testing requirements.

6.5.3 FCS_TLSS_EXT.1.1 Test #3

Item	Data

Test Assurance Activity	If **RSA key exchange is used in one of the selected cipher suites**, the evaluator shall use a client to send a properly constructed Key Exchange message with a modified EncryptedPreMasterSecret field during the TLS handshake. The evaluator shall verify that the handshake is not completed successfully, and no application data flows.
Test Steps	• The evaluator created a server certificate that was signed by the Root CA using the XCA tool. • The evaluator ensured that the server certificate was signed by the RootCA. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator loaded the server certificate (https_server_253.pem) along with the key to the TOE's system keyvault. • The evaluator configured the TOE's Administration server in System > Admin server where the port was set 8001 in General. • The evaluator selected the TLS protocol, supported cipher suites and the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and saved the configuration. • The evaluator used the "acumen-tlss-pkg" as a TLS client to send a properly constructed Key Exchange message with a modified EncryptedPreMasterSecret field during the TLS handshake and verified that the handshake is not completed successfully, and no application data flows as per the tool output. • The evaluator observed the packet capture on the client to ensure that the handshake is not completed successfully, and no application data flows when the TLS client sent a properly constructed Key Exchange message with a modified EncryptedPreMasterSecret field during the TLS handshake.
Pass/Fail with Explanation	Pass. The TOE rejects the connection when the client sends a properly constructed key exchange message but a modified EncryptedPreMasterSecret field. This meets the testing requirements.

6.5.4 FCS_TLSS_EXT.1.1 Test #4.2

Item	Data
Test Assurance Activity	Modify a byte in the data of the client's Finished handshake message, and verify that the server rejects the connection and does not send any application data.
Test Steps	• The evaluator created a server certificate that was signed by the Root CA using the XCA tool. • The evaluator ensured that the server certificate was signed by the RootCA. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator loaded the server certificate (https_server_253.pem) along with the key to the TOE's system keyvault. • The evaluator configured the TOE's Administration server in System > Admin server where the port was set 8001 in General. • The evaluator selected the TLS protocol, supported cipher suites and the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and saved the configuration. • The evaluator used the acumen-tlss-pkg tool as a TLS client to connect to the server with a modified client's Finished handshake message and verified that the

	server rejects the connection due to “DECRYPT _ERROR” alert as seen in the tool output. • The evaluator observed the packet capture on the client and ensured that the server rejects the connection due to “DECRYPT _ERROR” alert after the client attempted to connect to the server with a modified client’s Finished handshake message.
Pass/Fail with Explanation	Pass. The TOE successfully rejects the connection after receiving the finished message before receiving the change cipher spec message. This meets testing requirements.

6.5.5 FCS_TLSS_EXT.1.1 Test #4.3

Item	Data
Test Assurance Activity	Demonstrate that the TOE will not resume a session for which the client failed to complete the handshake (independent of TOE support for session resumption): **Test 4.3i [conditional]:** If the TOE does not support session resumption based on session IDs according to RFC4346 (TLS1.1) or RFC5246 (TLS1.2) or session tickets according to RFC5077, the evaluator shall perform the following test: a) The evaluator shall send a Client Hello with a zero-length session identifier and with a SessionTicket extension containing a zero-length ticket. b) The evaluator shall verify the server does not send a NewSessionTicket handshake message (at any point in the handshake). c) The evaluator shall verify the Server Hello message contains a zero-length session identifier or passes the following steps: Note: The following steps are only performed if the ServerHello message contains a non-zero length SessionID. d) The evaluator shall complete the TLS handshake and capture the SessionID from the ServerHello. e) The evaluator shall send a ClientHello containing the SessionID captured in step d). This can be done by keeping the TLS session in step d) open or start a new TLS session using the SessionID captured in step d). f) The evaluator shall verify the TOE (1) implicitly rejects the SessionID by sending a ServerHello containing a different SessionID and by performing a full handshake (as shown in Figure 1 of RFC 4346 or RFC 5246), or (2) terminates the connection in some way that prevents the flow of application data. **TD0588 Applied**
Test Steps	• The evaluator created a server certificate that was signed by the Root CA using the XCA tool. • The evaluator ensured that the server certificate was signed by the RootCA. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator loaded the server certificate (https_server_253.pem) along with the key to the TOE’s system keyvault. • The evaluator configured the TOE’s Administration server in System > Admin server where the port was set 8001 in General.

	• The evaluator selected the TLS protocol, supported cipher suites and the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and saved the configuration. • The evaluator used the acumen-tlss-pkg tool to generate a Fatal alert during the TLS handshake from the client before the client sends a ChangeCipherSpec message and then sent a Client Hello with the session identifier from the previous incomplete session and verified that the server does not resume the previous dead session. The connection succeeded as per the tool output as the TOE does not set a session ID which implies that there was no dead session. • The evaluator observed the packet capture (packet 1-5) and observed the Fatal alert during the TLS handshake from the client before the client sends a ChangeCipherSpec message. • The evaluator then observed the packet capture (packet 6-13) to ensure that the connection succeeded as that the TOE does not set a session ID which implies that there was no dead session. • The evaluator ensured that a client hello was sent with a zero-length session identifier and with a SessionTicket extension containing a zero-length ticket as highlighted below (which implies that the client is not specifying a session to resume). • The evaluator also verified through the packet capture that the server did not send a NewSessionTicket handshake message (at any point in the handshake) which implies that the server does not support Session tickets. • The evaluator verified that the Server Hello message contains a zero-length session identifier as seen in packet 2 which confirms that the TOE does not support session resumption using session IDs.
Pass/Fail with Explanation	Pass. The TOE does not send a session ID when attempted to connect with the previous incomplete session which implies there are no dead sessions. The TOE sends a Server Hello message containing a zero-length session identifier in response to a Client Hello with a zero-length session identifier which implies that the TOE does not support session resumption using session IDs. The TOE sends oes not send a NewSessionTicket handshake message (at any point in the handshake) which implies that the TOE does not support session resumption using Session tickets. This meets the testing requirements.

6.5.6 FCS_TLSS_EXT.1.1 Test #4.4

Item	Data
Test Assurance Activity	Send a message consisting of random bytes from the client after the client has issued the ChangeCipherSpec message and verify that the server denies the connection.
Test Steps	• The evaluator created a server certificate that was signed by the Root CA using the XCA tool. • The evaluator ensured that the server certificate was signed by the RootCA. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator loaded the server certificate (https_server_253.pem) along with the key to the TOE's system keyvault. • The evaluator configured the TOE's Administration server in System > Admin server where the port was set 8001 in General.

	• The evaluator selected the TLS protocol, supported cipher suites and the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and saved the configuration. • The evaluator used the acumen-tlss-pkg tool to attempt a connection to the TOE with a TLS connection and sent a message ("this is a garbled message") from the client after the client has issued the ChangeCipherSpec message and verified that a fatal alert was returned by the server. • The evaluator observed the packet capture and ensured that the client sent a random data which appears as application data before sending the finished message and ensured that the server returned an alert message.
Pass/Fail with Explanation	Pass. The TOE successfully rejects a connection when receiving a garbled message after the ChangeCipherSpec message. This meets testing requirements.

6.5.7 FCS_TLSS_EXT.1.2 Test #1

Item	Data
Test Assurance Activity	The evaluator shall send a Client Hello requesting a connection with version SSL 2.0 and verify that the server denies the connection. The evaluator shall repeat this test with SSL 3.0 and TLS 1.0, and TLS 1.1 if it is selected.
Test Steps	• The evaluator created a server certificate that was signed by the Root CA using the XCA tool. • The evaluator ensured that the server certificate was signed by the RootCA. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator loaded the server certificate (https_server _253.pem) along with the key to the TOE's system keyvault. • The evaluator configured the TOE's Administration server in System > Admin server where the port was set 8001 in General. • The evaluator selected the TLS protocol, supported cipher suites and the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and saved the configuration. • The evaluator used the acumen-tlss-pkg tool to initiate the TLS connection with SSL 3.0, SSL 2.0, TLS 1.0, TLS 1.1 and verified that the server returned a fatal alert for each non-supported TLS versions. • The evaluator observed the packet capture on the client and ensured that the server returned a fatal alert when the client attempted a connection with SSL 3.0 version. • The evaluator observed the packet capture on the client and ensured that the server reset the connection with a FIN packet and a reset packet when the client attempted a connection with SSL 2.0 version. • The evaluator observed the packet capture on the client and ensured that the server returned a fatal alert when the client attempted a connection with TLS 1.1 version. • The evaluator observed the packet capture on the client and ensured that the server returned a fatal alert when the client attempted a connection with TLS 1.0 version.

Pass/Fail with Explanation	Pass. The evaluator ensured that the server returned a fatal alert or reset packet when the client attempted to connect with an unsupported TLS version in the client hello. This meets the testing requirements.

6.5.8 FCS_TLSS_EXT.1.3 Test #1

Item	Data
Test Assurance Activity	Note that this testing can be accomplished in conjunction with other testing activities. For each of the following tests, determining that the size matches the expected size is sufficient. **Test 1:** [conditional] If **RSA-based key establishment is selected**, the evaluator shall configure the TOE with a certificate containing a supported RSA size and attempt a connection. The evaluator shall verify that the size used matches that which is configured and that the connection is successfully established. The evaluator shall repeat this test for each supported size of RSA-based key establishment.
Test Steps	• The evaluator created a server certificate containing a supported RSA size of 2048 bit that was signed by the Root CA using the XCA tool. • The evaluator ensured that the server certificate was containing the RSA size of 2048 bit and was signed by the RootCA. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator loaded the server certificate (https_server2048.pem) along with the key to the TOE's system keyvault. • The evaluator configured the TOE's Administration server in System > Admin server where the port was set 8001 in General. • The evaluator configured the TLS protocol, supported cipher suites and the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and saved the configuration. • The evaluator initiated a TLS connection using the openssl s_client resource to the TOE and ensured that the TLS handshake was successful with RSA key size of 2048 bit. • The evaluator observed the packet capture on the client and verified that the connection established with RSA size used (256*8=2048 bit). • The evaluator created a server certificate containing a supported RSA size of 3072 bit that was signed by the Root CA using the XCA tool. • The evaluator ensured that the server certificate was containing the RSA size of 3072 bit and was signed by the RootCA. • The evaluator loaded the server certificate (https_server3072.pem) along with the key to the TOE's system keyvault. • The evaluator configured the TLS protocol, supported cipher suites and the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and saved the configuration. • The evaluator initiated a TLS connection using the openssl s_client resource to the TOE and ensured that the TLS handshake was successful with RSA key size of 3072 bit. • The evaluator observed the packet capture on the client and verified that the connection established with RSA size used (384*8=3072 bit).

	• The evaluator created a server certificate containing a supported RSA size of 4096 bit that was signed by the Root CA using the XCA tool. • The evaluator ensured that the server certificate was containing the RSA size of 4096 bit and was signed by the RootCA. • The evaluator loaded the server certificate (https_server4096.pem) along with the key to the TOE's system keyvault. • The evaluator configured the TLS protocol, supported cipher suites and the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and saved the configuration. • The evaluator initiated a TLS connection using the openssl s_client resource to the TOE and ensured that the TLS handshake was successful with RSA key size of 4096 bit. • The evaluator observed the packet capture on the client and verified that the connection established with RSA size used (512*8=4096 bit).
Pass/Fail with Explanation	Pass. The evaluator verified that the TLS connection is successfully established with each supported RSA key size of 2048,3072 and 4096 bits. This meets the testing requirements.

6.5.9 FCS_TLSS_EXT.1.3 Test #3

Item	Data
Test Assurance Activity	Note that this testing can be accomplished in conjunction with other testing activities. For each of the following tests, determining that the size matches the expected size is sufficient. [conditional] *If ECDHE ciphers are selected, the evaluator shall attempt a connection using an ECDHE cipher suite with a supported curve. The evaluator shall verify that the key agreement parameters in the Key Exchange message are the ones configured. The evaluator shall repeat this test for each supported elliptic curve.*
Test Steps	• The evaluator created a server certificate that was signed by the Root CA using the XCA tool. • The evaluator ensured that the server certificate was signed by the RootCA. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator loaded the server certificate (https_server_253.pem) along with the key to the TOE's system keyvault. • The evaluator configured the TOE's Administration server in System > Admin server where the port was set 8001 in General. • The evaluator selected the TLS protocol, supported cipher suites and the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and saved the configuration. • The evaluator attempted a connection to the TOE using an ECDHE cipher suite (ECDHE-RSA-AES128-GCM-SHA256) with supported EC-DH curve (secp256r1) and verified that the TLS connection was successful. • The evaluator observed the packet capture on the client to ensure that TLS connection was successfully established with the TOE using an ECDHE cipher suite (ECDHE-RSA-AES128-GCM-SHA256) with supported EC-DH curve

	(secp256r1) and also confirmed that key agreement parameters in the key exchange are the ones configured. • The evaluator attempted a connection to the TOE using an ECDHE cipher suite (ECDHE-RSA-AES128-GCM-SHA256) with supported EC-DH curve (secp384r1) and verified that the TLS connection was successful. • The evaluator observed the packet capture on the client to ensure that TLS connection was successfully established with the TOE using an ECDHE cipher suite (ECDHE-RSA-AES128-GCM-SHA256) with supported EC-DH curve (secp384r1) and also confirmed that key agreement parameters in the key exchange are the ones configured.
Pass/Fail with Explanation	Pass. The evaluator verified that the TOE establishes a successful TLS connection with all supported EC-DH curves. This meets the testing requirements.

6.5.10 FCS_TLSS_EXT.2.2 Test #1

Item	Data
Test Assurance Activity	The evaluator shall configure the server to send a certificate request to the client. The client shall send a certificate_list structure which has a length of zero. The evaluator shall verify that the handshake is not finished successfully, and no application data flows.
Test Steps	• The evaluator created a server certificate that was signed by the Root CA using the XCA tool. • The evaluator ensured that the server certificate was signed by the RootCA. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator loaded the server certificate (https_server_253.pem) along with the key to the TOE's system keyvault. • The evaluator configured the TOE's Administration server in System > Admin server where the port was set 8001 in General. • The evaluator selected the TLS protocol, supported cipher suites, the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and also configured the server to send a client request by selecting the Client authentication as required and saved the configuration. • The evaluator created a client certificate (10.1.3.51_httpsclient.crt) and the client key (client_key.pem) using XCA tool. • The evaluator used the acumen-tlss-pkg tool as a client to initiate a TLS connection and send a certificate list structure which has a length of zero. The evaluator ensured with the tool output that a fatal alert was returned by the server. • The evaluator observed the packet capture and ensured that the TLS handshake did not finish successfully when the client sent a certificate list structure which has a length of zero.
Pass/Fail with Explanation	Pass. The TOE does not establish a TLS connection with the client when the client sends a certificate list structure which has a length of zero. This meets the testing requirements.

6.5.11 FCS_TLSS_EXT.2.2 Test #2

Item	Data
Test Assurance Activity	The evaluator shall configure the server to send a certificate request to the client. The client shall send no client certificate message, and instead send a client key exchange message in an attempt to continue the handshake. The evaluator shall verify that the handshake is not finished successfully, and no application data flows.
Test Steps	• The evaluator created a server certificate that was signed by the Root CA using the XCA tool. • The evaluator ensured that the server certificate was signed by the RootCA. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator loaded the server certificate (https_server_253.pem) along with the key to the TOE's system keyvault. • The evaluator configured the TOE's Administration server in System > Admin server where the port was set 8001 in General. • The evaluator selected the TLS protocol, supported cipher suites, the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and also configured the server to send a client request by selecting the Client authentication as required and saved the configuration. • The evaluator created a client certificate (10.1.3.51_httpsclient.crt) and the client key (client_key.pem) using XCA tool. • The evaluator used the acumen-tlss-pkg tool to attempt a TLS connection with the server by not sending a client certificate message, and instead send a client key exchange message in an attempt to continue the handshake. The evaluator verified that a fatal alert was returned by the server. • The evaluator observed the packet capture and ensured that the TLS handshake did not finish successfully when the client does not send a client certificate message, and instead send a client key exchange message in an attempt to continue the handshake.
Pass/Fail with Explanation	Pass. The evaluator verified that the TLS handshake did not finish successfully and the server returned a fatal alert when the client does not send a client certificate message, and instead send a client key exchange message in an attempt to continue the handshake. This meets the testing requirements.

6.5.12 FCS_TLSS_EXT.2.2 Test #3

Item	Data
Test Assurance Activity	The evaluator shall configure the server to send a certificate request to the client without the supported_signature_algorithm used by the client's certificate. The evaluator shall attempt a connection using the client certificate and verify that the handshake is not finished successfully, and no application data flows.
Test Steps	• The evaluator created a server certificate that was signed by the Root CA using the XCA tool. • The evaluator ensured that the server certificate was signed by the RootCA. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate.

	• The evaluator loaded the server certificate (https_server_253.pem) along with the key to the TOE's system keyvault. • The evaluator configured the TOE's Administration server in System > Admin server where the port was set 8001 in General. • The evaluator selected the TLS protocol, supported cipher suites, the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and also configured the server to send a client request by selecting the Client authentication as required and saved the configuration. • The evaluator created a client certificate (10.1.3.51_httpsclient.crt) and the client key (client_key.pem) using XCA tool. • The evaluator used the acumen-tlss-pkg tool to attempt a TLS connection with server and modify the signature algorithm used by the client's certificate to an unsupported signature algorithm (RSA_MD5) and verified that a fatal alert was returned by the server. • The evaluator observed the packet capture on the client and ensured that the TLS handshake did not finish successfully when the server received a client certificate with unsupported signature algorithm.
Pass/Fail with Explanation	Pass. The evaluator verified that the TLS handshake did not finish successfully when the server received a client certificate with unsupported signature algorithm.

6.5.13 FCS_TLSS_EXT.2.2 Test #4

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that using a certificate without a valid certification path results in the function failing. Using the administrative guidance, the evaluator shall then load a certificate or certificates needed to validate the certificate to be used in the function and demonstrate that the function succeeds. The evaluator then shall delete one of the certificates and show that the function fails.
Test Steps	• The evaluator created a RootCA certificate that signed the server certificate (https_server) and also created a RootCA_client certificate that signed the client certificate using the XCA tool. • The evaluator ensured that the server certificate was signed by the RootCA. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator loaded the server certificate (https_server_253.pem) along with the key to the TOE's system keyvault. • The evaluator configured the TOE's Administration server in System > Admin server where the port was set 8001 in General. • The evaluator selected the TLS protocol, supported cipher suites, the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and saved the configuration. • The evaluator confirmed that the RootCA_client certificate that signed the client certificate was not uploaded to the TOE's system keyvault. • The evaluator uploaded the client certificate to the client VM and ensured that the certificate was signed by the RootCA_client certificate.

	• The evaluator initiated the TLS connection using the client certificate that was signed by the RootCA_client certificate which was not present in the TOE's trust store (system keyvault) and ensured that the TLS connection did not succeed due to certificate unknown error. • The evaluator observed the packet capture to ensure that the TLS handshake was not successful due to certificate unknown alert returned by the server after the client sent the certificate. • Note: A TLS handshake is considered to be successful when the server can validate the client certificate presented in the TLS handshake with the Client's Root certificate imported in the server's trust store. This results in exchange of application data packets. • The evaluator further observed the logs located at /opt/HelpSystems/GoAnywhere/userdata/logs to ensure that the TLS handshake did not complete as the TOE was unable to construct a valid chain and no issuer certificate for the client certificate in the certificate was found. • The evaluator then loaded the RootCA_client certificate authority needed to validate the client certificate to the TOE's trust store (system keyvault) in Encryption > Key Management services. • The evaluator confirmed that the certificate authority was imported successfully. • The evaluator initiated the TLS connection using the client certificate that was signed by the RootCA_client certificate which was imported in the TOE's trust store (system keyvault) and ensured that the TLS connection established with SSL session observed in the terminal output with no errors observed validating the client certificate presented. • The evaluator observed the packet capture on the client VM to confirm that the TLS connection established with Application Data sent to the server indicating that the Hanshake was successful. • Note: A TLS handshake is considered to be successful when the server can validate the client certificate presented in the TLS handshake with the Client's Root certificate imported in the server's trust store. This results in exchange of application data packets. • The evaluator then deleted the RootCA_client certificate authority from the TOE's certificate trust store (system keyvault) that signed the client's certificate. • The evaluator confirmed that the RootCA_client certificate authority was not present in the TOE's trust store. • The evaluator initiated the TLS connection using the client certificate that was signed by the RootCA_client certificate which was not present in the TOE's trust store (system keyvault) and ensured that the TLS connection did not complete due to certificate unknown error. • The evaluator observed the packet capture to ensure that the TLS handshake was not successful due to certificate unknown alert returned by the server after the client sent the certificate. • Note: A TLS handshake is considered to be successful when the server can validate the client certificate presented in the TLS handshake with the Client's Root certificate imported in the server's trust store. This results in exchange of application data packets.

	• The evaluator further observed the logs located at /opt/HelpSystems/GoAnywhere/userdata/logs to ensure to ensure that the TLS handshake did not complete as the TOE was unable to construct a valid chain and no issuer certificate for the client certificate in the certificate was found.
Pass/Fail with Explanation	Pass. The evaluator verified that when the server was presented with a client certificate without having its issuer certificate in the TOE's trust store or the certification path, the server failed to validate the client certificate which resulted in function failing. This meets the testing requirements.

6.5.14 FCS_TLSS_EXT.2.2 Test #5

Item	Data
Test Assurance Activity	The aim of this test is to check the response of the server when it receives a client identity certificate that is signed by an impostor CA (either Root CA or intermediate CA). To carry out this test the evaluator shall configure the client to send a client identity certificate with an issuer field that identifies a CA recognized by the TOE as a trusted CA, but where the key used for the signature on the client certificate does not in fact correspond to the CA certificate trusted by the TOE (meaning that the client certificate is invalid because its certification path does not in fact terminate in the claimed CA certificate). The evaluator shall verify that the attempted connection is denied.
Test Steps	• The evaluator created a RootCA certificate authority with key RootCA_key that was used to sign the client identity certificate. • The evaluator verified the issuer filed to be CN=RootCA. • The evaluator uploaded the RootCA certificate authority to the TOE's trust store (system keyvault) and ensured that the TOE recognizes the CA with "RootCA" in the issuer field. • The evaluator created a RootCA_imposter certificate authority with the same issuer field "RootCA" that was previously trusted on the TOE but with a different key RootCA_imposter_key and used this certificate authority to sign the client identity certificate. • The evaluator verified the issuer filed to be CN=RootCA that is same as the one trusted by the TOE. • The evaluator uploaded the client identity certificate that was signed by the RootCA_imposter certificate authority to the TLS client VM and ensured it to have an issuer field "CN=RootCA" that identifies the RootCA recognised by the TOE as a trusted CA. • The evaluator initiated the TLS connection using the client identity certificate signed by the Imposter Certificate authority and ensure that the TLS handshake failed due to certificate unknown error returned by the server. • The evaluator observed the packet capture to ensure that the TLS handshake failed due to certificate unknown error returned by the server after receiving the client identity certificate that was signed by the imposter CA. • The evaluator observed the logs on the TOE at /opt/HelpSystems/GoAnywhere/userdata/logs and ensured that the TLS handshake was not successful as the TOE was unable to construct a valid chain

	and further confirmed that the TOE found the trust anchor but the certificate validation failed as the certificate does not verify with the supplied key.
Pass/Fail with Explanation	Pass. The evaluator ensured the TOE to respond with a certificate unknown error when it received a client identity certificate that is signed by an impostor CA. This meets the testing requirements.

6.5.15 FCS_TLSS_EXT.2.2 Test #6

Item	Data
Test Assurance Activity	The evaluator shall configure the client to send a certificate with the Client Authentication purpose in the extendedKeyUsage field and verify that the server accepts the attempted connection. The evaluator shall repeat this test without the Client Authentication purpose and shall verify that the server denies the connection. Ideally, the two certificates should be identical except for the Client Authentication purpose.
Test Steps	• The evaluator created a server certificate that was signed by the Root CA using the XCA tool. • The evaluator ensured that the server certificate was signed by the RootCA. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator loaded the server certificate (https_server_253.pem) along with the key to the TOE's system keyvault. • The evaluator configured the TOE's Administration server in System > Admin server where the port was set 8001 in General. • The evaluator selected the TLS protocol, supported cipher suites, the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and also configured the server to send a client request by selecting the Client authentication as required and saved the configuration. • The evaluator created a client certificate that was signed by the RootCA that was previously uploaded to the TOE's trust store using XCA tool. • The evaluator uploaded the certificate to the Client VM and ensured the client certificate that is being used for a TLS connection contain Client Authentication purpose in the extendedKeyUsage extension. • The evaluator initiated a connection using the openssl s_client resource using the client certificate uploaded and ensured that the TLS handshake was successful. • The evaluator observed the packet capture on the client and confirmed the client certificate to have Client Authentication purpose in the extendedKeyUsage extension and the TLS handshake was successful. • The evaluator created an identical certificate that is similar in structure using xca, the types of identifiers used, and the chain of trust but lacks the Client Authentication purpose in the extendedKeyUsage extension. • The evaluator uploaded the client certificate to the client VM and ensured it to be an identical certificate that is similar in structure, the types of identifiers used, and the chain of trust but lacks the Client Authentication purpose in the extendedKeyUsage extension.

	• The evaluator initiated a TLS connection using the certificate and ensured that the connection was not successful as the server returned a certificate unknown error. • The evaluator observed the packet capture on the client and ensured that TLS handshake was not successful and the server returned the unknown certificate error after receiving the client certificate that lacks Client Authentication purpose in the extendedKeyUsage extension. • The evaluator observed the logs on the TOE at /opt/HelpSystems/GoAnywhere/userdata/logs and ensured the TLS handshake was not successful due to invalid client extended key usage.
Pass/Fail with Explanation	Pass. The TOE had a successful connection when the certificate did contain the client authentication purpose. The TOE also denied a connection when the certificate did not contain the client authentication purpose. This meets testing requirements.

6.5.16 FCS_TLSS_EXT.2.2 Test #7(a)

Item	Data
Test Assurance Activity	Configure the server to require mutual authentication and then modify a byte in the client's certificate. The evaluator shall verify that the server rejects the connection.
Test Steps	• The evaluator created a server certificate that was signed by the Root CA using the XCA tool. • The evaluator ensured that the server certificate was signed by the RootCA. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator loaded the server certificate (https_server_253.pem) along with the key to the TOE's system keyvault. • The evaluator configured the TOE's Administration server in System > Admin server where the port was set 8001 in General. • The evaluator selected the TLS protocol, supported cipher suites, the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and also configured the server to send for mutual authentication by selecting the Client authentication as required and saved the configuration. • The evaluator used the acumen-tlss-pkg tool to initiate a TLS connection to the server with the client certificate and modify the last byte of the client's certificate during the handshake and confirmed that a Fatal alert was returned by the server. • The evaluator observed the packet capture and ensured that the last byte in the client's certificate was modified from ad to 41 and ensured that a fatal alert "certificate unknown" was returned by the server.
Pass/Fail with Explanation	Pass. The evaluator ensured that the server returns a fatal alert to the client and the TLS handshake fails when it receives a client certificate with a modified byte.

6.5.17 FCS_TLSS_EXT.2.2 Test #7(b)

Item	Data

Test Assurance Activity	Configure the server to require mutual authentication and then modify a byte in the signature block of the client's Certificate Verify handshake message. The evaluator shall verify that the server rejects the connection.
Test Steps	• The evaluator created a server certificate that was signed by the Root CA using the XCA tool. • The evaluator ensured that the server certificate was signed by the RootCA. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator loaded the server certificate (https_server_253.pem) along with the key to the TOE's system keyvault. • The evaluator configured the TOE's Administration server in System > Admin server where the port was set 8001 in General. • The evaluator selected the TLS protocol, supported cipher suites, the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and also configured the server to send for mutual authentication by selecting the Client authentication as required and saved the configuration. • The evaluator used the acumen-tlss-pkg tool to initiate a TLS connection to the server with the client certificate and modify 8 bytes in the signature block of the client's Certificate Verify handshake message during the handshake and confirmed that a Fatal alert was returned by the server. • The evaluator observed the packet capture and ensured that the 8 bytes in the signature block of the client's Certificate Verify handshake message were modified as per the tool output and confirmed that a fatal alert was returned by the server.
Pass/Fail with Explanation	Pass. The evaluator ensured that the server returns a fatal alert to the client and the TLS handshake fails when it modified signature block in the client's Certificate Verify handshake message.

6.5.18 FCS_TLSS_EXT.2.3 Test #1

Item	Data
Test Assurance Activity	The evaluator shall send a client certificate with an identifier that does not match any of the expected identifiers and verify that the server denies the connection. The matching itself might be performed outside the TOE (e.g. when passing the certificate on to a directory server for comparison).
Test Steps	Note: The TOE does the identifier check in the client certificate when the admin users authenticate using certificate to login to the TOE. • Certificates authenticating Admin Users can be verified by the Admin User's username, email address, or both. The Admin User's username is checked against the Subject Distinguished Name (DN) common name (CN) for a match • The evaluator created a server certificate that was signed by the Root CA using the XCA tool. • The evaluator ensured that the server certificate was signed by the RootCA. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator loaded the server certificate (https_server_253.pem) along with the key to the TOE's system keyvault.

	• The evaluator configured the TOE's Administration server in System > Admin server where the port was set 8001 in General. • The evaluator selected the TLS protocol, supported cipher suites, the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and also configured the server to send for mutual authentication by selecting the Client authentication as required and saved the configuration. • The evaluator created a client certificate for the admin user "test" with an incorrect CN/SAN that was signed by the RootCA trusted on the TOE. • The evaluator uploaded the certificate to the client VM in p12 format and ensured it to have an incorrect CN=test_incorrect. • The evaluator created an admin user named "test" with authentication type set to certificate on the TOE and provided the SHA1 fingerprint of the client certificate that is being presented. The evaluator selected the expected identifier to be the username (test) in SAN/DN validation. • The evaluator ensured that the admin user test was created with authentication type set to certificate and username was selected for SAN/DN validation. • The evaluator installed the client certificate incorrect CN=test_incorrect in the Firefox browser that will be used to login to the TOE. • The evaluator attempted to login to the TOE using the Firefox browser with client certificate installed and ensured the login attempt was not successful using the certificate with incorrect CN and the TOE reverted back requesting for authentication credentials. Note: If a matching certificate is found during the connection, the Admin User will automatically authenticate. However, if a match is not found, the Admin User can still login to the Go Anywhere server with a username and password. • The evaluator observed the logs on the TOE and ensured that the admin user "test" failed to login. • The evaluator further observed the debug logs at /opt/HelpSystems/GoAnywhere/userdata/logs and ensured that the digital certificate that was used for client authentication was invalid. • The evaluator attempted to connect to the TOE with the certificate and verified that the certificate was deemed invalid as identifier in the CN did not match the expected identifier "username". However, the TLS connection is interpreted to be the application layer connection (i.e., administrator GUI connection) and reverting back to requesting for authentication credentials led to observe a successful TLS connection in the packet capture.
Pass/Fail with Explanation	Pass. The evaluator confirmed that when the client presented a certificate with an identifier that did not match the expected identifier by the server for authentication, the server deemed the certificate to be invalid and did not allow the client to authenticate using the certificate. This meets the testing requirements.

6.6 EP_SSHC (Linux)

6.6.1 FCS_COP.1(1) Test #1

Item	Data
Test Assurance Activity	*If perform encryption/decryption services is chosen, the evaluator shall verify that the TSS describes the counter mechanism including rationale that the counter values*

provided are unique.

AES-CTR Tests:

- Test 1: Known Answer Tests (KATs)
  There are four Known Answer Tests (KATs) described below. For all KATs, the plaintext, IV, and ciphertext values shall be 128-bit blocks. The results from each test may either be obtained by the validator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation.

  To test the encrypt functionality, the evaluator shall supply a set of 10 plaintext values and obtain the ciphertext value that results from encryption of the given plaintext using a key value of all zeros and an IV of all zeros. Five plaintext values shall be encrypted with a 128-bit all zeros key, and the other five shall be encrypted with a 256-bit all zeros key. To test the decrypt functionality, the evaluator shall perform the same test as for encrypt, using 10 ciphertext values as input.

  To test the encrypt functionality, the evaluator shall supply a set of 10 key values and obtain the ciphertext value that results from encryption of an all zeros plaintext using the given key value and an IV of all zeros. Five of the key values shall be 128-bit keys, and the other five shall be 256-bit keys. To test the decrypt functionality, the evaluator shall perform the same test as for encrypt, using an all zero ciphertext value as input.

  To test the encrypt functionality, the evaluator shall supply the two sets of key values described below and obtain the ciphertext values that result from AES encryption of an all zeros plaintext using the given key values and an IV of all zeros. The first set of keys shall have 128 128-bit keys, and the second shall have 256 256-bit keys. Key_i in each set shall have the leftmost i bits be ones and the rightmost N-i bits be zeros, for i in [1, N]. To test the decrypt functionality, the evaluator shall supply the two sets of key and ciphertext value pairs described below and obtain the plaintext value that results from decryption of the given ciphertext using the given key values and an IV of all zeros. The first set of key/ciphertext pairs shall have 128 128-bit key/ciphertext pairs, and the second set of key/ciphertext pairs shall have 256 256-bit pairs. Key_i in each set shall have the leftmost i bits be ones and the rightmost N-i bits be zeros for i in [1, N]. The ciphertext value in each pair shall be the value that results in an all zeros plaintext when decrypted with its corresponding key.

  To test the encrypt functionality, the evaluator shall supply the set of 128

	plaintext values described below and obtain the two ciphertext values that result from encryption of the given plaintext using a 128-bit key value of all zeros and using a 256 bit key value of all zeros, respectively, and an IV of all zeros. Plaintext value i in each set shall have the leftmost bits be ones and the rightmost 128-i bits be zeros, for i in [1, 128]. To test the decrypt functionality, the evaluator shall perform the same test as for encrypt, using ciphertext values of the same form as the plaintext in the encrypt test as input. • Test 2: Multi-Block Message Test The evaluator shall test the encrypt functionality by encrypting an i-block message where 1 less-than i less-than-or-equal to 10. For each i the evaluator shall choose a key, IV, and plaintext message of length i blocks and encrypt the message, using the mode to be tested, with the chosen key. The ciphertext shall be compared to the result of encrypting the same plaintext message with the same key and IV using a known good implementation. The evaluator shall also test the decrypt functionality by decrypting an i-block message where 1 less-than i less-than-or-equal to 10. For each i the evaluator shall choose a key and a ciphertext message of length i blocks and decrypt the message, using the mode to be tested, with the chosen key. The plaintext shall be compared to the result of decrypting the same ciphertext message with the same key using a known good implementation. • Test 3: Monte-Carlo Test For AES-CTR mode perform the Monte Carlo Test for ECB Mode on the encryption engine of the counter mode implementation. There is no need to test the decryption engine. The evaluator shall test the encrypt functionality using 200 plaintext/key pairs. 100 of these shall use 128 bit keys, and 100 of these shall use 256 bit keys. The plaintext values shall be 128-bit blocks. For each pair, 1000 iterations shall be run as follows: For AES-ECB mode # Input: PT, Key for i = 1 to 1000: CT[i] = AES-ECB-Encrypt(Key, PT) PT = CT[i] • The ciphertext computed in the 1000th iteration is the result for that trial. This result shall be compared to the result of running 1000 iterations with the same values using a known good implementation.
Pass/Fail with Explanation	Pass. The testing requirements have been satisfied by validating each of the claimed cryptographic algorithms for conformance to the requirements specified in their respective standards

6.6.2 FCS_SSHC_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	The evaluator will, for each public key algorithm supported, show that the TOE supports the use of that public key algorithm to authenticate a user connection to an SSH server. Any configuration activities required to support this test shall be performed according to instructions in the guidance documentation.
Test Steps	• Attempt a connection to a remote SSH server configured with the TOE public key. **SSH-RSA** • Generate a ssh key pair. • Load the private key onto the TOE. • The corresponding public key is added to the authorized_keys file on the server. • Choose the authentication method as public key on the TOE. • Attempt a connection from the TOE to the server with user acumensec using an rsa private key. • Note the logs to verify that the user was successfully authenticated using ssh-rsa algorithm. • The evaluator observed the logs on the server to verify that the TOE uses ssh-rsa (userauth_pubkey) public key algorithm to authenticate the user 'acumensec' to an SSH server. • Verify with packet capture that the connection is successful. **Rsa-sha2-256** • Choose the authentication method as public key on the TOE. • Attempt a connection from the TOE to the server with user acumensec using an rsa private key . Not the RSA signature algorithm Is now set to rsa-sha2-256. • The evaluator observed the logs on the server to verify that the TOE uses rsa-sha2-256 (userauth_pubkey) public key algorithm to authenticate the user 'acumensec' to an SSH server. • Verify with packet capture that is connection is successful. **Rsa-sha2-512** • Choose the authentication method as public key. • Attempt a connection from the TOE to the server with user acumensec using an rsa private key . The corresponding public key is added to the authorized_keys file on the server. • The evaluator observed the logs on the server to verify that the TOE uses rsa-sha2-512 (userauth_pubkey) public key algorithm to authenticate the user 'acumensec' to an SSH server. • Verify with packet capture that the connection is successful. **Ecdsa-sha2-nistp256** • Generate a ssh key pair. • Load the private key onto the TOE. • Choose the authentication method as public key and the Host Key Signature Algorithm as ecdsa-sha2-nistp256.

	• Attempt a connection from the TOE to the server with user acumensec using an ecdsa private key . The corresponding public key is added to the authorized_keys file on the server. • The evaluator observed the logs on the server to verify that the TOE uses ecdsa-sha2-nistp256 (userauth_pubkey) public key algorithm to authenticate the user 'acumensec' to an SSH server. • Verify with packet capture that the connection is successful. **Ecdsa-sha2-nistp384** • Generate a ssh key pair. • Load the private key onto the TOE. • Choose the authentication method as public key on the TOE. • Attempt a connection from the TOE to the server with user acumensec using an ecdsa private key . The corresponding public key is added to the authorized_keys file on the server. • The evaluator observed the logs on the server to verify that the TOE uses ecdsa-sha2-nistp384 (userauth_pubkey) public key algorithm to authenticate the user 'acumensec' to an SSH server. • Verify with packet capture that the connection is successful.
Pass/Fail with Explanation	Pass. The user connection from the TOE to the server is authenticated successfully using each of the public key algorithms.

6.6.3 FCS_SSHC_EXT.1.1 Test #2

Item	Data
Test Assurance Activity	[Conditional] Using the guidance documentation, the evaluator will configure the TOE to perform password-based authentication to an SSH server, and demonstrate that a user can be successfully authenticated by the TOE to an SSH server using a password as an authenticator. **TD0420 has been applied.**
Test Steps	• The evaluator configured the TOE to perform password-based authentication to an SSH server. • Attempt a connection to an SSH server and show the user authentication succeeds. • Verify with packet capture to ensure that the connection was successful.
Pass/Fail with Explanation	Pass. The user can be successfully authenticated from the TOE to the server using a valid password.

6.6.4 FCS_SSHC_EXT.1.2 Test #1

Item	Data
Test Assurance Activity	The evaluator will demonstrate that if the TOE receives a packet larger than that specified in this component, that packet is dropped.
Test Steps	• Connect to a remote server running a tool that will send a packet larger than the allow packet size. • The evaluator ensured that the TOE terminates the connection.

Pass/Fail with Explanation	Pass. The connection between the TOE and server was terminated after the TOE received a large packet.

6.6.5 FCS_SSHC_EXT.1.3 Test #1

Item	Data
Test Assurance Activity	The evaluator will establish an SSH connection using each of the encryption algorithms specified by the requirement. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test.
Test Steps	• Connect to a remote SSH server using each of the claimed encryption algorithms. **aes128-cbc** • Configure the server to allow only aes128-cbc algorithm. • Attempt a connection from the TOE to the server and verify the connection succeeds. • Verify through packet capture that the connection succeeds. **aes256-cbc** • Configure the server to allow only aes256-cbc algorithm. • Show the connection is successful. • Verify through packet capture the connection succeeds. **aes128-ctr** • Configure the server to allow only aes128-ctr algorithm. • Show the connection is successful. • Verify with packet capture that the connection succeeds. **aes256-ctr** • Configure the server to allow only aes256-ctr algorithm. • Show the connection is successful. • Verify with packet capture the connection succeeds.
Pass/Fail with Explanation	Pass. The TOE successfully connects to a remote SSH server using each of the claimed encryption algorithms.

6.6.6 FCS_SSHC_EXT.1.3 Test #2

Item	Data
Test Assurance Activity	The evaluator will configure an SSH server to only allow the 3des-cbc encryption algorithm and no other encryption algorithms. The evaluator will attempt to establish an SSH connection from the TOE to the SSH server and observe that the connection is rejected.
Test Steps	• Configure an SSH server to only allow the 3des-cbc encryption algorithm and no other encryption algorithms. • Attempt a connection to a remote server configured to use 3des-cbc encryption algorithm only. Show the TOE rejects the connection. • Verify with packet capture that the connection is rejected.
Pass/Fail with Explanation	Pass. The TOE rejects a connection to a remote server when attempted to connect using 3des-cbc algorithm.

6.6.7 FCS_SSHC_EXT.1.4 Test #1

Item	Data
Test Assurance Activity	The evaluator will establish a SSH connection using each of the public key algorithms specified by the requirement to authenticate an SSH server to the TOE. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test.
Expected Test Results	As a part of FCS_SSHC_EXT.1.1 Test#1, • The evaluator for each public key algorithm supported, showed using debug logs that the TOE supports the use of that public key algorithm to authenticate a user connection to an SSH server. The evaluator also verified the successful negotiation of the SSH connection with encrypted packets exchanged between the client and the server which implies that SSH server in response authenticated to the TOE for each of the public key algorithm used to authenticate the user to the SSH server.
Pass/Fail with Explanation	Pass. The test requirements were tested as a part of FCS_SSHC_EXT.1.1 Test#1.

6.6.8 FCS_SSHC_EXT.1.4 Test #2

Item	Data
Test Assurance Activity	The evaluator will configure an SSH server to only allow the ssh-dsa public key algorithm and no other public key algorithms. The evaluator will attempt to establish an SSH connection from the TOE to the SSH server and observe that the connection is rejected.
Test Steps	• Configure an SSH server to only allow the ssh-dsa public key algorithm. • Attempt a connection from the TOE to the server and show that the connection is rejected. • Verify through packet capture the connection is rejected.
Pass/Fail with Explanation	Pass. The connection from the TOE to the server is rejected when the server allows only ssh-dsa public key algorithm.

6.6.9 FCS_SSHC_EXT.1.5 Test #1

Item	Data
Test Assurance Activity	The evaluator will establish a SSH connection using each of the integrity algorithms, except "implicit", specified by the requirement. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test. **TD0446 has been applied.**
Test Steps	• Attempt a connection from the TOE to the server using each of the claimed integrity algorithms. **Hmac-sha1** • Configure the server to allow only hmac-sha1 integrity algorithm. • Attempt a connection from the TOE to the server and verify the connection succeeds. • Verify through the packet capture that the connection is successful. **Hmac-sha1-96** • Configure the server to allow only hmac-sha1-96 integrity algorithm.

	• Attempt a connection from the TOE to the server and verify the connection succeeds. • Verify through the packet capture that the connection is successful. **Hmac-sha2-256** • Configure the server to allow only hmac-sha2-256 integrity algorithm. • Attempt a connection from the TOE to the server and verify the connection succeeds. • Verify through the packet capture that the connection is successful.
Pass/Fail with Explanation	Pass. The connection from the TOE to the server is successful using each of the claimed integrity algorithms.

6.6.10 FCS_SSHC_EXT.1.5 Test #2

Item	Data
Test Assurance Activity	The evaluator will configure an SSH server to only allow the "none" MAC algorithm. The evaluator will attempt to connect from the TOE to the SSH server and observe that the attempt fails. Note: To ensure the proposed MAC algorithm is used, the evaluator shall ensure a non-aes*- gcm@openssh.com encryption algorithm is negotiated while performing this test. **TD0446 has been applied.**
Test Steps	• Attempt to connect to a remote SSH server configured to only support 'none' MAC algorithm using the acumen-sshc tool. • Show that the TOE rejects the connection. • Verify through the packet capture that the connection was not successful.
Pass/Fail with Explanation	Pass. The connection from the TOE to the server is rejected when attempted to connect with the "none" mac algorithm.

6.6.11 FCS_SSHC_EXT.1.5 Test #3

Item	Data
Test Assurance Activity	Test 3: The evaluator will configure an SSH server to only allow the hmac- md5 MAC algorithm. The evaluator will attempt to connect from the TOE to the SSH server and observe that the attempt fails. Note: To ensure the proposed MAC algorithm is used, the evaluator shall ensure a non-aes*- gcm@openssh.com encryption algorithm is negotiated while performing this test. **TD0446 has been applied.**
Test Steps	• Configure an SSH server to only allow the hmac- md5 MAC algorithm. • Attempt a connection from the TOE to a server which only allows hmac-md5 MAC algorithm. Show that the connection is rejected. • The evaluator observed the packet capture to ensure that the connection was not successful.

Pass/Fail with Explanation	Pass. The connection from the TOE to the server was rejected when the server allows only hmac-md5 algorithm.

6.6.12 FCS_SSHC_EXT.1.6 Test #1

Item	Data
Test Assurance Activity	The evaluator will configure an SSH server to permit all allowed key exchange methods. The evaluator will attempt to connect from the TOE to the SSH server using each allowed key exchange method and observe that each attempt succeeds.
Test Steps	• The evaluator configured the acumen-sshc tool to be an SSH server waiting for SSH connections on all allowed key exchange methods. The evaluator attempted a connection from the TOE to the server using each of the claimed key exchange methods. **Dh-group14** • The evaluator observed the TOE output and ensured that the SSH connection was successful. • The evaluator observed the packet capture and ensured that the SSH connection was successful using diffie-hellman-group14-sha1 key exchange methods. **Ecdh-sha2-nistp256** • The evaluator observed the TOE output and ensured that the SSH connection was successful. • The evaluator observed the packet capture and ensured that the SSH connection was successful using ecdh-sha2-nistp256 key exchange methods. **Ecdh-sha2-nistp384** • The evaluator observed the TOE output and ensured that the SSH connection was successful. • The evaluator observed the packet capture and ensured that the SSH connection was successful using ecdh-sha2-nistp384 key exchange methods. **Ecdh-sha2-nistp521** • The evaluator observed the TOE output and ensured that the SSH connection was successful. • The evaluator observed the packet capture and ensured that the SSH connection was successful using ecdh-sha2-nistp521 key exchange methods.
Pass/Fail with Explanation	Pass. The TOE accepts connections to a remote SSH server using each of the claimed key exchange algorithm.

6.6.13 FCS_SSHC_EXT.1.7 Test #1

Item	Data
Test Assurance Activity	The evaluator will configure the TOE to create a log entry when a rekey occurs. The evaluator will connect to the TOE with an SSH client and cause a rekey to occur according to the selection(s) in the ST, and subsequently the evaluator uses available methods and tools to verify that rekeying occurs. This could be done, e.g., by checking that a corresponding audit event has been generated by the TOE, if the TOE supports auditing of rekey events.

	TD0331 has been applied.
Test Steps	**SSH Data Based Rekey** • The evaluator configured the TOE to ensure that the SSH client connection be rekeyed after no more than 1 Gigabyte of data has been transmitted using that key in system.properties. • The evaluator created a SSH/SFTP server resource called "test" to establish an SSH connection to the server 10.1.3.51 on port 22 in Resources tab. • The evaluator created a workflow for the TOE as a client to establish an SSH/SFTP connection to the SFTP server. The previously created resource "test" is chosen here. • The evaluator set the source file of 1GigaByte (1GB.zip) to be transferred from the SFTP server to the TOE acting as SFTP client to cause to the rekey to occur. • The evaluator executed the workflow and ensured that the SFTP connection was successful. • The evaluator observed the logs to ensured that TOE audited the new keys generated and also the rekey event logs that occurred prior to 1 Gigabyte of data transmitted and confirmed that the TOE initiated a rekey(TOE sent SSH_MSG_KEXINIT initiating rekey as Key re-exchange is started by sending an SSH_MSG_KEXINIT packet). **SSH Time Based Rekey** • The evaluator configured the TOE to ensure that the SSH client connection be rekeyed after no more than 1 hour of time the SSH lifetime using that key in system.properties. • The evaluator created a SSH/SFTP server resource called "test" to establish an SSH connection to the server 10.1.3.51 on port 22 in Resources tab. • The evaluator created a workflow for the TOE as a client to establish an SSH/SFTP connection to the SFTP server (test). The evaluator created a Get Files resource to extract a file from the SFTP server and then set a delay of 61 minutes to keep the SSH connection alive more than an hour and then added another Get Files resource to extract a file from the SFTP server. • The evaluator executed the workflow and ensured that the SFTP connection was successful. • The evaluator observed the logs to ensure that the TOE audited the new keys generated at 2:03pm and also the rekey event logs that occurred prior to 1 hour of keeping the SSH connection alive and confirmed that the TOE initiated a rekey at 3:03pm (TOE sent SSH_MSG_KEXINIT initiating rekey as Key re-exchange is started by sending an SSH_MSG_KEXINIT packet).
Pass/Fail with Explanation	Pass. The evaluator connected the TOE acting as an SSH client to the SSH server and caused a rekey to occur prior to 1 Gigabyte of data transmitted and prior to 1 hour of keeping the SSH connection alive and subsequently the evaluator observed the audit logs to verify that rekeying occurs.

6.6.14 FCS_SSHC_EXT.1.8 Test #1

Item	Data
Test Assurance Activity	The evaluator will delete all entries in the TOE's list of recognized SSH server host keys and, if selected, all entries in the TOE's list of trusted certification authorities. The evaluator will initiate a connection from the TOE to an SSH server. The evaluator shall ensure that the TOE either rejects the connection or displays the SSH server's public key (either the key bytes themselves or a hash of the key using any allowed hash algorithm) and prompts the user to accept or deny the key before continuing the connection.
Test Steps	• The evaluator ensured that Implicit trust of SSH connections is not allowed by setting the radio button to "No". • The evaluator created an SSH server resource "test" with the server IP set to 10.1.3.51 and port number set to 22. • The TOE does-not store the recognized SSH server host keys but rather store the information on the resource for the SSH/SFTP server the user is connecting to, and the configuration takes place on the Connection tab of the SSH Server resource. The evaluator ensured that the host key was not configured in the SSH server resource. • The evaluator created a Workflow to initiate a connection to the SFTP/SSH server to get files from it and ensured that the connection was not successful. • The evaluator observed the audit logs generated to confirm that the SSH connection was not successful as the host key could not be verified by the TOE.
Pass/Fail with Explanation	Pass. The evaluator ensured that the TOE rejects the SSH connection as the host key could not be verified by the TOE as it was not configured in the TOE's list of recognized SSH server host keys.

6.6.15 FCS_SSHC_EXT.1.8 Test #2

Item	Data
Test Assurance Activity	The evaluator will add an entry associating a host name with a public key into the TOE's local database. The evaluator will replace, on the corresponding SSH server, the server's host key with a different host key. The evaluator will initiate a connection from the TOE to the SSH server using password-based authentication, shall ensure that the TOE rejects the connection, and shall ensure that the password was not transmitted to the SSH server (for example, by instrumenting the SSH server with a debugging capability to output received passwords).
Test Steps	• The evaluator added a public key into the TOE's local database. • The evaluator specified the fingerprint of the server's public key, which will be used to authenticate the server. The evaluator ensured that that SSH connection was successful. • The evaluator reconfigured the server such that the server host key is replaced with a new key. • The evaluator added a public key into the TOE's local database. • The evaluator attempted a connection from the TOE to the SSH server using password-based authentication while the older fingerprint of the server's public key was configured and ensured that the connection failed.

	• The evaluator observed the logs on the server and ensured that the SSH connection was not established due to "unknown host key". The evaluator further confirmed checking the logs that the server received a disconnect from the TOE prior to sending the password which implies that the TOE did not transmit the password to the server. • The evaluator observed the packet capture to ensure that the SSH connection was not successful.
Pass/Fail with Explanation	Pass. The TOE rejects the connection to the SSH server and does not transmit the password to the server if the host key of the server is replaced with a different key.

6.7 EP_SSHS (Linux)

6.7.1 FCS_SSHS_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	The evaluator will, for each public key algorithm supported, show that the TOE supports the use of that public key algorithm to authenticate a user connection from an SSH client. Any configuration activities required to support this test shall be performed according to instructions in the guidance documentation.
Test Steps	• Generate a ssh rsa key pair. • The evaluator observed the private key and public key generated. • Load the public key into the web user SSH keys on the TOE. • Set the web user authentication method to public key. • Set the web user Feature to SFTP. **SSH-RSA** • Choose the server public key signature algorithm to ssh-rsa. • Attempt a connection from the SSH client to the server using a public key to authenticate the user connection and ensure that the connection was successful. The evaluator observed the logs on the client to ensure that the server (TOE) accepted the public-key algorithm ssh-rsa (log: Server accepts key) to authenticate the user connection from the SSH client and the authentication succeeded. • Verify via packet capture that the connection is successful. **RSA-SHA2-256** • Choose the server public key signature algorithm to rsa-sha2-256. • Restart the SFTP service. • Attempt a connection from the SSH client to the server using a public key to authenticate the user connection and ensure that the connection was successful. The evaluator observed the logs on the client to ensure that the server (TOE) accepted the public-key algorithm rsa-sha2-256 (log: Server accepts key) to authenticate the user connection from the SSH client and the authentication succeeded. • Verify with wireshark capture that the connection was successful. **RSA-SHA2-512** • Choose the server public key signature algorithm to rsa-sha2-512 • Restart the SFTP service.

	• Attempt a connection from the SSH client to the server using a public key to authenticate the user connection and ensure that the connection was successful. The evaluator observed the logs on the client to ensure that the server (TOE) accepted the public-key algorithm rsa-sha2-512 (log: Server accepts key) to authenticate the user connection from the SSH client and the authentication succeeded. • Verify via packet capture that the connection is successful. **ECDSA-SHA2-NISTP256** • Generate a ssh ecdsa key pair. • The evaluator observed the private key and public key generated. • Load the public key into the web user SSH keys on the TOE. • Choose the server public key signature algorithm to ecdsa-sha2-nistp256 • Attempt a connection from the SSH client to the server using a public key to authenticate the user connection and ensure that the connection was successful. The evaluator observed the logs on the client to ensure that the server (TOE) accepted the public-key algorithm ecdsa-sha2-nistp256 (log: Server accepts key) to authenticate the user connection from the SSH client and the authentication succeeded. • Verify via packet capture to ensure that the SSH connection was successful. **ECDSA-SHA2-NISTP384** • Choose the server public key signature algorithm to ecdsa-sha2-nistp384 • Attempt a connection from the SSH client to the server using a public key to authenticate the user connection and ensure that the SSH was successful. The evaluator observed the logs on the client to ensure that the server (TOE) accepted the public-key algorithm ecdsa-sha2-nistp384 (log: Server accepts key) to authenticate the user connection from the SSH client and the authentication succeeded. • Verify with packet capture that the SSH connection was successful.
Pass/Fail with Explanation	Pass. The TOE supports each of the public key algorithm to authenticate a user connection.

6.7.2 FCS_SSHS_EXT.1.1 Test #2

Item	Data
Test Assurance Activity	The evaluator shall choose one public key algorithm supported by the TOE. The evaluator shall generate a new key pair for that algorithm without configuring the TOE to recognize the public key for authentication. The evaluator shall use an SSH client to attempt to connect to the TOE with the new key pair and demonstrate that authentication fails.
Test Steps	• Set the SSH user authentication type to public key. • Create a new keypair without adding the public key to the TOE. • Attempt to login using the newly created private key. • Show the connection is rejected and login is unsuccessful. • The evaluator observed the packet capture and ensured that the connection was not successful.

Pass/Fail with Explanation	Pass. The authentication from the client to the TOE fails when attempted to connect with an unknown key.

6.7.3 FCS_SSHS_EXT.1.1 Test #3

Item	Data
Test Assurance Activity	[Conditional] Using the guidance documentation, the evaluator will configure the TOE to perform password-based authentication on a client and demonstrate that a user can be successfully authenticated by the TOE using a password as an authenticator. **TD0420 has been applied.**
Test Steps	• Set the user's authentication type as Password based. • Attempt to login to the TOE using a valid username/password combination. • The evaluator entered the password. • The evaluator ensured that the connection was successful. • The evaluator ensured via packet capture that the SSH connection was successful.
Pass/Fail with Explanation	Pass. The TOE allows a user to be successfully authenticated with correct login credentials.

6.7.4 FCS_SSHS_EXT.1.1 Test #4

Item	Data
Test Assurance Activity	[Conditional] The evaluator shall use an SSH client, enter an incorrect password to attempt to authenticate to the TOE, and demonstrate that the authentication fails. **TD0420 has been applied.**
Test Steps	• Attempt a connection to the TOE using an invalid password (wrong password = TRY) • The evaluator confirmed that the connection was not successful, and the authentication failed. • The evaluator observed the packet capture and ensured that the SSH connection was not successful.
Pass/Fail with Explanation	Pass. The SSH client cannot authenticate using an incorrect password.

6.7.5 FCS_SSHS_EXT.1.2 Test #1

Item	Data
Test Assurance Activity	The evaluator will demonstrate that if the TOE receives a packet larger than that specified in this component, that packet is dropped.
Test Steps	• The evaluator configured the TOE as SFTP/SSH server with authentication type set to password. • The evaluator used the acumen-sshfix tool as an SSH client to send a packet larger than 65535 bytes and ensured that the packet is dropped.

	• The evaluator observed the tool output and ensured that the connection was not successful. • The evaluator observed the packet capture and ensured that the connection was not successful.
Pass/Fail with Explanation	Pass. The TOE drops large packets that are received within an SSH session.

6.7.6 FCS_SSHS_EXT.1.3 Test #1

Item	Data
Test Assurance Activity	The evaluator will initiate an SSH connection using each of the encryption algorithms specified by the requirement. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test.
Test Steps	• Configure the SSH server on the TOE to enable the claimed encryption algorithms. • The evaluator initiated an SSH connection using aes128-cbc as the encryption algorithm and ensured that the connection established. • The evaluator observed the packet capture and ensured that the SSH connection was successful using aes128-cbc as the encryption algorithm. • The evaluator initiated an SSH connection using aes256-cbc as the encryption algorithm and ensured that the connection established. • The evaluator observed the packet capture and ensured that the SSH connection was successful using aes256-cbc as the encryption algorithm. • The evaluator initiated an SSH connection using aes128-ctr as the encryption algorithm and ensured that the connection established. • The evaluator observed the packet capture and ensured that the SSH connection was successful using aes128-ctr as the encryption algorithm. • The evaluator initiated an SSH connection using aes256-ctr as the encryption algorithm and ensured that the connection established. • The evaluator observed the packet capture and ensured that the SSH connection was successful using aes256-ctr as the encryption algorithm.
Pass/Fail with Explanation	Pass. The connection from a client to the TOE is successful using each of the claimed encryption algorithms.

6.7.7 FCS_SSHS_EXT.1.3 Test #2

Item	Data
Test Assurance Activity	The evaluator will configure an SSH client to only propose the 3des-cbc encryption algorithm and no other encryption algorithms. The evaluator will attempt to establish an SSH connection from the client to the TOE server and observe that the connection is rejected.
Test Steps	• The evaluator attempted to connect to the TOE using 3des-cbc encryption algorithm and no other encryption algorithms. • The evaluator verified that the connection is rejected as the there was no matching cipher found. • The evaluator observed the packet capture and ensured that the SSH connection was not successful.

Pass/Fail with Explanation	Pass. The connection from the client to the TOE is rejected when the client attempts to connect with 3des-cbc algorithm.

6.7.8 FCS_SSHS_EXT.1.4 Test #1

Item	Data
Test Assurance Activity	Using an appropriately configured client, the evaluator will establish an SSH connection using each of the public key algorithms specified by the requirement to authenticate. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test.
Expected Test Results	As a part of FCS_SSHS_EXT.1.1 Test#1, • The evaluator for each public key algorithm supported, showed using debug logs that the TOE supports the use of that public key algorithm to authenticate a user connection from an SSH client. The evaluator also verified the successful negotiation of the SSH connection with encrypted packets exchanged between the client and the server which implies that SSH client in response authenticated to the TOE for each of the public key algorithm used to authenticate the user connection from an SSH client.
Pass/Fail with Explanation	Pass. The test requirements were tested as a part of FCS_SSHC_EXT.1.1 Test#1.

6.7.9 FCS_SSHS_EXT.1.4 Test #2

Item	Data
Test Assurance Activity	The evaluator will configure an SSH client to propose only the ssh-dsa public key algorithm and no other public key algorithms. Using this client, the evaluator will attempt to establish an SSH connection to the TOE and observe that the connection is rejected.
Test Steps	• The evaluator generated an ssh dsa key pair using ssh-keygen. • The evaluator ensured that the SSH key-pair was generated. • The evaluator uploaded the public-key on the TOE. • The evaluator ensure that the dsa public key was uploaded to the TOE with name dsakey. • The evaluator checked the server public-key algorithms and ensured that the ssh-dss (dsa) public key algorithm is not supported by the TOE. • The evaluator attempted to select the uploaded DSA key for the TOE to leverage it as the Host Key by the SFTP server but the TOE rejected it displaying that there is no valid public key algorithm for the DSA Host key that was attempted to select. • The evaluator attempted a connection from the SSH client to the server using only the ssh-dss (dsa) public key algorithm to authenticate the user connection and observed that the connection was not successful. • The evaluator further verified the packet-capture to ensure that the TOE rejected the connection.
Pass/Fail with Explanation	Pass. The connection attempt between the client and the TOE fails when the public key algorithm is ssh-dsa.

6.7.10 FCS_SSHS_EXT.1.5 Test #1

Item	Data
Test Assurance Activity	Using an appropriately configured client, the evaluator will establish a SSH connection using each of the integrity algorithms, except "implicit", specified by the requirement. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test. **TD0446 has been applied.**
Test Steps	• The evaluator configured the TOE to support the claimed integrity algorithms. • The evaluator used the acumen-sshfix tool as an SSH client to establish an SSH connection using each of the supported integrity algorithms and ensured that the connection was successful. • The evaluator observed the packet capture to ensure that the SSH connection was successful using each of the claimed integrity algorithms.
Pass/Fail with Explanation	Pass. The TOE allows a successful connection from the client using each of the claimed integrity algorithms.

6.7.11 FCS_SSHS_EXT.1.5 Test #2

Item	Data
Test Assurance Activity	The evaluator will configure an SSH client to only allow the “none” MAC algorithm. Using this client, the evaluator will attempt to connect to the TOE and observe that the attempt fails. Note: To ensure the proposed MAC algorithm is used, the evaluator shall ensure a non-aes*- gcm@openssh.com encryption algorithm is negotiated while performing this test. **TD0446 has been applied.**
Test Steps	• The evaluator configured the TOE as an SSH server to support only claimed MAC algorithms. • The evaluator used the acumen-sshfix tool as an SSH client to establish an SSH connection using “none” as the integrity algorithm and ensured that the SSH negotiation failed. • The evaluator observed the packet capture to ensure that the SSH negotiation failed when “none” MAC algorithm was presented by the SSH client.
Pass/Fail with Explanation	Pass. The connection is rejected when a client attempts to connect with the none MAC algorithm.

6.7.12 FCS_SSHS_EXT.1.5 Test #3

Item	Data
Test Assurance Activity	The evaluator will configure an SSH client to only allow the hmac- md5 MAC algorithm. using this client, the evaluator will attempt to connect to the TOE and observe that the attempt fails

	Note: To ensure the proposed MAC algorithm is used, the evaluator shall ensure a non-aes*- gcm@openssh.com encryption algorithm is negotiated while performing this test. **TD0446 has been applied.**
Test Steps	• The evaluator attempted a connection to the TOE from a client using hmac-md5 MAC algorithm and ensured that the connection is rejected. • The evaluator observed the packet capture and ensured that connection attempt failed when the SSH client attempted using hmac- md5 MAC algorithm.
Pass/Fail with Explanation	Pass. The connection was rejected when a client attempted to connect with hmac-md5 MAC algorithm.

6.7.13 FCS_SSHS_EXT.1.6 Test #1

Item	Data
Test Assurance Activity	For each of the allowed key exchange methods, the evaluator will configure an SSH client to propose only it and attempt to connect to the TOE and observe that each attempt succeeds.
Test Steps	• The evaluator configured the TOE as an SFTP/SSH server using each of the claimed key exchange methods. • The evaluator used the acumen-sshsfix tool as an SSH client to propose each of the claimed key exchange methods. • The evaluator observed the tool output which indicated that SSH connection succeeded with each of the key exchange methods. (diffie-hellman-group14-sha1, ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521) • The evaluator observed the packet capture which indicated that SSH connection succeeded with each of the key exchange methods.
Pass/Fail with Explanation	Pass. The TOE allows a client to connect using each of the claimed key exchange methods.

6.7.14 FCS_SSHS_EXT.1.6 Test #2

Item	Data
Test Assurance Activity	The evaluator shall configure an SSH client to only allow the diffiehellman-group1-sha1 key exchange. The evaluator shall attempt to connect from the SSH client to the SSH Server and observe that the attempt fails.
Test Steps	• The evaluator used the acumen-sshsfix tool as an SSH client to propose diffiehellman-group1-sha1 as the claimed key exchange method. • The evaluator observed the tool output and ensured that the connection was rejected. • The evaluator observed the packet capture and ensured that the attempt to connect from the SSH client to the SSH Server using diffiehellman-group1-sha1 as the claimed key exchange method failed.
Pass/Fail with Explanation	Pass. The connection to the TOE was unsuccessful when attempted to connect with an unsupported key exchange algorithm.

6.7.15 FCS_SSHS_EXT.1.7 Test #1

Item	Data
Test Assurance Activity	The evaluator will configure the TOE to create a log entry when a rekey occurs. The evaluator will connect to the TOE with an SSH client and cause a rekey to occur according to the selection(s) in the ST, and subsequently the evaluator uses available methods and tools to verify that rekeying occurs. This could be done, e.g., by checking that a corresponding audit event has been generated by the TOE, if the TOE supports auditing of rekey events. **TD0331 has been applied.**
Test Steps	• The evaluator configured the TOE to act as an SSH server with local IP address 10.1.3.253 and port 1214 with authentication type set to password. • The evaluator configured the maximum bytes before rekey to 1 Gigabyte data in system.properties. • The evaluator used the acumen-sshs tool as an SSH client to connect to the TOE configured as an SSH server. The tool kept the connection alive until more than 1 Gigabyte is transferred using the key and ensured that the SSH connection rekeyed before 1 Gigabyte of data has been transmitted as seen in the tool output. • The evaluator configured the TOE to act as an SSH server with local IP address 10.1.3.253 and port 1214 with authentication type set to password. • The evaluator configured the maximum seconds before rekey to 3600 seconds in system.properties. • The evaluator used the acumen-sshs tool as an SSH client to connect to the TOE configured as an SSH server. The tool kept the connection alive for more than 1 hour of time using the key and ensured that the SSH connection rekeyed before 1 hour of time has passed as seen in the tool output.
Pass/Fail with Explanation	Pass. The evaluator ensured that the SSH connection was rekeyed by the TOE after no more than 1 Gigabyte of data has been transmitted and no more than 1 hour using that key.

6.8 Static Analysis (Linux)

6.8.1 FCS_STO_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	For all credentials for which the application **implements functionality**, the evaluator shall verify credentials are encrypted according to FCS_COP.1(1)
Pass/Fail with Explanation	Pass. The application shall not store any credentials in non-volatile memory.

6.8.2 FDP_DEC_EXT.1.1 Test #1

Item	Data

Test Assurance Activity	**For Linux:** The evaluator shall verify that either the application software or its documentation provides a list of the hardware resources it accesses. **Non-applicable platforms removed.** **TD0434 has been applied.** **TD0515 has been applied**
Pass/Fail with Explanation	Pass. The evaluator reviewed the section named "TOE access to platform resources" in the GoAnywhere MFT Guidance Document and confirmed that Network connectivity is the only hardware platform resource accessed by the TOE.

6.8.3 FDP_DEC_EXT.1.2 Test #1

Item	Data
Test Assurance Activity	**For Linux:** The evaluator shall verify that either the application software or its documentation provides a list of sensitive information repositories it accesses. **Non-applicable platforms removed.** **TD0515 has been applied.**
Pass/Fail with Explanation	Pass. The evaluator reviewed the section named "TOE access to platform resources" in the GoAnywhere MFT Guidance Document and confirmed that System logs are the only sensitive information repository accessed by the TOE.

6.8.4 FPT_AEX_EXT.1.2 Test #1

Item	Data
Test Assurance Activity	The evaluator shall verify that no memory mapping requests are made with write and execute permissions. The method of doing so varies per platform. **For Linux:** The evaluator shall perform static analysis on the application to verify that both • mmap is never be invoked with both the PROT_WRITE and PROT_EXEC permissions, and • mprotect is never invoked with the PROT_EXEC permission. **Non-applicable platforms removed.**
Test Steps	• Use the strace command to perform static analysis on the application • Verify the mmap is never invoked with both PROT_WRITE and PROT_EXEC and mprotect is never invoked with PROT_EXEC permission
Pass/Fail with Explanation	Pass. The evaluator verified that no memory mapping requests are made with write and execute permissions. This meets the test requirements.

6.8.5 FPT_AEX_EXT.1.5 Test #1

Item	Data
Test Assurance Activity	The evaluator will inspect every native executable included in the TOE to ensure that stack-based buffer overflow protection is present. **For Windows:** Applications that run as Managed Code in the .NET Framework do not require these stack protections. Applications developed in Object Pascal using the Delphi IDE compiled with RangeChecking enabled comply with this element. For other code, the evaluator shall review the TSS and verify that the /GS flag was used during compilation. The evaluator shall run a tool like, BinScope, that can verify the correct usage of /GS.

	For PE , the evaluator will disassemble each and ensure the following sequence appears: mov rcx, QWORD PTR [rsp+(...)] xor rcx, (...) call (...) . **For ELF executables**, the evaluator will ensure that each contains references to the symbol **__stack_chk_fail**. Tools such as Canary Detector may help automate these activities. **Non-applicable platforms removed.**
Pass/Fail with Explanation	Pass. This test is not applicable as the TOE is not composed of PE or ELF executables. Also, the TOE is composed of Java code. All Java objects are strictly typed with explicit sizes, so it is not possible to overflow a buffer in Java code. Therefore, the TOE does not require stack-based buffer overflow protection.

6.8.6 FPT_API_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	The evaluator shall verify that the TSS lists the platform APIs used in the application. The evaluator shall then compare the list with the supported APIs (available through e.g. developer accounts, platform developer groups) and ensure that all APIs listed in the TSS are supported.
Test Steps	• The evaluator verified that the TSS lists the platform APIs used in the application. • The evaluator compared the list with the supported APIs (available through e.g. developer accounts, platform developer groups) and ensure that all APIs listed in the TSS are supported. • The evaluator further did research about all the shared libraries (so files) and java archive (jar) files that were claimed in the TSS and ensured that all the files in the claimed list were part of the OpenJDK package which in-turn was found in the platform developer documentation. This indicates that the TOE uses only platform documented APIs.
Pass/Fail with Explanation	Pass. All the API's listed in the TSS are supported.

6.8.7 FPT_TUD_EXT.2.1 Test #1

Item	Data
Test Assurance Activity	The evaluator shall verify that application updates are distributed in the format supported by the platform. This varies per platform: **For Linux:** The evaluator shall ensure that the application is packaged in the format of the package management infrastructure of the chosen distribution. For example, applications running on Red Hat and Red Hat derivatives shall be packaged in RPM format. Applications running on Debian and Debian derivatives shall be packaged in DEB format.

	Non-applicable platforms removed.
Test Steps	• Confirm that the TOE's installation package comes in RPM format. • Confirm that the TOE's upgrader package comes in RPM format.
Pass/Fail with Explanation	Pass. The TOE installer is packaged in the RPM format.

6.9 X509 (Linux)

6.9.1 FIA_X509_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: - The node certificate to be tested, - Two Intermediate CAs, and the - The self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. Test 1: The evaluator shall demonstrate that validating a certificate without a valid certification path results in the function failing, for each of the following reasons, in turn: • by establishing a certificate path in which one of the issuing certificates is not a CA certificate, • by omitting the basicConstraints field in one of the issuing certificates, • by setting the basicConstraints field in an issuing certificate to have CA=False, • by omitting the CA signing bit of the key usage field in an issuing certificate, and • by setting the path length field of a valid CA field to a value strictly less than the certificate path. The evaluator shall then establish a valid certificate path consisting of valid CA certificates and demonstrate that the function succeeds. The evaluator shall then remove trust in one of the CA certificates and show that the function fails. **TD0668 has been applied.**
Test Steps	• The evaluator created a chain of four certificates using the XCA tool. • The evaluator makes necessary changes to this certificate database as per the test requirements. Attempt a connection from the TOE to the server in which one of the issuing certificates is not a CA certificate. Note: The following test performed satisifies the FCS_TLSC_EXT.1.3 Test #1b requirement where the evaluator demonstrated that modifying the certificate chain

	used by the server to be invalid resulted in an authentication failure as the TOE was unable to construct a valid certificate path reason being one of the issuing certificates in the certificate path is not a CA certificate. • The evaluator used the XCA tool to ensure that one of the issuing certificates (ICA2_notCAcert) in the certificate path is not a CA certificate by transforming the original ICA2 issuing certificate to non-CA certificate. • The evaluator uploaded the Self signed CA certificate to the TOE's trust store (system keyvault). • The evaluator uploaded the certificate in pem chain format containing both the ICA1 and ICA2_notCAcert to the mysql directory of the database server. • The evaluator ensured that the certificate in pem chain format contains both ICA1 and ICA2_notCAcert. • The evaluator uploaded the server certificate 10.1.3.51.crt and the server key 10.1.3.51_key.pem to the mysql directory of the database server. • The evaluator configured the MySQL database server to leverage the uploaded server certificates and the server key for the TLS handshake with the client. • The evaluator configured the TOE to reach the Database server at Resources > Database Servers where the JDBC Driver was set to mariadb.jdbc .driver as the server was a MariaDB SQL server and provided the necessary URL information to reach the database server along with username and password of the database created on the server. WADATA is the database created on the MySQL server for the TOE on the Windows Platform • The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources created and confirmed that the client could not connect to the server as the server returned a certificate_unknown error to the client. • The evaluator observed the packet capture on the server and confirmed that the TLS handshake was not successful as one of the issuing certificates in the certificate path is not a CA certificate due to which the TOE was unable to construct a valid certificate path which resulted in Certificate Unknown error returned to the server. • The evaluator further observed the debug logs on the TOE at /opt/HelpSystems/GoAnywhere/userdata/logs to confirm that the TOE was unable to construct a valid chain and the certification path could not be validated as the certificate presented was not a CA certificate. Attempt a connection from the TOE to the server by omitting the basicConstraints field in one of the issuing certificates presented by the server. • The evaluator used the XCA tool to ensure that one of the issuing certificates (ICA2_nbc) in the certificate path does not have the basicConstraints by transforming the original ICA2 issuing certificate to an ICA_nbc omitting the basicConstraints field. • The evaluator ensured that the Self signed CA certificate is present in the TOE's trust store (system keyvault). • The evaluator uploaded the certificate in pem chain format containing both the ICA1 and ICA2_nbc to the mysql directory of the database server.

- The evaluator ensured that the certificate in pem chain format contains both ICA1 and ICA2_nbc.
- The evaluator ensured that the server certificate 10.1.3.51.crt and the server key 10.1.3.51_key.pem is present in the mysql directory of the database server.
- The evaluator configured the MySQL database server to leverage the uploaded server certificates and the server key for the TLS handshake with the client.
- The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the client could not connect to the server as the server returned a certificate_unknown error to the client.
- The evaluator observed the packet capture on the server and confirmed that the TLS handshake was not successful. Note that the ICA2 certificate does not contain basicConstraints in the extension field.
- The evaluator further observed the debug logs on the TOE at /opt/HelpSystems/GoAnywhere/userdata/logs to confirm that the TOE was unable to construct a valid chain and the certification path could not be validated as the Intermediate certificate lacks basic constraints.

Attempt a connection from the TOE to the server by setting the basicConstraints field in an issuing certificate to have CA=false

- The evaluator ensured that the Self signed CA certificate is present in the TOE's trust store (system keyvault).
- The evaluator exported the ICA2.crt file from the entire certificate chain that was created using XCA tool.
- The evaluator used the acumen x509-mod tool to modify the original ICA2.crt certificate file and output a modified ICA2_fbc.crt certificate file with BasicConstrants field set to false as per the test requirement. The evaluator then verified that the modified certificate has the correct subject and was signed by the correct certificate authority that created using XCA tool.
- The evaluator viewed the modified certificate ICA2_fbc.crt
- The evaluator created a single PEM encoded file with ICA1 and ICA2_fbc that can be presented to the client for certificate path validation.
- The evaluator verified that the pem encoded certificate file created previously have the BasicConstraints field set to false.
- The evaluator ensured that the server certificate 10.1.3.51.crt and the server key 10.1.3.51_key.pem is present in the mysql directory of the database server.
- The evaluator configured the MySQL database server to leverage the uploaded server certificates and the server key for the TLS handshake with the client.
- The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the client could not connect to the server as the server returned a certificate_unknown error to the client.
- The evaluator observed the packet capture on the server and confirmed that the TLS handshake was not successful. Note that the ICA2 certificate has the basicConstraints in the extension field set to false.
- The evaluator further observed the debug logs on the TOE at /opt/HelpSystems/GoAnywhere/userdata/logs to confirm that the TOE was

unable to construct a valid chain and the certification path could not be validated as the certificate presented was not a CA certificate.

Attempt a connection from the TOE to the server by omitting a CA signing bit of the key usage field

- The evaluator used the XCA tool to ensure that one of the issuing certificates (ICA2_Nosigbit) in the certificate path does not have a CA signing bit in the key usage field by transforming the original ICA2 issuing certificate to ICA2_Nosigbit certificate.
- The evaluator ensured that the Self signed CA certificate is present in the TOE's trust store (system keyvault).
- The evaluator uploaded the certificate encoded in pem chain format containing 10.1.3.51.crt, ICA1.crt and ICA2_Nosigbit.crt to the mysql directory of the database server.
- The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the client could not connect to the server as the server returned a certificate_unknown error to the client.
- The evaluator used the acumen-tlsc-mysql tool as a TLS server waiting for connection on IP address 10.1.3.51 and port 3307 and observed a fatal alert : certificate_unknown error when the client attempted a TLS connection.
- The evaluator observed the packet capture on the server and confirmed that the TLS handshake was not successful. Note that the ICA2 certificate has the Certificate signing bit set to false.
- The evaluator further observed the debug logs on the TOE at /opt/HelpSystems/GoAnywhere/userdata/logs to confirm that the TOE was unable to construct a valid chain and the certification path could not be validated as the issuer certificate keyusage extension is critical and does not permit key signing.

Attempt a connection from the TOE to the server by setting the path length field of a valid CA field to a value strictly lesser than the certificate path

- Set the pathlength field of RootCA certificate to 1
- Set pathlength field of IntCA1 certificate and IntCA2 certificate to zero
- The evaluator used the XCA tool to ensure that one of the issuing certificates (ICA1) in the certificate path has the path length field set to a value 0 that is strictly lesser than the certificate path. i.e., a CA with a path length constraint of zero cannot have any subordinate CAs. However, the ICA1 has a subordinate ICA2 while the path length is set to 0.
- The evaluator ensured that the Self signed CA certificate is present in the TOE's trust store (system keyvault).
- The evaluator uploaded the ICA1_pathlen.crt to the server and ensured that the path length is set 0.
- The evaluator uploaded a pem encoded chain consisting of ICA1_pathlen.crt and ICA2.crt to the mysql directory of the database server.
- The evaluator ensured that the server certificate 10.1.3.51.crt and the server key 10.1.3.51_key.pem is present in the mysql directory of the database server.

- The evaluator configured the MySQL database server to leverage the uploaded server certificates and the server key for the TLS handshake with the client.
- The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the client could not connect to the server as the server returned a certificate_unknown error to the client.
- The evaluator observed the packet capture on the server and confirmed that the TLS handshake was not successful. Note that the ICA1 certificate has the path length set to 0.
- The evaluator further observed the debug logs on the TOE at /opt/HelpSystems/GoAnywhere/userdata/logs to confirm that the TOE was unable to construct a valid chain and the certification path could not be validated as the max path length is not greater than zero.

The evaluator shall then establish a valid certificate path consisting of valid CA certificates and demonstrate that the function succeeds.
Note: The following test performed satisifies the FCS_TLSC_EXT.1.3 Test #1 requirement where the evaluator demonstrated that a server using a certificate with a valid certification path establishes a successful TLS handshake.

- The evaluator created a chain of four certificates using the XCA tool:
- The node certificate to be tested : 10.1.3.51 is the database server certificate loaded to the TLS server
- Two Intermediate Cas : ICA1 and ICA2 are the intermediate certificate authorities which are loaded to the TLS server. The ICA2.pem certificate file consists of both the ICA1 and ICA2 certificates.
- The self-signed Root CA : CA is the self-signed CA certificate which is loaded to the TOE or the TLS client to the database server.
- The evaluator ensured that the Self signed CA certificate is present in the TOE's trust store (system keyvault).
- The evaluator uploaded the certificate in pem chain format containing both the ICA1 and ICA2 to the mysql directory of the database server.
- The evaluator ensured that the server certificate 10.1.3.51.crt and the server key 10.1.3.51_key.pem is present in the mysql directory of the database server.
- The evaluator configured the MySQL database server to leverage the uploaded server certificates and the server key for the TLS handshake with the client.
- The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the TLS connection successfully established as the resource test was successful.
- The evaluator observed the packet capture on the server and confirmed that the TLS handshake was successful with valid certification path.

Attempt a connection from the TOE to the server by removing trust in one of the CA.
Note: The following test performed satisifies the FCS_TLSC_EXT.1.3 Test #1c requirement where the evaluator demonstrated that modifying the trust store element to be untrusted (by deleting the Intermediate certificate from the trust store) and attempting a connection from the server resulted in an authentication failure as the TOE

	was unable to construct a valid certificate path reason being one of the issuing certificates in the certificate path is not present in the TOE's trust store. • The evaluator then removed the trust ICA1 from the pem encoded chain and presented only the ICA2 certificate that signed the server certificate from the server side. • The evaluator ensured that only the Self signed CA certificate is present in the TOE's trust store (system keyvault)) and modified the trust store to confirm that the ICA1 was not present in the trust store which makes it untrusted. • The evaluator configured the MySQL database server to leverage the uploaded server certificate, server key and present only the ICA2 certificate in the capath for the TLS handshake with the client. • The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the client could not connect to the server as the server returned a certificate_unknown error to the client. • The evaluator observed the packet capture on the server and confirmed that the TLS handshake was not successful. Note that the ICA1 certificate was not presented by the server in the certificate path. • The evaluator further observed the debug logs on the TOE at /opt/HelpSystems/GoAnywhere/userdata/logs to confirm that the TOE was unable to construct a valid chain as no issuer certificate in the certification path was found.
Pass/Fail with Explanation	Pass. The TOE will not validate a certificate without a valid certification path but it will accept that same certificate when it has the valid Certificate chain. This meets the testing requirements.

6.9.2 FIA_X509_EXT.1.1 Test #2

Item	Data
Test Assurance Activity	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. Test 2: The evaluator shall demonstrate that validating an expired certificate results in the function failing. **TD0668 has been applied.**
Test Steps	Note: The following test performed satisifies the FCS_TLSC_EXT.1.3 Test #3 requirement where the evaluator demonstrated that server using a certificate which has passed its expiration date results in an authentication failure which was confirmed through packet capture where the client could not connect to the server as the server returned a certificate_unknown error to the client and also confirmed through TOE logs that the certificate expired on 20210212050000GMT+00:00. which corresponds to the server certificate.

	• The evaluator created a chain of four certificates using the XCA tool. • The evaluator used the XCA tool to create an expired certificate that expired on February 12, 2021 12:00:00 AM EST. • The evaluator ensured that the Self signed CA certificate is present in the TOE's trust store (system keyvault). • The evaluator uploaded the certificate in pem chain format containing both the ICA1 and ICA2 to the mysql directory of the database server. • The evaluator ensured that the server certificate expired on February 12, 2021 12:00:00 AM EST and uploaded the expired certificate to the mysql directory of the database server. • The evaluator checked the current date and time on the database server and ensured that the certificate expired as per the current time. • The evaluator checked the current date and time on the TOE Platform and ensured that the certificate expired as per the current time. • The evaluator configured the MySQL database server to leverage the uploaded expired server certificate, ICA2.pem and the server key for the TLS handshake with the client. • The evaluator configured the TOE to reach the Database server at Resources > Database Servers where the JDBC Driver was set to mariadb.jdbc .driver as the server was a MariaDB SQL server and provided the necessary URL information to reach the database server along with username and password of the database created on the server. GADATA is the database created on the MySQL server for the TOE on the Linux Platform. • The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the client could not connect to the server as the server returned a certificate_unknown error to the client. • The evaluator observed the packet capture on the server and confirmed that the TLS handshake was not successful. • The evaluator further observed the debug logs on the TOE at /opt/HelpSystems/GoAnywhere/userdata/logs to confirm that the certificate expired on 20210212050000GMT+00:00.
Pass/Fail with Explanation	Pass. The TOE does not validate an expired certificate and the TLS connection failed. This meets the testing requirements.

6.9.3 FIA_X509_EXT.1.1 Test #3

Item	Data
Test Assurance Activity	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created.

	Test 3: The evaluator shall test that the TOE can properly handle revoked certificates- "conditional on whether CRL, OCSP, OCSP Stapling or OCSP Multi-stapling is selected; if multiple methods are selected, then the following tests shall be performed for each method: The evaluator shall test revocation of the node certificate. The evaluator shall also test revocation of an intermediate CA certificate (i.e. the intermediate CA certificate should be revoked by the root CA), if intermediate CA certificates are supported. If OCSP stapling per RFC 6066 is the only supported revocation method, this test is omitted. The evaluator shall ensure that a valid certificate is used, and that the validation function succeeds. The evaluator then attempts the test with a certificate that has been revoked (for each method chosen in the selection) to ensure when the certificate is no longer valid that the validation function fails. **TD0668 has been applied.**
Test Steps	Note: The following test performed satisifies the FCS_TLSC_EXT.1.3 Test #2 and FIA_X509_EXT.2.2 Test #2 requirement where the evaluator demonstrated that the server using a certificate which has been revoked results in an authentication failure which was confirmed through packet capture where the client could not connect to the server as the server returned a certificate_unknown error to the client and also confirmed through TOE logs that the Certificate 82:F7:34:04:5D:C4:24:C3:64:0E:A1:16:2E:16:2B:04:0D:32:CB:C2 has been revoked by CRL at 'http://10.1.3.51/ICA2.crl' which corresponds to the server certificate. • The evaluator shall test revocation of the node certificate. • The evaluator created a chain of four certificates using the XCA tool: • The node certificate to be tested : 10.1.3.51 is the database server certificate loaded to the TLS server • Two Intermediate Cas : ICA1 and ICA2 are the intermediate certificate authorities which are loaded to the TLS server. The ICA2.pem certificate file consists of both the ICA1 and ICA2 certificates. • The self-signed Root CA : CA is the self-signed CA certificate which is loaded to the TOE's trust store. • The evaluator used the XCA tool and revoked the server certificate. • The evaluator generated CRLs using the xca tool. • The evaluator uploaded the CRLs in the html directory of the web server and ensured that there are no revoked certificates in CA.crl, ICA1.crl and have the server certificate as revoked in ICA2.crl • The evaluator ensured that the Self signed CA certificate is present in the TOE's trust store (system keyvault). • The evaluator uploaded the certificate in pem chain format containing both the ICA1 and ICA2 to the mysql directory of the database server. • The evaluator uploaded the revoked server certificate 10.1.3.51.crt and the server key 10.1.3.51_key.pem in the mysql directory of the database server. • The evaluator configured the MySQL database server to leverage the uploaded server certificates and the server key for the TLS handshake with the client.

	• The evaluator ensured to have the CRL check enabled for server certificates and specified the web server's URL information to fetch all the CRLs from the web server. • The evaluator configured the TOE to reach the Database server at Resources > Database Servers where the JDBC Driver was set to mariadb.jdbc .driver as the server was a MariaDB SQL server and provided the necessary URL information to reach the database server along with username and password of the database created on the server. GADATA is the database created on the MySQL server for the TOE on the Linux Platform. • The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the client could not connect to the server as the server returned a certificate_unknown error to the client. • The evaluator observed the packet capture on the server and verified that the client fetched the CRLs required to validate the certificates and ensured to return a fatal alert to the server. • The evaluator further observed the debug logs on the TOE at /opt/HelpSystems/GoAnywhere/userdata/logs to confirm that the Certificate 82:F7:34:04:5D:C4:24:C3:64:0E:A1:16:2E:16:2B:04:0D:32:CB:C2 has been revoked by CRL at 'http://10.1.3.51/ICA2.crl' which corresponds to the server certificate. • The evaluator used the XCA tool and unrevoked the server certificate. The evaluator then revoked the ICA2 certificate that was signed by its root certificate authority ICA1. • The evaluator generated CRLs using the xca tool. • The evaluator uploaded the CRLs in the html directory of the web server and ensured that there are no revoked certificates in CA.crl, ICA2.crl and have the ICA2 certificate as revoked in ICA1.crl. • The evaluator restarted the apache2 webserver to ensure the TOE fetches the updated CRLs. • The evaluator configured the MySQL database server to leverage the uploaded server certificates and the server key for the TLS handshake with the client. • The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the client could not connect to the server as the server returned a certificate_unknown error to the client. • The evaluator observed the packet capture on the server and verified that the client fetched the CRLs required to validate the certificates and ensured to return a fatal alert to the server. • The evaluator further observed the debug logs on the TOE at /opt/HelpSystems/GoAnywhere/userdata/logs to confirm that the Certificate 85:0D:CD:92:6E:04:CA:1E:43:D9:D1:76:41:61:A6:B6:30:61:BF:74 has been revoked by CRL at 'http://10.1.3.51/ICA1.crl' which corresponds to the ICA2 certificate that was signed by its root authority ICA1.

	• The evaluator shall ensure that a valid certificate is used, and that the validation function succeeds. • The evaluator used the XCA tool and unrevoked the ICA2 certificate and ensured that there are no revoked certificates in the chain. • The evaluator generated CRLs using the xca tool. • The evaluator uploaded the CRLs in the html directory of the web server and ensured that there are no revoked certificates in CA.crl, ICA1.crl, ICA2.crl. • The evaluator restarted the apache2 webserver to ensure the TOE fetches the updated CRLs. • The evaluator configured the MySQL database server to leverage the uploaded server certificates and the server key for the TLS handshake with the client. • The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the client was able to successfully communicate with the database server. • The evaluator observed the packet capture on the server and verified that the client fetched the CRLs required to validate the certificates and ensured that the TLS handshake was successful.
Pass/Fail with Explanation	Pass. The evaluator ensured when the node certificate or intermediate CA certificate is revoked and no longer valid, then the TLS handshake fails, and the validation function fails. This meetings the testing requirements.

6.9.4 FIA_X509_EXT.1.1 Test #4

Item	Data
Test Assurance Activity	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. Test 4 If any OCSP option is selected, the evaluator shall **ensure the TSF has no other source of revocation information available and** configure the OCSP server or use a man-in-the-middle tool to present **an OCSP response signed by** a certificate that does not have the OCSP signing purpose and **which is the only source of revocation status information advertised by the CA issuing the certificate being validated. The evaluator shall** verify that validation of the OCSP response fails **and that the TOE treats the certificate being checked as invalid and rejects the connection**. If CRL is selected, the evaluator shall **likewise** configure the CA **to be the only source of revocation status information, and** to sign a CRL with a certificate that does not have the cRLsign key usage bit set. **The evaluator shall** verify that validation of the CRL fails **and that the TOE treats the certificate being checked as invalid and rejects the connection**. ***Note: The intent of this test is to ensure a TSF does not trust invalid revocation status information. A TSF receiving invalid revocation status information from the only advertised certificate status provider should treat the certificate whose status is being checked as invalid. This should generally be treated differently from the case***

	where the TSF is not able to establish a connection to check revocation status information, but it is acceptable that the TSF ignore any invalid information and attempt to find another source of revocation status (another advertised provider, a locally configured provider, or cached information) and treat this situation as not having a connection to a valid certificate status provider. **TD0668 has been applied.**
Test Steps	• The evaluator created a chain of four certificates using the XCA tool: The node certificate to be tested : 10.1.3.51 is the database server certificate loaded to the TLS server Two Intermediate Cas : ICA1 and ICA2_noCRLsig are the intermediate certificate authorities which are loaded to the TLS server. The ICA2_noCRLsig.pem certificate file consists of both the ICA1 and ICA2_noCRLsig certificates. The ICA2_noCRLsig certificate does not have the cRLsign key usage bit set. The self-signed Root CA : CA is the self-signed CA certificate which is loaded to the TOE or the TLS client to the database server. • The evaluator generated the CRLs using the XCA tool and ensured that the ICA2.crl was signed by the ICA2_noCRLsig certificate that does not have the cRLsign key usage bit set. • The evaluator uploaded the CRLs in the html directory of the web server and ensured that there are no revoked certificates in CA.crl, ICA1.crl, ICA2.crl. • The evaluator ensured that the Self signed CA certificate is present in the TOE's trust store (system keyvault). • The evaluator uploaded the certificate in pem chain format containing both the ICA1 and ICA2_noCRLsig to the mysql directory of the database server. The evaluator ensured the certificate does not have the cRLsign key usage bit set. • The evaluator uploaded the revoked server certificate 10.1.3.51.crt and the server key 10.1.3.51_key.pem in the mysql directory of the database server. • The evaluator configured the MySQL database server to leverage the uploaded server certificate and the server key for the TLS handshake with the client and specified the ICA2_noCRLsig.pem chain as the CA. • The evaluator ensured to have the CRL check enabled for server certificates and specified the web server's URL information to fetch all the CRLs from the web server. • The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the client could not connect to the server as the server returned a certificate_unknown error to the client. NOTE: When CRL check is enabled and the certificate signing the CRL does not have a crlSign bit enabled, the TOE fails to validate the CRL and the TLS connection fails. • The evaluator observed the packet capture on the server and ensured verified that the client fetched the CRLs required to validate the certificates but the client ensured to return a fatal alert to the server as the TOE fails to validate the CRL that was signed by the certificate which does not have the crlSign bit enabled.

	• The evaluator further observed the debug logs on the TOE at /opt/HelpSystems/GoAnywhere/userdata/logs to confirm that the TLS handshake was not successful as the certificate's CA does not contain the crlSign bit.
Pass/Fail with Explanation	Pass. The connection is rejected by the TOE which fails to validate the CRL because it is signed by a certificate not containing the cRLSign bit. This meets the testing requirements.

6.9.5 FIA_X509_EXT.1.1 Test #5

Item	Data
Test Assurance Activity	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. Test 5: The evaluator shall modify any byte in the first eight bytes of the certificate and demonstrate that the certificate fails to validate. (The certificate will fail to parse correctly.) **TD0668 has been applied.**
Test Steps	• The evaluator created a chain of four certificates using the XCA tool: The node certificate to be tested : 10.1.3.51 is the database server certificate presented during the TLS handshake. Two Intermediate Cas : ICA1 and ICA2 are the intermediate certificate authorities. The 10.1.3.51.pem certificate file consists of the ICA1, ICA2 and 10.1.3.51 certificates in pem encoded chain format. The self-signed Root CA : CA is the self-signed CA certificate which is loaded to the TOE's trust store. • The evaluator ensured that the Self signed CA certificate is present in the TOE's trust store (system keyvault). • The evaluator uploaded the server certificate which is encoded in pem chain format along with ICA1 and ICA2 certificates and the server key 10.1.3.51_key.pem in the mysql directory of the database server. • The evaluator used the "acumen-tlsc-mysql" tool as a server waiting for TLS connections on IP address 10.1.3.51 and port 3307 presenting the certificate chain 10.1.3.51.pem as the server certificate and the server key 10.1.3.51_key.pem with ID 15 that corresponds to the current test. The evaluator observed the tool output and ensured that a fatal alert was returned to the server after the server presented a certificate with first 8 bytes modified. • The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the client could not connect to the server as the server returned a certificate_unknown error to the client.

	• The evaluator observed the packet capture on the server and ensured that the server presented a certificate with the first 8 modified while the client returned a fatal alert to the server.
Pass/Fail with Explanation	Pass. The evaluator modified the first eight bytes of the certificate being presented by the server and ensured that the certificate fails to validate, and the TLS handshake fails. This meets the testing requirements.

6.9.6 FIA_X509_EXT.1.1 Test #6

Item	Data
Test Assurance Activity	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. Test 6: The evaluator shall modify any byte in the last byte of the certificate and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.) **TD0668 has been applied.**
Test Steps	• The evaluator created a chain of four certificates using the XCA tool: The node certificate to be tested : 10.1.3.51 is the database server certificate presented during the TLS handshake. Two Intermediate Cas : ICA1 and ICA2 are the intermediate certificate authorities. The 10.1.3.51.pem certificate file consists of the ICA1, ICA2 and 10.1.3.51 certificates in pem encoded chain format. The self-signed Root CA : CA is the self-signed CA certificate which is loaded to the TOE's trust store. • The self-signed Root CA : CA is the self-signed CA certificate which is loaded to the TOE or the TLS client to the database server. • The evaluator ensured that the Self signed CA certificate is present in the TOE's trust store (system keyvault). • The evaluator uploaded the server certificate 10.1.3.51.pem which is encoded in pem chain format along with ICA1 and ICA2 certificates and the server key 10.1.3.51_key.pem in the mysql directory of the database server. • The evaluator used the "acumen-tlsc-mysql" tool as a server waiting for TLS connections on IP address 10.1.3.51 and port 3307 presenting the certificate chain 10.1.3.51.pem as the server certificate and the server key 10.1.3.51_key.pem with ID 16 that corresponds to the current test. The evaluator observed the tool output and ensured that a fatal alert was returned to the server after the server presented a certificate with the last byte modified. • The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the client could not connect to the server as the server returned a certificate_unknown error to the client.

	• The evaluator observed the packet capture on the server and ensured that the server presented a certificate with the last byte modified while the client returned a fatal alert to the server.
Pass/Fail with Explanation	Pass. The evaluator modified the last byte of the certificate and demonstrated that the certificate fails to validate. This meets the testing requirements.

6.9.7 FIA_X509_EXT.1.1 Test #7

Item	Data
Test Assurance Activity	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. Test 7: The evaluator shall modify any byte in the public key of the certificate and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.) **TD0668 has been applied.**
Test Steps	• The evaluator created a chain of four certificates using the XCA tool: The node certificate to be tested : 10.1.3.51 is the database server certificate presented during the TLS handshake. Two Intermediate Cas : ICA1 and ICA2 are the intermediate certificate authorities. The 10.1.3.51.pem certificate file consists of the ICA1, ICA2 and 10.1.3.51 certificates in pem encoded chain format. • The self-signed Root CA : CA is the self-signed CA certificate which is loaded to the TOE's trust store. • The self-signed Root CA : CA is the self-signed CA certificate which is loaded to the TOE or the TLS client to the database server. • The evaluator ensured that the Self signed CA certificate is present in the TOE's trust store (system keyvault). • The evaluator uploaded the server certificate 10.1.3.51.pem which is encoded in pem chain format along with ICA1 and ICA2 certificates and the server key 10.1.3.51_key.pem in the mysql directory of the database server. • The evaluator used the "acumen-tlsc-mysql" tool as a server waiting for TLS connections on IP address 10.1.3.51 and port 3307 presenting the certificate chain 10.1.3.51.pem as the server certificate and the server key 10.1.3.51_key.pem with ID 17 that corresponds to the current test. The evaluator observed the tool output and ensured that a fatal alert was returned to the server after the server presented a certificate with a modified public key. • The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the client could not connect to the server as the server returned a certificate_unknown error to the client.

	• The evaluator observed the packet capture on the server and ensured that the server presented a certificate with the public key modified while the client returned a fatal alert to the server.
Pass/Fail with Explanation	Pass. The TOE fails to validate a modified server certificate. This meets the testing requirements.

6.9.8 FIA_X509_EXT.1.1 Test #8a

Item	Data
Test Assurance Activity	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. Test 8a: (Conditional on support for EC certificates as indicated in FCS_COP.1(3)). The evaluator shall establish a valid, trusted certificate chain consisting of an EC leaf certificate, an EC Intermediate CA certificate not designated as a trust anchor, and an EC certificate designated as a trusted anchor, where the elliptic curve parameters are specified as a named curve. The evaluator shall confirm that the TOE validates the certificate chain. **TD0668 has been applied.**
Test Steps	• The evaluator established a valid, trusted certificate chain consisting of an EC leaf server certificate (10.1.3.51_ec), an EC Intermediate CA certificate (ICA_ec) not designated as a trust anchor, and an EC certificate (CA_ec) designated as a trusted anchor, where the elliptic curve parameters are specified as a named curve using the XCA tool. • The evaluator ensured that the CA_ec certificate was configured with the EC elliptic curve parameters specified as a named curve. • The evaluator uploaded the ICA_ec certificate to the mysql directory of the database server and ensured that it was configured with the EC elliptic curve parameters specified as a named curve. • The evaluator uploaded the 10.1.3.51_ec.crt certificate to the mysql directory of the database server and ensured that it was configured with the EC elliptic curve parameters specified as a named curve. • The evaluator ensured that the Self signed CA_ec certificate is present in the TOE's trust store (system keyvault). • The evaluator configured the MySQL database server to leverage the uploaded server certificate, ICA_ec and the ec server key for the TLS handshake with the client. • The evaluator configured the TOE to reach the Database server at Resources > Database Servers where the JDBC Driver was set to mariadb.jdbc .driver as the server was a MariaDB SQL server and provided the necessary URL information to reach the database server along with username and password of the database created on the server. WADATA is the database created on the MySQL server for the TOE on the Windows Platform.

	• The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the client successfully communicated with the data base server with the EC certificates configured. • The evaluator observed the packet capture and confirmed that the TOE validates the EC certificate chain where the elliptic curve parameters are specified as a named curve.
Pass/Fail with Explanation	Pass. The TOE successfully validates an EC certificate chain when the elliptic curve parameters are specified as a named curve. This meets the testing requirements.

6.9.9 FIA_X509_EXT.1.1 Test #8b

Item	Data
Test Assurance Activity	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. Test 8b: (Conditional on support for EC certificates as indicated in FCS_COP.1(3)). The evaluator shall replace the intermediate certificate in the certificate chain for Test 8a with a modified certificate, where the modified intermediate CA has a public key information field where the EC parameters uses an explicit format version of the Elliptic Curve parameters in the public key information field of the intermediate CA certificate from Test 8a, and the modified Intermediate CA certificate is signed by the trusted EC root CA, but having no other changes. The evaluator shall confirm the TOE treats the certificate as invalid. **TD0668 has been applied.**
Test Steps	• The evaluator established a valid, trusted certificate chain consisting of an EC leaf server certificate (10.1.3.51_ec), an EC Intermediate CA certificate (ICA_ec) not designated as a trust anchor, and an EC certificate (CA_ec) designated as a trusted anchor, where the elliptic curve parameters are specified as a named curve using the XCA tool. • The evaluator ensured that the CA_ec certificate was configured with the EC elliptic curve parameters specified as a named curve. • The evaluator uploaded the ICA_ec certificate to the mysql directory of the database server and ensured that it was configured with the EC elliptic curve parameters specified as a named curve. • The evaluator uploaded the 10.1.3.51_ec.crt certificate to the mysql directory of the database server and ensured that it was configured with the EC elliptic curve parameters specified as a named curve. • The evaluator ensured that the Self signed CA_ec certificate is present in the TOE's trust store (system keyvault). • The evaluator used the acumen x509-mod tool to modify the original ICA_ec.crt certificate file where the elliptic curve parameters are specified as a

	named curve and output a modified ICA_ec_mod.crt certificate file that has a public key information field where the EC parameters uses an explicit format version of the Elliptic Curve parameters in the public key information field as per the test requirement. The evaluator then verified that the modified certificate has the correct subject and was signed by the correct certificate authority that created using XCA tool. • The evaluator ensured the modified Intermediate CA certificate uses an explicit format version of the Elliptic Curve parameters in the public key information field. • The evaluator configured the MySQL database server to leverage the uploaded server certificate, ICA_ec_mod and the ec server key for the TLS handshake with the client. • The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the client could not connect to the server as the server returned a certificate_unknown error to the client. • The evaluator observed the packet capture on the server and confirmed that the TLS handshake was not successful. • The evaluator further observed the debug logs on the TOE at /opt/HelpSystems/GoAnywhere/userdata/logs to confirm that only named elliptic curves are allowed by the TOE while the certificate for 'ICA' contains an implicit or specified curve.
Pass/Fail with Explanation	Pass. The evaluator confirmed that the TOE treats the intermediate CA that has a public key information field where the EC parameters uses an explicit format version of the Elliptic Curve parameters in the public key information field as invalid. This meets the testing requirements.

6.9.10 FIA_X509_EXT.1.2 Test #1

Item	Data
Test Assurance Activity	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: - The node certificate to be tested, - Two Intermediate CAs, and - The self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. The evaluator shall ensure that the certificate of at least one of the CAs in the chain does not contain the basicConstraints extension. The evaluator shall confirm that validation of the certificate path fails: (i) as part of the validation of the peer certificate belonging to this chain; and/or

	(ii) when attempting to add the CA certificate without the basicConstraints extension to the TOE's trust store. **TD0495 has been applied.**
Test Steps	• The evaluator created a chain of four certificates using the XCA tool. • The evaluator used the XCA tool to ensure that one of the issuing certificates (ICA2_nbc) in the certificate path does not have the basicConstraints by transforming the original ICA2 issuing certificate to a ICA_nbc omitting the basicConstraints field. • The evaluator ensured that the Self signed CA certificate is present in the TOE's trust store (system keyvault). • The evaluator uploaded the certificate in pem chain format containing both the ICA1 and ICA2_nbc to the mysql directory of the database server. • The evaluator ensured that the certificate in pem chain format contains both ICA1 and ICA2_nbc. • The evaluator ensured that the server certificate 10.1.3.51.crt and the server key 10.1.3.51_key.pem is present in the mysql directory of the database server. • The evaluator configured the MySQL database server to leverage the uploaded server certificates and the server key for the TLS handshake with the client. • The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the client could not connect to the server as the server returned a certificate_unknown error to the client. • The evaluator observed the packet capture on the server and confirmed that the TLS handshake was not successful. Note that the ICA2 certificate does not contain basicConstraints in the extension field. • The evaluator further observed the debug logs on the TOE at /opt/HelpSystems/GoAnywhere/userdata/logs to confirm that the TOE was unable to construct a valid chain and the certification path could not be validated as the Intermediate certificate lacks basic constraints.
Pass/Fail with Explanation	Pass. The TOE fails to validate a certificate with no basicConstraints section and rejects it. This meets the testing requirements.

6.9.11 FIA_X509_EXT.1.2 Test #2

Item	Data
Test Assurance Activity	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: - The node certificate to be tested, - Two Intermediate CAs, and - The self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created.

	The evaluator shall ensure that the certificate of at least one of the CAs in the chain has the CA flag in the basicConstraints extension not set (or set to FALSE). The evaluator shall confirm that validation of the certificate path fails (i) as part of the validation of the peer certificate belonging to this chain; and/or (ii) when attempting to add the CA certificate with the CA flag not set (or set to FALSE) in the basicConstraints extension to the TOE's trust store **TD0495 has been applied.**
Test Steps	• The evaluator created a chain of four certificates using the XCA tool. • The evaluator ensured that the Self signed CA certificate is present in the TOE's trust store (system keyvault). • The evaluator exported the ICA2.crt file from the entire certificate chain that was created using XCA tool. • The evaluator used the acumen x509-mod tool to modify the original ICA2.crt certificate file and output a modified ICA2_fbc.crt certificate file with BasicConstrants field set to false as per the test requirement. The evaluator then verified that the modified certificate has the correct subject and was signed by the correct certificate authority that created using XCA tool. • The evaluator viewed the modified certificate ICA2_fbc.crt • The evaluator created a single PEM encoded file with ICA1 and ICA2_fbc that can be presented to the client for certificate path validation. • The evaluator verified that the pem encoded certificate file created previously have the BasicConstraints field set to false. • The evaluator ensured that the server certificate 10.1.3.51.crt and the server key 10.1.3.51_key.pem is present in the mysql directory of the database server. • The evaluator configured the MySQL database server to leverage the uploaded server certificates and the server key for the TLS handshake with the client. • The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the client could not connect to the server as the server returned a certificate_unknown error to the client. • The evaluator observed the packet capture on the server and confirmed that the TLS handshake was not successful. Note that the ICA2 certificate has the basicConstraints in the extension field set to false. • The evaluator further observed the debug logs on the TOE at /opt/HelpSystems/GoAnywhere/userdata/logs to confirm that the TOE was unable to construct a valid chain and the certification path could not be validated as the certificate presented was not a CA certificate.
Pass/Fail with Explanation	Pass. CA Certificates with the basicConstraints flag set to false are rejected by the TOE. This meets the testing requirements.

6.9.12 FIA_X509_EXT.1.2 Test #3

TD0495 removes this test.

6.9.13 FIA_X509_EXT.2.2 Test #1

Item	Data

Test Assurance Activity	The evaluator shall perform the following test for each trusted channel: Test 1: The evaluator shall demonstrate that using a valid certificate that requires certificate validation checking to be performed in at least some part by communicating with a non-TOE IT entity. The evaluator shall then manipulate the environment so that the TOE is unable to verify the validity of the certificate and observe that the action selected in FIA_X509_EXT.2.2 is performed. If the selected action is administrator-configurable, then the evaluator shall follow the operational guidance to determine that all supported administrator-configurable options behave in their documented manner.
Test Steps	TOE as client: • The evaluator created a chain of four certificates using the XCA tool. • The evaluator uploaded the CRLs in the html directory of the web server and ensured that there are no revoked certificates in CA.crl, ICA1.crl, ICA2.crl. • The evaluator ensured that the Self signed CA certificate is present in the TOE's trust store (system keyvault). • The evaluator uploaded the certificate in pem chain format containing both the ICA1 and ICA2 to the mysql directory of the database server. • The evaluator ensured that the server certificate 10.1.3.51.crt and the server key 10.1.3.51_key.pem is present in the mysql directory of the database server. • The evaluator ensured to have the CRL check enabled for server certificates and specified the web server's URL information to fetch all the CRLs from the web server. • The evaluator configured the TOE to reach the Database server at Resources > Database Servers where the JDBC Driver was set to mariadb.jdbc .driver as the server was a MariaDB SQL server and provided the necessary URL information to reach the database server along with username and password of the database created on the server. WADATA is the database created on the MySQL server for the TOE on the Windows Platform. • The evaluator configured the MySQL database server to leverage the uploaded server certificates and the server key for the TLS handshake with the client. • The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the TLS connection successfully established as the resource test was successful. • The evaluator observed the packet capture on the server and verified that the client fetched the CRLs required to validate the certificates and ensured that the TLS handshake was successful. • The evaluator then manipulated the environment by shutting down the apache2 webserver on the non-TOE IT entity which the TOE is communicating with to verify the validity of the certificate. • The evaluator reattempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the TLS connection did not establish as the client returned a fatal alert "certificate unknown" to the server.

	• The evaluator observed the packet capture on the server and verified that the client was unable to fetch the CRLs required to validate the certificates and confirmed that the TLS connection did not establish. • The evaluator further observed the debug logs on the TOE at /opt/HelpSystems/GoAnywhere/userdata/logs to confirm the error was that the client was unable to fetch the Certificate Revocation List from URL 'http://10.1.3.51/CA.crl. TOE as server: • The evaluator used the XCA tool to create the required certificates. • The evaluator generated CRLs using the XCA tool. • The evaluator uploaded the CRLs in the html directory of the web server and ensured that there are no revoked certificates in CA.crl, ICA1_client.crl and have the server certificate as revoked in ICA2_client.crl • The evaluator ensured that the Self signed CA certificate is present in the TOE's trust store (system keyvault). • The evaluator uploaded the certificate in pem chain format containing both the ICA1_client and ICA2_client to the client VM. • The evaluator uploaded the server certificate 10.1.3.51_client.pem and the server key 10.1.3.51_client_key.pem in the Client VM. • The evaluator ensured to have the CRL check enabled for client certificates and specified the web server's URL information to fetch all the CRLs from the web server. • The evaluator ensured that the CA and ICA certificates that signed the server_x509 certificate are uploaded to the TOE' trust store. • The evaluator configured the TOE to use the server_x509 certificate for TLS connection used for remote administration. The evaluator also ensured that the client authentication is required. • The evaluator attempted a connection from the VM (TLS client) to the remote Admin server using the openssl s_client and confirmed that the client could connect to the server. • The evaluator observed the packet capture on the server and verified that the TOE fetched the CRLs required to validate the client certificates and ensured to establish a connection with the TOE. • The evaluator then manipulated the environment by shutting down the apache2 webserver on the non-TOE IT entity which the TOE is communicating with to verify the validity of the certificate. • The evaluator attempted a connection from the VM (TLS client) to the remote Admin server using the openssl s_client and confirmed that the client could not connect to the server. • The evaluator observed the packet capture on the client and verified that the server was unable to fetch the CRLs required to validate the certificates and confirmed that the TLS connection did not establish. • The evaluator further observed the debug logs on the TOE at /opt/HelpSystems/GoAnywhere/userdata/logs to confirm the error was that the client was unable to fetch the Certificate Revocation List from URL 'http://10.1.3.51/CA.crl.

Pass/Fail with Explanation	Pass. The evaluator verified that when the CRL server was online, the revocation check was successful. The evaluator then manipulated the connection and made the CRL server offline after the cached CRL has expired and the TOE failed to load the updated CRL's and validate the certificates. This meets the testing requirements.

6.9.14 FIA_X509_EXT.2.2 Test #2

Item	Data
Test Assurance Activity	The evaluator shall perform the following test for each trusted channel: Test 2: The evaluator shall demonstrate that an invalid certificate that requires certificate validation checking to be performed in at least some part by communicating with a non-TOE IT entity cannot be accepted.
Expected Results	As a part of FIA_X509_EXT.1.1 Test #3, • The evaluator observed the packet capture on the server and verified that the client fetched the CRLs required to validate the server certificate by communicating with a non-TOE IT entity (CRL web server) and ensured to return a fatal alert to the server as the server certificate was revoked. • The evaluator further observed the debug logs on the TOE at /opt/HelpSystems/GoAnywhere/userdata/logs to confirm that the Certificate 82:F7:34:04:5D:C4:24:C3:64:0E:A1:16:2E:16:2B:04:0D:32:CB:C2 has been revoked by CRL at 'http://10.1.3.51/ICA2.crl' which corresponds to the server certificate. (Added a Note in FIA_X509_EXT.1.1 Test #3 which satisfied the current requirement)
Pass/Fail with Explanation	Pass. This test is performed in conjunction with FIA_X509_EXT.1.1 Test #3 where the evaluator demonstrated that the validation check of the certificate was performed by communicating with a non-TOE IT entity (CRL web server) and ensured that it cannot be accepted as the certificate was deemed invalid (revoked).

6.9.15 FCS_HTTPS_EXT.1.3 /Client

Item	**Data**
Test Assurance Activity	The evaluator shall demonstrate that using a certificate without a valid certification path results in the selected action in the SFR. If "notify the user" is selected in the SFR, then the evaluator shall also determine that the user is notified of the certificate validation failure. Using the administrative guidance, the evaluator shall then load a certificate or certificates to the Trust Anchor Database needed to validate the certificate to be used in the function and demonstrate that the function succeeds. The evaluator then shall delete one of the certificates, and show that again, using a certificate without a valid certification path results in the selected action in the SFR, and if "notify the user" was selected in the SFR, the user is notified of the validation failure. **TD0668 has been applied.**
Test Steps	• The evaluator created the necessary certificates to perform the test using the XCA tool. • The evaluator created key vault named "https_client" that is required to setup the certificates that must be used by the TOE in Encryption > Key Management Services > Add Key vault.

	• The evaluator ensured that the key vault was created. • The evaluator uploaded the CA_client certificate to the TOE'S trust store (system keyvault) where the trusted root certificates are stored. • The evaluator uploaded the client.pem certificate in pem encoded format into the keyvault that was previously created on the TOE which will be presented as a client certificate during HTTPS/TLS handshake. • The evaluator ensured that the client certificate was installed on the TOE. • The evaluator added the HTTPS server resource where the details regarding the HTTPS server with which client have to communicate is configured. The evaluator specified the server host IP address as 10.1.3.51. • The evaluator set the server port to 443. • The evaluator selected the key vault "https_client" that was previously created and the client certificate that was uploaded on the TOE. • Note: The evaluator observed that Key Management system will be used for validating the HTTPS server's identity. • The evaluator saved the HTTPS server resource created. • The evaluator ensured that the CA_server certificate authority that signed the server certificate was not present in the TOE's trust store (Encryption > Key Management Services > System keyvault) where the trust certificates are stored. • The evaluator created a project in Workflows and configured the project to leverage the HTTPS resource that was previously created. The evaluator added a POST function under the HTTPS resource to send a text file to the HTTPS server. • The evaluator uploaded the server certificate and the key that was created on the Web server that is being used as a HTTPS server. • The evaluator ensured that the server certificate was signed by the CA_Server certificate authority. • The evaluator configured the server to use the uploaded certificate and key for TLS/HTTPS handshake and wait for a TLS connection on IP address 10.1.3.51 and port 443. • The evaluator initiated a TLS connection the send a file to the HTTPS server and ensured that the TLS/HTTPS connection failed due to certificate unknown error returned by the client to the server. • The evaluator observed the logs on the TOE to ensure that the TLS/HTTPS handshake was not successful due to certificate unknown error. • The evaluator further observed the stack trace to ensure that the TLS handshake failed as the TOE could not find the issuer certificate for the server certificate in certificate path. • The evaluator observed the packet capture on the server and ensured that the client returned a fatal alert: Certificate unknown to the server. • The evaluator then uploaded the trusted CA_server certificate that signed the server certificate to the Trust store (system keyvault). • The evaluator attempted the HTTPS/TLS connection to send the text file to the HTTPS server and ensured that the project executed the task with no errors.

	• The evaluator observed the logs and ensured that the task was successfully executed. • The evaluator observed the packet capture on the server to ensure that the TLS/HTTPS handshake with the HTTPS server was successful. • The evaluator then deleted the CA_server certificate authority that signed the server certificate. • The evaluator ensured that the CA_server certificate authority is not present in the TOE's trust store. (system keyvault) • The evaluator initiated a TLS connection the send a file to the HTTPS server and ensured that the TLS/HTTPS connection failed due to certificate unknown error returned by the client to the server. • The evaluator observed the logs on the TOE to ensure that the TLS/HTTPS handshake was not successful due to certificate unknown error. • The evaluator further observed the stack trace to ensure that the TLS handshake failed as the TOE could not find the issuer certificate for the server certificate in certificate path. • The evaluator observed the packet capture on the server and ensured that the client returned a fatal alert: Certificate unknown to the server.
Pass/Fail with Explanation	Pass. The HTTPS connection succeeds only when the TOE can successfully validate the certificate chain. This meets the test requirements.

6.9.16 FCS_HTTPS_EXT.2/HTTPS with Mutual authentication

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that using a certificate without a valid certification path results in the selected action in the SFR. Using the administrative guidance, the evaluator shall then load a certificate or certificates to the Trust Anchor Database needed to validate the certificate to be used in the function and demonstrate that the function succeeds. The evaluator then shall delete one of the certificates, and show that again, using a certificate without a valid certification path results in the selected action in the SFR. **TD0668 has been applied.**
Test Steps	• The evaluator created a RootCA certificate that signed the server certificate (https_server) and also created a RootCA_client certificate that signed the client certificate using the XCA tool. • The evaluator ensured that the server certificate was signed by the RootCA. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator loaded the server certificate (https_server_253.pem) along with the key to the TOE's system keyvault. • The evaluator configured the TOE's Administration server in System > Admin server where the port was set 8001 in General. • The evaluator selected the TLS protocol, supported cipher suites, the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and saved the configuration. • The evaluator confirmed that the RootCA_client certificate that signed the client certificate was not uploaded to the TOE's system keyvault.

	• The evaluator uploaded the client certificate to the client VM and ensured that the certificate was signed by the RootCA_client certificate. • The evaluator initiated the TLS connection using the client certificate that was signed by the RootCA_client certificate which was not present in the TOE's trust store (system keyvault) and ensured that the TLS connection did not succeed due to certificate unknown error. • The evaluator observed the packet capture to ensure that the TLS handshake was not successful due to certificate unknown alert returned by the server after the client sent the certificate. • The evaluator further observed the logs located at /opt/HelpSystems/GoAnywhere/userdata/logs to ensure that the TLS handshake was unsuccessful as the TOE was unable to construct a valid chain and no issuer certificate for the client certificate in the certificate was found. • The evaluator then loaded the RootCA_client certificate authority needed to validate the client certificate to the TOE's trust store (system keyvault) in Encryption > Key Management services. • The evaluator confirmed that the certificate authority was imported successfully. • The evaluator initiated the TLS connection using the client certificate that was signed by the RootCA_client certificate which was imported in the TOE's trust store (system keyvault) and ensured that the TLS connection established successfully. • The evaluator observed the packet capture on the client VM to confirm that the TLS connection established successfully. • The evaluator then deleted the RootCA_client certificate authority from the TOE's certificate trust store (system keyvault) that signed the client's certificate. • The evaluator confirmed that the RootCA_client certificate authority was not present in the TOE's trust store. • The evaluator initiated the TLS connection using the client certificate that was signed by the RootCA_client certificate which was not present in the TOE's trust store (system keyvault) and ensured that the TLS connection did not succeed due to certificate unknown error. • The evaluator observed the packet capture to ensure that the TLS handshake was not successful due to certificate unknown alert returned by the server after the client sent the certificate. • The evaluator further observed the logs located at /opt/HelpSystems/GoAnywhere/userdata/logs to ensure that the TLS handshake was unsuccessful as no issuer certificate for the client certificate in the certificate was found.
Pass/Fail with Explanation	Pass. The evaluator verified that when the server was presented with a client certificate without having its issuer certificate in the TOE's trust store or the certification path resulted in failure to validate the certificate. This meets the testing requirements.

6.10 Filesystem (Windows)

6.10.1 FMT_CFG_EXT.1.2 Test #1

Item	Data
Test Assurance Activity	The evaluator shall install and run the application. The evaluator shall inspect the filesystem of the platform (to the extent possible) for any files created by the application and ensure that their permissions are adequate to protect them. The method of doing so varies per platform. For Windows: The evaluator shall run the SysInternals tools, Process Monitor and Access Check (or tools of equivalent capability, like icacls.exe) for Classic Desktop applications to verify that files written to disk during an application's installation have the correct file permissions, such that a standard user cannot modify the application or its data files. For Windows Universal Applications the evaluator shall consider the requirement met because of the AppContainer sandbox. TD0519 has been applied.
Test Steps	• The evaluator installed and run the application. During installation, the TOE added two folders as follows: C:\Program Files\HelpSystems C:\ProgramData\HelpSystems • The evaluator ran the tool Accesschk.exe and verified that the filesystem of the platform for any files created by the application have the correct file permissions. The below screenshots indicate that there are no files that can be modified by a user. • Search the folder (C:\Program Files\HelpSystems) hierarchy for all files that can be modified by Users. The below screenshots indicate that there are no files that can be modified by a user and the user has only read permissions (R). • Search the folder (C:\ProgramData\HelpSystems) hierarchy for all files that can be modified by Users. The below screenshots indicate that all the user configuration files (XML files) cannot be modified by a user and the user has only read permissions (R) to these configuration files.
Pass/Fail with Explanation	Pass. The files written to disk during the TOE's installation have the correct file permissions, such that a standard user cannot modify the application or its data files.

6.10.2 FMT_MEC_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	If "invoke the mechanisms recommended by the platform vendor for storing and setting configuration options" is chosen, the method of testing varies per platform as follows: For Windows: The evaluator shall determine and verify that Windows Universal Applications use either the Windows.Storage namespace, Windows.UI.ApplicationSettings namespace, or the IsolatedStorageSettings namespace for storing application specific settings. For .NET applications, the evaluator shall determine and verify that the application uses one of the locations listed in https://docs.microsoft.com/en-us/dotnet/framework/configure-apps/ for storing application specific settings.For Classic Desktop applications, the evaluator shall run the application while monitoring it with the SysInternals tool ProcMon and make changes to

	its configuration. The evaluator shall verify that ProcMon logs show corresponding changes to the Windows Registry or C:\ProgramData\ directory. Non-applicable platforms removed. TD0437 has been applied. TD0465 has been applied. TD0543 has been applied.
Test Steps	• The evaluator started the SysInternal tool ProcMon with the following input filter and observed no output. • The evaluator ran the application and observed the configuration at System -> Security Settings. • The evaluator updated the security settings by making the following highlighted changes in the Security Settings. • The evaluator verified that the ProcMon logs showing corresponding changes to the C:\ProgramData\HelpSystems. • The evaluator then observed the SFTP configuration at Services -> Service Manager while monitoring it with the ProcMon tool. • The evaluator updated the SFTP server configuration by updating the name from default to helpsystems and verified that the ProcMon logs showing corresponding changes to the C:\ProgramData\HelpSystems. • The evaluator then observed the HTTPS configuration at Services -> Service Manager. • The evaluator updated the HTTPS server configuration by updating the name from default to helpsystems and verified that the ProcMon logs showing corresponding changes to the C:\ProgramData\HelpSystems. • The evaluator verified that the ProcMon logs show corresponding changes to the C:\ProgramData\HelpSystems while making changes to the TOE's configuration.
Pass/Fail with Explanation	Pass. The evaluator verified that the ProcMon logs show corresponding changes to the C:\ProgramData\HelpSystems while making changes to the TOE's configuration. This meets the testing requirements.

6.10.3 FPT_AEX_EXT.1.4 Test #1

Item	Data
Test Assurance Activity	The evaluator shall run the application and determine where it writes its files. For files where the user does not choose the destination, the evaluator shall check whether the destination directory contains executable files. This varies per platform: For Windows: For Windows Universal Applications the evaluator shall consider the requirement met because the platform forces applications to write all data within the application working directory (sandbox). For Windows Desktop Applications the evaluator shall run the program, mimicking normal usage, and note where all user-modifiable files are written. The evaluator shall ensure that there are no executable files stored in the same directories to which the application wrote user-modifiable files. Non-applicable platforms removed. TD0445 has been applied.
Test Steps	• The evaluator identified the TOE executable as tomcat.exe located at C:\Program Files\HelpSystems\GoAnywhere\tomcat\bin.

	• The evaluator noted the PID of the TOE's executable running to be 7408. • The evaluator started the SysInternal tool ProcMon with the following input filter and observed no output. • The evaluator ran the application and observed the configuration at System -> Security Settings. • The evaluator updated the security settings mimicking normal usage by making the following highlighted changes in the Security Settings. • The evaluator verified that the ProcMon logs and noted that the user-modifiable files are written to the C:\ProgramData\HelpSystems. • The evaluator then observed the SFTP configuration at Services -> Service Manager while monitoring it with the ProcMon tool. • The evaluator updated the SFTP server configuration mimicking normal usage by updating the name from default to helpsystems. The evaluator observed that the ProcMon logs and noted that the user-modifiable files are written to the C:\ProgramData\HelpSystems. • The evaluator then observed the HTTPS configuration at Services -> Service Manager. • The evaluator updated the HTTPS server configuration mimicking normal usage by updating the name from default to helpsystems. The evaluator observed that the ProcMon logs and noted that the user-modifiable files are written to the C:\ProgramData\HelpSystems. • As observed, all the user modifiable files were written to the Helpsystems directory located at C:\ProgramData. The evaluator verified that the HelpSystems directory located at C:\ProgramData does not contain any executables. Also, the evaluator ensured that the tomcat directory containing executables do not have data files written to them.
Pass/Fail with Explanation	Pass. The evaluator ensured that all the user modifiable files were written to the Helpsystems directory located at C:\ProgramData. The evaluator verified that the HelpSystems directory located at C:\ProgramData does not contain any executables. Also, the evaluator ensured that the tomcat directory containing executables do not have data files written to them. This meets the testing requirements.

6.10.4 FPT_IDV_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	The evaluator shall install the application, then check for the existence of version information. If SWID tags is selected the evaluator shall check for a .swidtag file. The evaluator shall open the file and verify that is contains at least a SoftwareIdentity element and an Entity element.
Test Steps	• The evaluator installed the application, then checked for the existence of version information. • The evaluator checked for a .swidtag file and opened the file and verified that it contains the SoftwareIdentity element and an Entity element.
Pass/Fail with Explanation	Pass. The evaluator ensured that the application came with a .swidtag file, opened the file and verified that it contains the SoftwareIdentity element and an Entity element.

6.10.5 FPT_LIB_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	The evaluator shall install the application and survey its installation directory for dynamic libraries. The evaluator shall verify that libraries found to be packaged with or employed by the application are limited to those in the assignment.
Test Steps	• The evaluator surveyed the installation directory for dynamic libraries and and found the libraries at C:\Program Files\HelpSystems\GoAnywhere\lib. • The evaluator verified that libraries found to be packaged with or employed by the application are limited to those in the assignment i.e., the third-party libraries listed in section 6.3 of the Security Target.
Pass/Fail with Explanation	Pass. The evaluator Verified that libraries found to be packaged with or employed by the application are limited to those in the assignment. This meets testing requirements.

6.10.6 FPT_TUD_EXT.1.3 Test #1

Item	Data
Test Assurance Activity	The evaluator shall verify that the application's executable files are not changed by the application. The evaluator shall complete the following test: For all other platforms: The evaluator shall install the application and then locate all of its executable files. The evaluator shall then, for each file, save off either a hash of the file or a copy of the file itself. The evaluator shall then run the application and exercise all features of the application as described in the ST. The evaluator shall then compare each executable file with the either the saved hash or the saved copy of the file. The evaluator shall verify that these are identical. TD0548 has been applied.
Test Steps	• The evaluator installed the application and located all of its executable files using HashMyFiles tool. • The evaluator shall then, for each file, saved off the hash of the file using HashMyFiles tool. The entire list of files and their hashes are provided in the form of a .txt file below. • The evaluator then ran the application and exercised all features of the application as described in the ST. • The evaluator then generated hash for each executable using Hashmyfiles tool after running exercising all the features on the TOE. The entire list of files after the TOE was exercised is provided in the form of a .txt file below. • The evaluator then compared each executable file with the saved hash files obtained using HashMyfiles tool and ensured that these are identical.
Pass/Fail with Explanation	Pass. The evaluator verified that all executable files are identical before and after the application is run. This meets testing requirements.

6.10.7 FPT_TUD_EXT.2.2 Test #1

Item	Data
Test Assurance Activity	For All Other Platforms: The evaluator shall record the path of every file on the entire filesystem prior to installation of the application, and then install and run the application. Afterwards, the evaluator shall then uninstall the application, and compare

	the resulting filesystem to the initial record to verify that no files, other than configuration, output, and audit/log files, have been added to the filesystem. Non-applicable platforms removed.
Test Steps	• Before installing the TOE, the evaluator executed the below command: "dir /B /S > before_install.txt" in the root directory C:\ to record the path of every file on the entire filesystem prior to installation of the application and redirected it to a text file. • The evaluator started installing the TOE. • After installation, the Help systems directory in C:\Program Files is as follows. • After installation, the Help systems directory in C:\ProgramData is as follows. • The evaluator ran the application and exercised various features of the TOE. • The evaluator then uninstalled the application by following the steps. • After uninstalling the TOE, the evaluator executed the below command: "dir /B /S > after_uninstall.txt" in the root directory C:\ to get the complete filesystem list and redirect it to a text file. • The evaluator compared both the using software WinMerge. • After uninstalling the TOE, the Help systems folder contains the following subfolders. The gamft.lic is a license file that is manually added by the evaluator after obtaining it from the vendor on request and it is not created as a part of installation process. • The 'tomcat.exe' file that is left in the 'bin' directory is a file that is created at installation time. The 'temp' and 'work' directories are both output directories. The 'conf' and 'logs' are configuration and audit/log files. • The 'index' and 'addonWorkspace' is an output directory while the 'logs' is an audit/log directory. • The evaluator observed that the ProgramData folder did not have any subfolders related to Help systems after uninstalling the application.
Pass/Fail with Explanation	Pass. The evaluator observed that no files other than the configuration, log and output file were added to the system.

6.11 Network (Windows)

6.11.1 FCS_CKM.2.1 – RSA

Item	Data
Test Assurance Activity	The evaluator shall verify the correctness of the TSF's implementation of RSAES-PKCS1-v1_5 by using a known good implementation for each protocol selected in FTP_DIT_EXT.1 that uses RSAES-PKCS1-v1_5.
Expected Results	As per TSS, The TSF performs RSAES-PKCS1-v1_5 key transport with 2048-bit, 3072-bit, and 4096-bit keys in TLS. This test is performed in conjunction with FCS_TLSS_EXT.1.3 Test #1. As per FCS_TLSS_EXT.1.3 Test #1, • The evaluator initiated a TLS connection using the openssl s_client resource to the TOE and ensured that the TLS handshake was successful with RSA key size of 2048 bit.

	• The evaluator observed the packet capture on the client and verified that the connection established with RSA size used (256*8=2048 bit). • The evaluator initiated a TLS connection using the openssl s_client resource to the TOE and ensured that the TLS handshake was successful with RSA key size of 3072 bit. • The evaluator observed the packet capture on the client and verified that the connection established with RSA size used (384*8=3072 bit). • The evaluator initiated a TLS connection using the openssl s_client resource to the TOE and ensured that the TLS handshake was successful with RSA key size of 4096 bit. • The evaluator observed the packet capture on the client and verified that the connection established with RSA size used (512*8=4096 bit). The evaluator confirmed the correctness of the TOE's implementation of RSAES-PKCS1-v1_5 from the above results which indicated that that the client was able to establish a successful TLS connection with 2048-bit, 3072-bit, and 4096-bit keys in TLS.
Pass/Fail with Explanation	Pass. The evaluator verified the correctness of the TSF's implementation of RSAES-PKCS1-v1_5 as a part of FCS_TLSS_EXT.1.3 Test #1.

6.11.2 FCS_CKM.2.1 – DH14

Item	Data
Test Assurance Activity	The evaluator shall verify the correctness of the TSF's implementation of Diffie-Hellman group 14 by using a known good implementation for each protocol selected in FTP_DIT_EXT.1 that uses Diffie-Hellman group 14.
Pass/Fail with Explanation	Pass. The evaluator verified the correctness of the TSF's implementation of Diffie-Hellman group 14 as a part of FCS_SSHS_EXT1.6 Test #1 and FCS_SSHC_EXT1.6 Test #1.

6.11.3 FCS_HTTPS_EXT.1.1/Client Test #1

Item	Data
Test Assurance Activity	The evaluator shall attempt to establish an HTTPS connection with a webserver, observe the traffic with a packet analyzer, and verify that the connection succeeds and that the traffic is identified as TLS or HTTPS. **TD0668 has been applied.**
Test Steps	• Configure the TOE and the HTTPS server for HTTPS POST function. • Create the RootCA and the server certificates using XCA. • Upload RootCA onto the TOE's trust store. • Configure the server to accept SSL. • Establish an HTTPS POST connection from the TOE to a webserver. • Verify that the connection succeeds, and traffic is encrypted with TLS and identified as TLS.
Pass/Fail with Explanation	Pass. The TOE attempted to establish a TLS connection with a web server, observed the traffic with packet analyzer(Wireshark) and verified successful connection. The evaluator also identified the traffic as TLS.

6.11.4 FCS_HTTPS_EXT.1.1/Server Test #1

Item	Data
Test Assurance Activity	The evaluator shall attempt to establish an HTTPS connection to the TOE using a client, observe the traffic with a packet analyzer, and verify that the connection succeeds and that the traffic is identified as TLS or HTTPS. **TD0668 has been applied.**
Test Steps	• The evaluator ensured that the TOE's default https web server runs on port 8001. • Establish an HTTPS connection from a web browser (client) to the TOE. • Verify that the connection succeeds, and traffic is encrypted with TLS and identified as TLS.
Pass/Fail with Explanation	Pass. The evaluator attempted to establish a TLS connection to the TOE using a client (web browser), observed the traffic with a packet analyzer(Wireshark) and verified that the connection succeds. The evaluator also identified the traffic as TLS.

6.11.5 FDP_NET_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	Test 1: The evaluator shall run the application. While the application is running, the evaluator shall sniff network traffic ignoring all non-application associated traffic and verify that any network communications witnessed are documented in the TSS or are user-initiated.
Test Steps	• While the application is running, capture the packets using Wireshark and filter out all non-network related traffic. • The evaluator only observed the following traffic in the above packet capture. • The traffic with server 192.168.254.107 port 3389 is non application related and is the traffic associated between the evaluator's workstation and the VM using Remote Desktop connection (RDP). • Traffic observed throughout the packet capture between the TOE (10.1.3.50) and the MySQL server (10.1.3.51 port 3306) is the User Configured database • Traffic observed throughout the packet capture between the TOE (10.1.3.50 port 636) and the LDAP server (10.1.3.51) is the User Configured Authentication server. • Traffic observed throughout the packet capture between the TOE (10.1.3.50 port 8001) and the HTTPS (10.1.3.51) is the remote HTTPS administration of the TOE and is recorded in the TSS. • Traffic observed throughout the packet capture between the TOE (10.1.3.50) and the SSH server (10.1.3.51 port 22) is the user-initiated connection from the TOE to the TSS. • The NTP protocol packets shown below denotes the communication between the VM hosting the TOE and the Network Time protocol server to synchronize system time and is non application related. • The DNS protocol packets shown below denotes the communication between the VM hosting the TOE and the DNS server used for translating domain names into IP address and is non application related.

	• Traffic observed throughout the packet capture between the TOE (10.1.3.253 port 1214) and the SSH client (10.1.3.51) is the user-initiated connection from the TOE to the SSH server.
Pass/Fail with Explanation	Pass. All the network communication witnessed when the TOE is running are user initiated.

6.11.6 FDP_NET_EXT.1.1 Test #2

Item	Data
Test Assurance Activity	Test 2: The evaluator shall run the application. After the application initializes, the evaluator shall run network port scans to verify that any ports opened by the application have been captured in the ST for the third selection and its assignment. This includes connection-based protocols (e.g. TCP, DCCP) as well as connectionless protocols (e.g. UDP).
Test Steps	• Start the application and wait for the application to initialize. Run network port scan to verify all open ports are listed in the ST. • The evaluator provided rationale regarding the open ports which are not related to the TOE. • The evaluator provided rationale about the ports that are kept open by the TOE. • The evaluator ran the UDP port scan to determine the udp ports that are kept open by the TOE. • The evaluator provided rationale regarding all open ports which are not related to the TOE.
Pass/Fail with Explanation	Pass. The TOE does not open any unexpected ports. This meets the testing requirements.

6.11.7 FTP_DIT_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	Test 1: The evaluator shall exercise the application (attempting to transmit data; for example by connecting to remote systems or websites) while capturing packets from the application. The evaluator shall verify from the packet capture that the traffic is encrypted with HTTPS, TLS, DTLS, SSH, or IPsec in accordance with the selection in the ST. **TD0601 has been applied.** **TD0668 has been applied.**
Test Steps	TOE as HTTPS server: • Note: The TOE's default https web server runs on port 8001 • The evaluator configured the TOE to act as a HTTPS server in System -> Admin Server to serve the web clients who want to administrate the TOE remotely over HTTPS/TLS. • The evaluator attempted to establish an HTTPS connection from a web browser (client) to the TOE and was able to successfully access the TOE. • The evaluator verified that the connection succeeds, and traffic is encrypted with TLS. The evaluator also verified that sensitive data was not sent in plaintext and was sent as encrypted application data.

	TOE as TLS Client to the Database server: • The evaluator imported the RootCA certificate into the keyvault of the TOE that signed the server certificate. • The evaluator loaded the server certificate (server.crt) and the server key (server_key.pem) to the database server. • The evaluator configured the server to leverage the loaded certificate and key along with TLS_RSA_WITH_AES_128_CBC_SHA as the cipher suite. • The evaluator ensured that the configuration was applied to the mysql server. • The evaluator attempted a connection from the TOE to the Data base server and verified the connection to be successful. • The evaluator observed the packet capture and ensured that the traffic is encrypted with TLS. The evaluator also verified that sensitive data was not sent in plaintext and was sent as encrypted application data. TOE as SSH/SFTP server: • The evaluator set the user's authentication type as Password based. • The evaluator attempted to login to the TOE using a valid username/password combination • The evaluator ensured that the connection is successful. • The evaluator observed the packet capture and ensured that the traffic is encrypted with SSH when attempted to exercise the SSH service of the TOE. The evaluator also verified that sensitive data was not sent in plaintext and was sent as encrypted packets. TOE as SSH Client: • The evaluator configured the SSH server to allow aes128-cbc algorithm. • The evaluator attempted a connection from the TOE to the server and verify the connection succeeds. • The evaluator verified through packet capture that the traffic encrypted with SSH when attempted to exercise the SSH client service on the TOE. . The evaluator also verified that sensitive data was not sent in plaintext and was sent as encrypted packets.
Pass/Fail with Explanation	Pass. All the traffic captured when the TOE is exercised is either TLS or SSH.

6.11.8 FTP_DIT_EXT.1.1 Test #2

Item	Data
Test Assurance Activity	Test 2: The evaluator shall exercise the application (attempting to transmit data; for example by connecting to remote systems or websites) while capturing packets from the application. The evaluator shall review the packet capture and verify that no sensitive data is transmitted in the clear. **TD0601 has been applied.** **TD0668 has been applied.**
Expected Results	As per FTP_DIT_EXT.1.1 Test #1 • The evaluator verified using the wireshark capture that the traffic is encrypted with TLS and also ensured that sensitive data was not transmitted in plaintext and was sent as encrypted application data For TLS connections with external resources.

	• The evaluator observed the packet capture and ensured that the traffic is encrypted with SSH connections when attempted to exercise the SSH service of the TOE. The evaluator also verified that sensitive data was not sent in plaintext and was sent as encrypted packets. This test is done in conjunction with FTP_DIT_EXT.1.1 Test #1 where each Wireshark capture evidence was further analyzed to ensure that no sensitive data is transmitted as plain-text and was sent as encrypted application data for TLS connections and encrypted packets for SSH connections.
Pass/Fail with Explanation	Pass. The evaluator reviewed the packet capture for each connection and verified that no sensitive data is transmitted in the clear.

6.11.9 FTP_DIT_EXT.1.1 Test #3

Item	Data
Test Assurance Activity	Test 3: The evaluator shall inspect the TSS to determine if user credentials are transmitted. If credentials are transmitted the evaluator shall set the credential to a known value. The evaluator shall capture packets from the application while causing credentials to be transmitted as described in the TSS. The evaluator shall perform a string search of the captured network packets and verify that the plaintext credential previously set by the evaluator is not found. **TD0601 has been applied.** **TD0668 has been applied.**
Test Steps	TOE as TLS Client to the Database server: • The evaluator imported the RootCA certificate into the keyvault of the TOE that signed the server certificate. • The evaluator loaded the server certificate (server.crt) and the server key (server_key.pem) to the database server. • The evaluator configured the server to leverage the loaded certificate and key along with TLS_RSA_WITH_AES_128_CBC_SHA as the cipher suite. • The evaluator ensured that the configuration was applied to the mysql server. • The evaluator attempted a connection from the TOE to the Data base server using credentials user:WADATA and password:123TesT321 and verified the connection to be successful. • The evaluator observed the packet capture and ensured that the traffic is encrypted with TLS. The evaluator also verified that sensitive data was not sent in plaintext and was sent as encrypted application data. • The credentials used to access the Database server were username: WADATA and password: 123TesT321. The evaluator performed a string search of the captured network packets and verify that the plaintext credentials previously set by the evaluator are not found. TOE as SSH Client: • The evaluator set the SSH server as 10.1.3.51. • The evaluator attempted a connection from the TOE to the SFTP server and verify the connection succeeds.

	• The evaluator verified through packet capture that the traffic encrypted with SSH when attempted to exercise the SSH client service on the TOE. • The credentials used to access the SFTP server were username: acumensec and password: 123TesT321. The evaluator performed a string search of the captured network packets and verify that the plaintext credentials previously set by the evaluator are not found.
Pass/Fail with Explanation	Pass. The evaluator performed a string search of the captured network packets and verify that the plaintext credential previously set by the evaluator is not found. This meets testing requirements.

6.12 Operation (Windows)

6.12.1 FMT_CFG_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	If the application uses any default credentials, the evaluator shall run the following tests. Test 1: The evaluator shall install and run the application without generating or loading new credentials and verify that only the minimal application functionality required to set new credentials is available.
Test Steps	The evaluator observed the TSS which states that the TOE does not come with default credentials. Therefore, this test case is not applicable.
Pass/Fail with Explanation	Pass. the TOE does not come with default credentials.

6.12.2 FMT_CFG_EXT.1.1 Test #2

Item	Data
Test Assurance Activity	If the application uses any default credentials, the evaluator shall run the following tests. Test 2: The evaluator shall attempt to clear all credentials and verify that only the minimal application functionality required to set new credentials is available.
Test Steps	The evaluator observed the TSS which states that the TOE does not come with default credentials. Therefore, this test case is not applicable.
Pass/Fail with Explanation	Pass. the TOE does not come with default credentials.

6.12.3 FMT_CFG_EXT.1.1 Test #3

Item	Data
Test Assurance Activity	If the application uses any default credentials, the evaluator shall run the following tests.

	Test 3: The evaluator shall run the application, establish new credentials, and verify that the original default credentials no longer provide access to the application.
Test Steps	The evaluator observed the TSS which states that the TOE does not come with default credentials. Therefore, this test case is not applicable.
Pass/Fail with Explanation	Pass. the TOE does not come with default credentials.

6.12.4 FMT_SMF.1.1 Test #1

Item	Data
Test Assurance Activity	The evaluator shall test the application's ability to provide the management functions by configuring the application and testing each option selected from above. The evaluator is expected to test these functions in all the ways in which the ST and guidance documentation state the configuration can be managed.
Test Steps	• The Management functions of the TOE are as shown below Configure users • The evaluator created admin users by navigating to Users -> Add Admin User. • The evaluator configured the Admin User roles at Users -> Admin User Roles -> Add Role. • The evaluator set the username, authentication type and Roles and clicked Save. • The admin user 'testadmin' was created and was configured as an Agent manager, Auditor and Log viewer. • The evaluator created web users at Users -> Add Web Users. • The evaluator configured the username of the web user as 'fmtsmf'. • The evaluator configured the protocol section of the web user as SFTP. • The evaluator successfully created the web user named 'fmtsmf' configured for SFTP services. Configure database server • The evaluator configured the database server at System -> DataBase Configuration. Configure authentication server • The evaluator configured the authentication server at System -> Admin Server. • The evaluator ensured that the configuration was successful. Configure mail server • The evaluator configured the mail server at Resources -> SMTP Servers. • The evaluator set the port to 465 and configured the user and connection type. • The evaluator ensured that the configuration was successful. Configure file servers • The evaluator configured the File server at Services -> Service Manager and set the automatically start service to Yes and configured the upload restrictions. • The evaluator configured the File server algorithm parameters. • The evaluator configured the port on which the server runs. • The evaluator configured the Host keys. • The evaluator ensured that the configuration was successful. File transfer services

	• The evaluator created the SSH server resource. • The evaluator ensured that the TOE has the ability to link server resource and create a file transfer project • The evaluator executed the project and ensured that the file transfer was successful. Configure keys and certificates • The evaluator ensured that the TOE has the ability to create and manage Key vaults. • The evaluator ensured that the TOE has the ability to create key vaults. • The evaluator ensured that the TOE has the ability to add certificates. • The evaluator ensured that the TOE has the ability to add key pairs. • The evaluator ensured that the TOE has the ability to add file-based certs and keys. Configure cryptographic protocols • The evaluator ensured that the TOE has the ability to configure cryptographic protocols. • The evaluator ensured that the configuration was successful.
Pass/Fail with Explanation	Pass. The evaluator verified that the TOE can be configured as stated in the ST and Guidance documentation.

6.12.5 FPR_ANO_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	If require user approval before executing is selected, the evaluator shall run the application and exercise the functionality responsibly for transmitting PII and verify that user approval is required before transmission of the PII.
Test Steps	As stated in the ST, TOE does not expressly transmit any PII. Therefore, this test is considered satisfied.
Pass/Fail with Explanation	Pass. As stated in the ST, TOE does not expressly transmit any PII. Therefore, this test is considered satisfied.

6.12.6 FPT_AEX_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	The evaluator shall perform either a static or dynamic analysis to determine that no memory mappings are placed at an explicit and consistent address. The method of doing so varies per platform. For those platforms requiring the same application running on two different systems, the evaluator may alternatively use the same device. After collecting the first instance of mappings, the evaluator must uninstall the application, reboot the device, and reinstall the application to collect the second instance of mappings. For Windows: The evaluator shall run the same application on two different Windows systems and run a tool that will list all memory mapped addresses for the application. The evaluator shall then verify the two different instances share no mapping locations. The Microsoft SysInternals tool, VMMap, could be used to view memory addresses of a running application. The evaluator shall use a tool such as Microsoft's BinScope Binary Analyzer to confirm that the application has ASLR enabled.

	TD0544 has been applied. Non-applicable platforms removed.
Test Steps	• Open VMMap and attach it to the TOE process running on the first TOE Platform. • Open VMMap and attach it to the TOE process running on the second TOE platform. • Use Binscope to confirm that ASLR is enabled.
Pass/Fail with Explanation	Pass. Except for the tomcat.exe, a very small file of the TOE, which has minimal functionality an attacker could exploit, no other memory mapping locations are placed at a consistent and explicit memory location.

6.12.7 FPT_AEX_EXT.1.3 Test #1

Item	Data
Test Assurance Activity	The evaluator shall configure the platform in the ascribed manner and carry out one of the prescribed tests: For Windows: If the OS platform supports Windows Defender Exploit Guard (Windows 10 version 1709 or later), then the evaluator shall ensure that the application can run successfully with Windows Defender Exploit Guard Exploit Protection configured with the following minimum mitigations enabled; Control Flow Guard (CFG), Randomize memory allocations (Bottom-Up ASLR), Export address filtering (EAF), Import address filtering (IAF), and Data Execution Prevention (DEP). The following link describes how to enable Exploit Protection, https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-defender-exploit-guard/customize-exploit-protection. If the OS platform supports the Enhanced Mitigation Experience Toolkit (EMET) which can be installed on Windows 10 version 1703 and earlier, then the evaluator shall ensure that the application can run successfully with EMET configured with the following minimum mitigations enabled; Memory Protection Check, Randomize memory allocations (Bottom-Up ASLR), Export address filtering (EAF), and Data Execution Prevention (DEP). Non-applicable platforms removed. TD0435 has been applied.
Test Steps	• Identify the TOE's executable • Download, install and configure EMET on TOE's underlying platform. https://www.microsoft.com/en-us/download/confirmation.aspx?id=54264 • Start the TOE and verify the TOE successfully runs with EMET configured and the DEP,EAP,ASLR and Memory protection check enabled
Pass/Fail with Explanation	Pass. The TOE runs successfully when EMET is configured and the DEP, EAP, ASLR and Memory protection check enabled.

6.12.8 FPT_TUD_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	The evaluator shall check for an update using procedures described in either the application documentation or the platform documentation and verify that the application does not issue an error. If it is updated or if it reports that no update is available this requirement is considered to be met.

Test Steps	• The evaluator checked for an update at Help -> Check for Updates to query the application for the current version of the software and verified that the application does not issue an error. • The evaluator verified that no update is available, and the TOE is running on the latest version 6.8.3 which also matches the documented and installed version.
Pass/Fail with Explanation	Pass. The TOE has no update available, and the latest version of application is being used. This meets testing requirements.

6.12.9 FPT_TUD_EXT.1.2 Test #1

Item	Data
Test Assurance Activity	The evaluator shall query the application for the current version of the software according to the operational user guidance. The evaluator shall then verify that the current version matches that of the documented and installed version.
Test Steps	• The evaluator queried the application for the current version of the software according to the operational user guidance at Help -> About. • The evaluator then verified that the current version 6.8.3 matches that of the documented and installed version.
Pass/Fail with Explanation	Pass. The evaluator queried the application for the current version of the software according to the operational user guidance and verified that the current version matches that of the documented and installed version.

6.13 PKG_TLSC (Windows)

6.13.1 FCS_TLSC_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	The evaluator shall establish a TLS connection using each of the cipher suites specified by the requirement. This connection may be established as part of the establishment of a higher-level protocol, e.g., as part of an EAP session. It is sufficient to observe the successful negotiation of a cipher suite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic in an attempt to discern the cipher suite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).
Test Steps	• The evaluator created a server certificate that was signed by the Root CA and the server key using the XCA tool. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator loaded the server certificate (server.crt) and the server key (server_key.pem) to the database server. • The evaluator configured the server to leverage the loaded certificate and key along with each of the cipher suite supported by the TOE. ▪ TLS_RSA_WITH_AES_128_CBC_SHA ▪ TLS_RSA_WITH_AES_256_CBC_SHA ▪ TLS_RSA_WITH_AES_128_CBC_SHA256 ▪ TLS_RSA_WITH_AES_256_CBC_SHA256

	▪ TLS_RSA_WITH_AES_128_GCM_SHA256 ▪ TLS_RSA_WITH_AES_256_GCM_SHA384 ▪ TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 ▪ TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 • The evaluator attempted a connection from the TOE to the Data base server and verified the connection to be successful. • The evaluator observed the packet capture and ensured that the TLS connection was successful using the configured cipher suite. • The evaluator created a server ec certificate that was signed by the Root CA_ec and the server key using the XCA tool. • The evaluator imported the RootCA_ec certificate into the key vault of the TOE that signed the ec server certificate. • The evaluator loaded the server certificate (server_ec.crt) and the server key (server_ec_key.pem) to the database server. • The evaluator configured the server to leverage the loaded certificate and key along with along with each of the cipher suite supported by the TOE. ▪ TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 ▪ TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 ▪ TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 ▪ TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 • The evaluator attempted a connection from the TOE to the Data base server and verified the connection to be successful. • The evaluator observed the packet capture and ensured that the TLS connection was successful using the configured cipher suite.
Pass/Fail with Explanation	Pass. The evaluator established a TLS connection using each of the cipher suites with the Data Base server and observed the successful negotiation using each of the claiming cipher suite specified. This meets the testing requirements.

6.13.2 FCS_TLSC_EXT.1.1 Test #2

Item	Data
Test Assurance Activity	The goal of the following test is to verify that the TOE accepts only certificates with appropriate values in the extendedKeyUsage extension, and implicitly that the TOE correctly parses the extendedKeyUsage extension as part of X.509v3 server certificate validation. The evaluator shall attempt to establish the connection using a server with a server certificate that contains the Server Authentication purpose in the extendedKeyUsage extension and verify that a connection is established. The evaluator shall repeat this test using a different, but otherwise valid and trusted, certificate that lacks the Server Authentication purpose in the extendedKeyUsage extension and ensure that a connection is not established. Ideally, the two certificates should be similar in structure, the types of identifiers used, and the chain of trust.

Test Steps	• The evaluator imported the RootCA certificate into the keyvault of the TOE that signed the server certificate. • The evaluator ensured the server certificate that is being used for a TLS connection contain Server Authentication purpose in the extendedKeyUsage extension. • The evaluator loaded the server certificate (server.crt) and the server key (server_key.pem) to the database server and configured the server to leverage the loaded certificate and key to establish a TLS connection with the TOE. • The evaluator attempted a connection from the TOE to the Data base server and verified the connection to be successful. • The evaluator observed the packet capture and ensured that the connection using a server with a server certificate that contains the Server Authentication purpose in the extendedKeyUsage extension established successfully as seen in packet 2. • The evaluator created an identical certificate (server_nsa.crt) that is similar in structure, the types of identifiers used, and the chain of trust but lacks the Server Authentication purpose in the extendedKeyUsage extension. • The evaluator loaded the server certificate (server_nsa.crt) and the server key (server_key.pem) to the database server and configured the server to leverage the loaded certificate and key to establish a TLS connection with the TOE. • The evaluator attempted a connection from the TOE to the Data base server and verified that the connection did not establish. • The evaluator observed the packet capture and ensured that the connection using a server with a server certificate that lacks the Server Authentication purpose in the extendedKeyUsage extension did not establish. The packet 2 in the screen capture below shows the missing Extended key usage field in extensions. • The evaluator further verified the debug logs on the TOE to ensure that the connection did not establish due to invalid server extended key usage.
Pass/Fail with Explanation	Pass. The TOE established the connection using a server with a server certificate that contains the Server Authentication purpose in the extendedKeyUsage extension and did not establish a connection with a server certificate that lacks the Server Authentication purpose in the extendedKeyUsage extension. This meets the testing requirements.

6.13.3 FCS_TLSC_EXT.1.1 Test #3

Item	Data
Test Assurance Activity	The evaluator shall send a server certificate in the TLS connection that does not match the server-selected cipher suite (for example, send a ECDSA certificate while using the TLS_RSA_WITH_AES_128_CBC_SHA cipher suite or send a RSA certificate while using one of the ECDSA cipher suites.) The evaluator shall verify that the product disconnects after receiving the server's Certificate handshake message.
Test Steps	• The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator attempted a connection to the server and observed the output on the TOE that indicated that the TOE could not to the server.

	• The evaluator used the "acumen-tlsc-mysql" tool to send an RSA server certificate in the TLS connection while using the cipher suite TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 when the TOE attempted to connect to the server. The evaluator observed the tool output which indicated that the TLS connection did not establish due to FATAL alert returned by the server. • The evaluator observed the packet capture to ensure that the server sent an RSA server certificate (as seen in packet 3) in the TLS connection while using the cipher suite TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 (as seen in packet 2) when the TOE attempted to connect to the server. • The evaluator observed that the TOE disconnected after receiving the server's Certificate handshake message.
Pass/Fail with Explanation	Pass. The TOE disconnected with the remote server after receiving the server's Certificate handshake message as the server was using a cipher suite that did not match the certificate. This meets the testing requirements.

6.13.4 FCS_TLSC_EXT.1.1 Test #4

Item	Data
Test Assurance Activity	The evaluator shall configure the server to select the TLS_NULL_WITH_NULL_NULL cipher suite and verify that the client denies the connection.
Test Steps	• The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator attempted a connection to the server and observed the output on the TOE that indicated that the TOE could not to the server. • The evaluator used the "acumen-tlsc-mysql" tool to attempt a connection by the remote TLS server using the TLS_NULL_WITH_NULL_NULL cipher suite. The evaluator observed the tool output which indicated that the TLS connection did not establish due to FATAL alert returned by the server. • The evaluator observed the packet capture to ensure that the server attempted a connection with TLS_NULL_WITH_NULL_NULL cipher suite (as seen in packet 2) and verified that the client denies the connection (as seen in packet 3).
Pass/Fail with Explanation	Pass. The TOE denies a connection to a server using the TLS_NULL_WITH_NULL_NULL cipher suite. This meets the testing requirements.

6.13.5 FCS_TLSC_EXT.1.1 Test #5.1

Item	Data
Test Assurance Activity	Change the TLS version selected by the server in the Server Hello to an undefined TLS version (for example 1.5 represented by the two bytes 03 06) and verify that the client rejects the connection.
Test Steps	• The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator attempted a connection to the server and observed the output on the TOE that indicated that the TOE could not to the server.

	• The evaluator used the "acumen-tlsc-mysql" tool to attempt a connection by a remote TLS server using an undefined TLS version (0x0001) and observed the tool output which indicated that the TLS version was set to 0x0001. • The evaluator observed the packet capture and ensured that the TOE rejected the connection due to protocol version as the TLS version selected by the server in the Server Hello was set to an undefined TLS version (0x0001).
Pass/Fail with Explanation	Pass. The TOE rejected the connection when the server selects an undefined version of TLS and verified that the TOE rejected the connection. This meets the testing requirements.

6.13.6 FCS_TLSC_EXT.1.1 Test #5.2

Item	Data
Test Assurance Activity	Change the TLS version selected by the server in the Server Hello to the most recent unsupported TLS version (for example 1.1 represented by the two bytes 03 02) and verify that the client rejects the connection.
Test Steps	• The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator attempted a connection to the server and observed the output on the TOE that indicated that the TOE could not to the server. • The evaluator used the "acumen-tlsc-mysql" tool to send a Server Hello with an unsupported TLS version (TLS v1.1) in the TLS connection when the TOE attempted to connect to the server. The evaluator observed the tool output which indicated that the TLS connection did not establish due to FATAL alert: PROTOCOL_VERSION returned by the TOE. • The evaluator observed the packet capture to ensure that the server sent a Server Hello with an unsupported TLS version (TLS v1.1) in the TLS connection when the TOE attempted to connect to the server and verified that the TLS connection did not establish due to FATAL alert: PROTOCOL_VERSION returned by the server.
Pass/Fail with Explanation	Pass. The TOE rejected the connection when the server selects a recently unsupported version of TLS. This meets the testing requirements.

6.13.7 FCS_TLSC_EXT.1.1 Test #5.3

Item	Data
Test Assurance Activity	[conditional] If **DHE or ECDHE cipher suites are supported**, modify at least one byte in the server's nonce in the Server Hello handshake message, and verify that the client does not complete the handshake and no application data flows.
Test Steps	• The evaluator imported the RootCA_ec certificate into the key vault of the TOE that signed the server certificate. • The evaluator attempted a connection to the server and observed the output on the TOE that indicated that the TOE could not to the server. • The evaluator used the "acumen-tlsc-mysql" tool to attempt a connection with server that modifies the server's nonce in the server hello message. The evaluator observed the tool output which indicated that the TLS connection did not establish due to WARNING alert: CLOSE_NOTIFY returned by the TOE. Note:

	Certificate modification is done by acumen-tlsc-mysql tool as shown below to meet the test requirements. server.crt is the server certificate and the server_key.pem is the server private key. • The evaluator observed the packet capture and ensured that the client does not complete the handshake and no application data flows.
Pass/Fail with Explanation	Pass. The evaluator ensured that the client does not complete the TLS handshake with a remote server due to an invalid server nonce in the Server Hello message and no application data flows.

6.13.8 FCS_TLSC_EXT.1.1 Test #5.4

Item	Data
Test Assurance Activity	Modify the server's selected cipher suite in the Server Hello handshake message to be a cipher suite not presented in the Client Hello handshake message. The evaluator shall verify that the client does not complete the handshake and no application data flows.
Test Steps	• The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator attempted a connection to the server and observed the output on the TOE which indicated that the TOE could not to the server due to "illegal_paramater". • The evaluator used the "acumen-tlsc-mysql" tool to attempt a connection with server that modifies the server's selected cipher suite in the Server Hello handshake message to be a cipher suite not presented in the Client Hello handshake message. The evaluator observed the tool output which indicated that the TLS connection did not establish due to TLS error: ILLEGAL_PARAMETER returned by the TOE. Certificate modification is done by acumen-tlsc-mysql tool as shown below to meet the test requirements. server.crt is the server certificate and the server_key.pem is the server private key. • The evaluator observed the packet capture and ensured that the client does not complete the handshake due to "Illegal parameter" and no application data flows.
Pass/Fail with Explanation	Pass. The evaluator verified that the client does not complete the handshake and no application data flows when an unsupported cipher suite is presented in the Server Hello handshake message .

6.13.9 FCS_TLSC_EXT.1.1 Test #5.5

Item	Data
Test Assurance Activity	[conditional] If **DHE or ECDHE cipher suites are supported**, modify the signature block in the server's Key Exchange handshake message, and verify that the client does not complete the handshake and no application data flows. This test does not apply to cipher suites using RSA key exchange. If a TOE only supports RSA key exchange in conjunction with TLS, then this test shall be omitted.
Test Steps	• The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator used the "acumen-tlsc-mysql" tool to attempt a connection with server that modifies the signature block in the server's Key Exchange

	handshake message. The evaluator observed the tool output which indicated that the TLS connection did not establish as the client did not complete the handshake due to TLS error: DECRYPT_ERROR returned by the TOE. Note: Certificate modification is done by acumen-tlsc-mysql tool as shown below to meet the test requirements. server.crt is the server certificate and the server_key.pem is the server private key. • The evaluator attempted a connection to the server and observed the output on the TOE that indicated that the TOE could not connect to the server due to "decrypt_error". • The evaluator observed the packet capture and ensured that the client does not complete the handshake due to "Decrypt error" and no application data flows.
Pass/Fail with Explanation	Pass. The evaluator verified that the client does not complete the handshake and no application data flows when a modified signature is presented by the server in the server's Key Exchange handshake message.

6.13.10 FCS_TLSC_EXT.1.1 Test #5.6

Item	Data
Test Assurance Activity	Modify a byte in the Server Finished handshake message and verify that the handshake is not finished successfully, and no application data flows.
Test Steps	• The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator used the "acumen-tlsc-mysql" tool to attempt a connection with server that modifies a byte in the Server Finished handshake message. The evaluator observed the tool output which indicated that the TLS connection did not establish as the client did not complete the handshake due to TLS error: DECRYPT_ERROR returned by the TOE. Note: Certificate modification is done by acumen-tlsc-mysql tool as shown below to meet the test requirements. server.crt is the server certificate and the server_key.pem is the server private key. • The evaluator attempted a connection to the server and observed the output on the TOE that indicated that the TOE could not connect to the server due to "decrypt_error". • The evaluator observed the packet capture and ensured that the client does not complete the handshake and no application data flows.
Pass/Fail with Explanation	Pass. The evaluator verified that the client does not complete the handshake and no application data flows when a modified server's finished handshake message is sent.

6.13.11 FCS_TLSC_EXT.1.1 Test #5.7

Item	Data
Test Assurance Activity	Send a message consisting of random bytes from the server after the server has issued the Change Cipher Spec message and verify that the client does not complete the handshake and no application data flows. The message must still have a valid 5-byte record header in order to ensure the message will be parsed as TLS.

Test Steps	• The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator used the "acumen-tlsc-mysql" tool to attempt a connection with server that sends a message consisting of random bytes from the server after the server has issued the Change Cipher Spec message. The evaluator observed the tool output which indicated that the TLS connection did not establish as the client did not complete the handshake due to TLS error: UNEXPECTED_MESSAGE returned by the TOE. Note: Certificate modification is done by acumen-tlsc-mysql tool as shown below to meet the test requirements. server.crt is the server certificate and the server_key.pem is the server private key. • The evaluator attempted a connection to the server and observed the output on the TOE that indicated that the TOE could not connect to the server due to "unexpected_message". • The evaluator observed the packet capture and ensured that the client does not complete the handshake due to "encrypted alert". The application data shown in the below pcap is the garbled message that is sent before sending the finished message and must not be mistaken for application data that indicates a successful TLS connection. The evaluator ensured that the message still have a valid 5-byte record header in order to ensure the message will be parsed as TLS.
Pass/Fail with Explanation	Pass. The evaluator verified that the client did not complete the handshake and no application data flows when a message consisting of random bytes is sent from the server after the server has issued the Change Cipher Spec message.

6.13.12 FCS_TLSC_EXT.1.2 Test #1

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. Test 1: The evaluator shall present a server certificate that contains a CN that does not match the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection fails. Note that some systems might require the presence of the SAN extension. In this case the connection would still fail but for the reason of the missing SAN extension instead of the mismatch of CN and reference identifier. Both reasons are acceptable to pass Test 1. **TD0499 has been applied.**
Test Steps	• The evaluator ensured that the Hostname Verification is enabled on the TOE in Admin > Security Settings. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator created a server certificate that contains a CN (11.1.3.51) that does not match the reference identifier (10.1.3.51) and does not contain the SAN extension using XCA tool.

	• The evaluator uploaded the server key and the certificate on the server and ensured it to contain a CN (11.1.3.51) that does not match the reference identifier (10.1.3.51) and does not contain the SAN extension. • The evaluator configured the database server to leverage the uploaded server certificate and the key for TLS handshake. • The TOE automatically configures references identifiers based on the FQDN or IP address configured used to specify the TLS server. The evaluator attempted a connection to the server and ensured that the TOE could not connect to the server. • The evaluator observed the packet capture and ensured that the client does not complete the TLS handshake due to "certificate unknown" message • The evaluator further observed the logs on the TOE located at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs to confirm that the TOE could not connect to the server as there was no subject alternative name found matching IP address 10.1.3.51.
Pass/Fail with Explanation	Pass. The evaluator verified that the TLS connection failed when the server presented certificate that contains a CN that does not match the reference identifier and does not contain the SAN extension.

6.13.13 FCS_TLSC_EXT.1.2 Test #2

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. Test 2: The evaluator shall present a server certificate that contains a CN that matches the reference identifier, contains the SAN extension, but does not contain an identifier in the SAN that matches the reference identifier. The evaluator shall verify that the connection fails. The evaluator shall repeat this test for each supported SAN type. **TD0499 has been applied.**
Test Steps	**Note: The following test performed satisifies the FCS_TLSC_EXT.1.3 Test #4 requirement where the evaluator demonstrated that a server using a certificate which does not have a valid identifier (in the SAN) results in an authentication failure which was confirmed through wireshark packet capture where the client could not connect to the server as the server returned a certificate_unknown error to the client and also confirmed through TOE logs that the TOE could not connect to the server as there was no subject alternative name found matching IP address 10.1.3.51.** • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator created a server certificate that contains a CN (10.1.3.51) that matches the reference identifier (10.1.3.51) contains the SAN extension but does not contain an identifier in the SAN (11.1.3.51) that matches the reference identifier (10.1.3.51). • The evaluator uploaded the server key and the certificate on the server certificate that contains a CN (10.1.3.51) that does match the reference

	identifier (10.1.3.51) contains the SAN extension but does not contain an identifier in the SAN (11.1.3.51) that matches the reference identifier (10.1.3.51). • The evaluator configured the database server to leverage the uploaded server certificate and the key for TLS handshake. • The TOE automatically configures references identifiers based on the FQDN or IP address configured used to specify the TLS server. The evaluator attempted a connection to the server and ensured that the TOE could not connect to the server. • The evaluator observed the packet capture and ensured that the client does not complete the TLS handshake due to "certificate unknown" message. • The evaluator further observed the logs on the TOE located at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs to confirm that the TOE could not connect to the server as there was no subject alternative name found matching IP address 10.1.3.51. DNS: • The evaluator created a server certificate that contains a CN (sqlserver.acumensec.local) that matches the reference identifier (sqlserver.acumensec.local) contains the SAN extension but does not contain an identifier in the SAN (wrong.acumensec.local) that matches the reference identifier (sqlserver.acumensec.local). • The evaluator uploaded the certificate on the server and ensured it contains a CN (sqlserver.acumensec.local) that matches the reference identifier (sqlserver.acumensec.local) contains the SAN extension but does not contain an identifier in the SAN (wrong.acumensec.local) that matches the reference identifier (sqlserver.acumensec.local). • The evaluator configured the database server to leverage the uploaded server certificate and the key for TLS handshake. • The TOE automatically configures references identifiers based on the FQDN or IP address configured used to specify the TLS server. The evaluator attempted a connection to the server and ensured that the TOE could not connect to the server. • The evaluator observed the packet capture and ensured that the client does not complete the TLS handshake due to "certificate unknown" message. • The evaluator further observed the logs on the TOE located at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs to confirm that the TOE could not connect to the server as there was no subject alternative name found matching IP address 10.1.3.51.
Pass/Fail with Explanation	Pass. The evaluator verified with each supported identifier that the TLS connection fails when the server presented a certificate that contains a CN that matches the reference identifier, contains the SAN extension, but does not contain an identifier in the SAN that matches the reference identifier.

6.13.14 FCS_TLSC_EXT.1.2 Test #3

Item	Data

Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. Test 3: [conditional] If **the TOE does not mandate the presence of the SAN extension**, the evaluator shall present a server certificate that contains a CN that matches the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection succeeds. If the TOE does mandate the presence of the SAN extension, this Test shall be omitted. **TD0499 has been applied.**
Test Steps	Note: As mentioned in the TSS, when an IP address is configured, the TOE mandates the presence of a SAN. Hence this test is N/A when the reference identifiers are IP addresses. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator created a server certificate that contains a CN (sqlserver.acumensec.local) that matches the reference identifier (sqlserver.acumensec.local) and does not contain the SAN extension using the XCA tool. • The evaluator uploaded the server key and the certificate on the server and ensured it to contain a CN that matches the reference identifier(sqlserver.acumensec.local) and does not contain the SAN extension. • The evaluator configured the database server to leverage the uploaded server certificate and the key for TLS handshake. • The TOE automatically configures references identifiers based on the FQDN or IP address configured used to specify the TLS server. The evaluator attempted a connection to the server and ensured that the connection was successful. • The evaluator observed the packet capture and ensured that the TLS connection was successful.
Pass/Fail with Explanation	Pass. The evaluator confirmed that as per TSS, the TOE mandates the presence of SAN extension for IP address as reference identifier and omitted the test. The evaluator verified with FQDN as identifier that the TLS connection succeeds when the server presented a certificate that contains a CN that matches the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection succeeds.

6.13.15 FCS_TLSC_EXT.1.2 Test #4

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7.

	Test 4: The evaluator shall present a server certificate that contains a CN that does not match the reference identifier but does contain an identifier in the SAN that matches. The evaluator shall verify that the connection succeeds. **TD0499 has been applied.**
Test Steps	• The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator created a server certificate that contains a CN (11.1.3.51) that does not match the reference identifier (10.1.3.51) but does contain an identifier in the SAN (10.1.3.51) that matches using XCA tool. • The evaluator uploaded the server key and the certificate on the server and ensured it contains a CN (11.1.3.51) that does not match the reference identifier (10.1.3.51) but does contain an identifier in the SAN (10.1.3.51) that matches. • The evaluator configured the database server to leverage the uploaded server certificate and the key for TLS handshake. • The TOE automatically configures references identifiers based on the FQDN or IP address configured used to specify the TLS server. The evaluator attempted a connection to the server and ensured that the connection was successful. • The evaluator observed the packet capture and ensured that the TLS connection was successful.
Pass/Fail with Explanation	Pass. The evaluator verified that the TLS connection succeeds when the server presented a certificate that contains a CN that does not match the reference identifier but does contain an identifier in the SAN that matches.

6.13.16 FCS_TLSC_EXT.1.2 Test #5.1

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. The evaluator shall perform the following wildcard tests with each supported type of reference identifier. **Test 5.1:** [conditional]: If **wildcards are supported**, the evaluator shall present a server certificate containing a wildcard that is not in the left-most label of the presented identifier (e.g. foo.*.example.com) and verify that the connection fails. **TD0499 has been applied.**
Test Steps	CN: • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator created a server certificate containing a wildcard that is not in the left-most label of the presented identifier (sqlserver.*.acumensec.local). • The evaluator uploaded the server key and the certificate on the server and ensured it contains a wildcard that is not in the left-most label of the presented identifier (sqlserver.*.acumensec.local) in the CN.

	• The evaluator configured the database server to leverage the uploaded server certificate and the key for TLS handshake. • The TOE automatically configures references identifiers based on the FQDN or IP address configured used to specify the TLS server. The evaluator attempted a connection to the server and ensured that the TOE could not connect to the server. • The evaluator observed the packet capture and ensured that the client does not complete the TLS handshake due to “certificate unknown” message. • The evaluator further observed the logs on the TOE located at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs to confirm that the TOE could not connect to the server as the Common Name 'sqlserver.*.acumensec.local' is not allowed according to the strict hostname verification policy. SAN: • The evaluator created a server certificate containing a wildcard that is not in the left-most label of the presented identifier (sqlserver.*.acumensec.local). • The evaluator uploaded the server key and the certificate on the server and ensured it contains a wildcard that is not in the left-most label of the presented identifier (sqlserver.*.acumensec.local) in the SAN field. • The evaluator configured the database server to leverage the uploaded server certificate and the key for TLS handshake. • The TOE automatically configures references identifiers based on the FQDN or IP address configured used to specify the TLS server. The evaluator attempted a connection to the server and ensured that the TOE could not connect to the server. • The evaluator observed the packet capture and ensured that the client does not complete the TLS handshake due to “certificate unknown” message. • The evaluator further observed the logs on the TOE located at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs to confirm that the TOE could not connect to the server as the Subject Alternative Name 'sqlserver.*.acumensec.local' is not allowed according to the strict hostname verification policy.
Pass/Fail with Explanation	Pass. The evaluator verified that the TLS connection failed when the server presented a certificate containing a wildcard that is not in the left-most label of the presented identifier (sqlserver.*.acumensec.local).

6.13.17 FCS_TLSC_EXT.1.2 Test #5.2(a)

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. The evaluator shall perform the following wildcard tests with each supported type of reference identifier.

	Test 5.2: [conditional]: If **wildcards are supported**, the evaluator shall present a server certificate containing a wildcard in the left-most label but not preceding the public suffix (e.g. *.example.com). - The evaluator shall configure the reference identifier with a single left-most label (e.g. foo.example.com) and verify that the connection succeeds. **TD0499 has been applied.**
Test Steps	SAN: • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator created a server certificate that contains a wildcard in the left-most label but not preceding the public suffix (*.acumensec.local). • The evaluator uploaded the server key and the certificate on the server and ensured it contains a wildcard in the left-most label but not preceding the public suffix (*.acumensec.local) in the SAN field. • The evaluator configured the database server to leverage the uploaded server certificate and the key for TLS handshake. • The TOE automatically configures references identifiers based on the FQDN or IP address configured used to specify the TLS server. The evaluator attempted a connection to the server with identifier "sqlserver.acumensec.local" resolved to 10.1.3.51 and ensured that the connection was successful. • The evaluator observed the packet capture and ensured that the TLS handshake was successful. CN: • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator created a server certificate that contains a wildcard in the left-most label but not preceding the public suffix (*.acumensec.local). • The evaluator uploaded the server key and the certificate on the server and ensured it contains a wildcard in the left-most label but not preceding the public suffix (*.acumensec.local) in the CN field. • The evaluator configured the database server to leverage the uploaded server certificate and the key for TLS handshake. • The TOE automatically configures references identifiers based on the FQDN or IP address configured used to specify the TLS server. The evaluator attempted a connection to the server with identifier "sqlserver.acumensec.local" resolved to 10.1.3.51 and ensured that the connection was successful. • The evaluator observed the packet capture and ensured that the TLS handshake was successful.
Pass/Fail with Explanation	Pass. The evaluator verified that the TLS connection succeeds when the server sends a certificate with wildcard in the left-most label of the presented identifier.

6.13.18 FCS_TLSC_EXT.1.2 Test #5.2(b)

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection.

	If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. The evaluator shall perform the following wildcard tests with each supported type of reference identifier. Test 5.2: [conditional]: If **wildcards are supported**, the evaluator shall present a server certificate containing a wildcard in the left-most label but not preceding the public suffix (e.g. *.example.com). - The evaluator shall configure the reference identifier without a left-most label as in the certificate (e.g. example.com) and verify that the connection fails. **TD0499 has been applied.**
Test Steps	SAN: • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator created a server certificate that contains a wildcard in the left-most label but not preceding the public suffix (*.acumensec.local). • The evaluator uploaded the server key and the certificate on the server and ensured it to contain a server certificate that contains a wildcard in the left-most label but not preceding the public suffix (*.acumensec.local) in the SAN field. • The evaluator configured the database server to leverage the uploaded server certificate and the key for TLS handshake. • The TOE automatically configures references identifiers based on the FQDN or IP address configured used to specify the TLS server. The evaluator attempted a connection to the server with identifier “acumensec.local” resolved to 10.1.3.51 and ensured that the TOE could not connect to the server. • The evaluator observed the packet capture and ensured that the client does not complete the TLS handshake due to “certificate unknown” message. • The evaluator further observed the logs on the TOE located at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs to confirm that the TOE could not connect to the server as there was no subject alternative name found matching IP address 10.1.3.51. CN: • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator created a server certificate that contains a wildcard in the left-most label but not preceding the public suffix (*.acumensec.local). • The evaluator uploaded the server key and the certificate on the server and ensured it to contain a server certificate that contains a wildcard in the left-most label but not preceding the public suffix (*.acumensec.local) in the CN field. • The evaluator configured the database server to leverage the uploaded server certificate and the key for TLS handshake. • The TOE automatically configures references identifiers based on the FQDN or IP address configured used to specify the TLS server. The evaluator attempted a connection to the server with identifier “acumensec.local” resolved to 10.1.3.51 and ensured that the TOE could not connect to the server.

	• The evaluator observed the packet capture and ensured that the client does not complete the TLS handshake due to "certificate unknown" message. • The evaluator further observed the logs on the TOE located at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs to confirm that the TOE could not connect to the server as there was no subject alternative name found matching IP address 10.1.3.51.
Pass/Fail with Explanation	Pass. The evaluator verified that the TLS connection failed when the server presented a certificate containing a wildcard in the left-most label but not preceding the public suffix while the reference identifier did not contain a left most label.

6.13.19 FCS_TLSC_EXT.1.2 Test #5.2(c)

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. The evaluator shall perform the following wildcard tests with each supported type of reference identifier. Test 5.2: [conditional]: If **wildcards are supported**, the evaluator shall present a server certificate containing a wildcard in the left-most label but not preceding the public suffix (e.g. *.example.com). - The evaluator shall configure the reference identifier with two left-most labels (e.g. bar.foo.example.com) and verify that the connection fails. **TD0499 has been applied.**
Test Steps	SAN: • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator created a server certificate that contains a wildcard in the left-most label but not preceding the public suffix (*.acumensec.local). • The evaluator uploaded the server key and the certificate on the server and ensured it to contain a server certificate that contains a wildcard in the left-most label but not preceding the public suffix (*.acumensec.local) in the SAN field. • The evaluator configured the database server to leverage the uploaded server certificate and the key for TLS handshake. • The TOE automatically configures references identifiers based on the FQDN or IP address configured used to specify the TLS server. The evaluator attempted a connection to the server with identifier "random.sqlserver.acumensec.local" resolved to 10.1.3.51 and ensured that the TOE could not connect to the server. • The evaluator observed the packet capture and ensured that the client does not complete the TLS handshake due to "certificate unknown" message. • The evaluator further observed the logs on the TOE located at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs to confirm that the TOE could not connect to the server as there was no subject alternative name found matching IP address 10.1.3.51.

	CN: • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator created a server certificate that contains a wildcard in the left-most label but not preceding the public suffix (*.acumensec.local). • The evaluator uploaded the server key and the certificate on the server and ensured it to contain a server certificate that contains a wildcard in the left-most label but not preceding the public suffix (*.acumensec.local) in the CN field. • The evaluator configured the database server to leverage the uploaded server certificate and the key for TLS handshake. • The TOE automatically configures references identifiers based on the FQDN or IP address configured used to specify the TLS server. The evaluator attempted a connection to the server with identifier "random.sqlserver.acumensec.local" resolved to 10.1.3.51 and ensured that the TOE could not connect to the server. • The evaluator observed the packet capture and ensured that the client does not complete the TLS handshake due to "certificate unknown" message. • The evaluator further observed the logs on the TOE located at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs to confirm that the TOE could not connect to the server as there was no subject alternative name found matching IP address 10.1.3.51.
Pass/Fail with Explanation	Pass. The evaluator verified that the TLS connection failed when the reference identifier on the client was configured with two left-most labels while the server presented a certificate containing a wildcard in the left-most label but not preceding the public suffix.

6.13.20 FCS_TLSC_EXT.1.2 Test #5.3(a)

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. The evaluator shall perform the following wildcard tests with each supported type of reference identifier. Test 5.3: [conditional]: If **wildcards are supported**, the evaluator shall present a server certificate containing a wildcard in the left-most label immediately preceding the public suffix (e.g. *.com). - The evaluator shall configure the reference identifier with a single left-most label (e.g. foo.com) and verify that the connection fails. **TD0499 has been applied.**
Test Steps	SAN: • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator created a server certificate that contains a wildcard in the left-most label immediately preceding the public suffix (*.local).

	• The evaluator uploaded the server key and the certificate on the server and ensured it contains a wildcard in the left-most label immediately preceding the public suffix (*.local) in the SAN field. • The evaluator configured the database server to leverage the uploaded server certificate and the key for TLS handshake. • The TOE automatically configures references identifiers based on the FQDN or IP address configured used to specify the TLS server. The evaluator attempted a connection to the server with identifier “acumensec.local” resolved to 10.1.3.51 and ensured that the TOE could not connect to the server. • The evaluator observed the packet capture and ensured that the client does not complete the TLS handshake due to “certificate unknown” message. • The evaluator further observed the logs on the TOE located at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs to confirm that the TOE could not connect to the server as the Subject Alternative Name '*.local' is not allowed according to the strict hostname verification policy. CN: • The evaluator created a server certificate that contains a wildcard in the left-most label immediately preceding the public suffix (*.local). • The evaluator uploaded the server key and the certificate on the server and ensured it contains a wildcard in the left-most label immediately preceding the public suffix (*.local) in the CN field. • The evaluator configured the database server to leverage the uploaded server certificate and the key for TLS handshake. • The TOE automatically configures references identifiers based on the FQDN or IP address configured used to specify the TLS server. The evaluator attempted a connection to the server with identifier “acumensec.local” resolved to 10.1.3.51 and ensured that the TOE could not connect to the server. • The evaluator observed the packet capture and ensured that the client does not complete the TLS handshake due to “certificate unknown” message. • The evaluator further observed the logs on the TOE located at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs to confirm that the TOE could not connect to the server as the Common Name '*.local' is not allowed according to the strict hostname verification policy.
Pass/Fail with Explanation	Pass. The evaluator verified that the TLS connection fails when the server presented a certificate with wildcard in the left-most label of the presented identifier while the reference identifier configured on the TOE does not contain a left most label.

6.13.21 FCS_TLSC_EXT.1.2 Test #5.3(b)

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. The evaluator shall perform the following wildcard tests with each supported type of reference identifier.

	Test 5.3: [conditional]: If **wildcards are supported**, the evaluator shall present a server certificate containing a wildcard in the left-most label immediately preceding the public suffix (e.g. *.com). - The evaluator shall configure the reference identifier with two left-most labels (e.g. bar.foo.com) and verify that the connection fails. **TD0499 has been applied.**
Test Steps	SAN: • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator created a server certificate that contains a wildcard in the left-most label immediately preceding the public suffix (*.local). • The evaluator uploaded the server key and the certificate on the server and ensured it contains a wildcard in the left-most label immediately preceding the public suffix (*.local) in the SAN field. • The evaluator configured the database server to leverage the uploaded server certificate and the key for TLS handshake. • The TOE automatically configures references identifiers based on the FQDN or IP address configured used to specify the TLS server. The evaluator attempted a connection to the server with identifier “sqlserver.acumensec.local” resolved to 10.1.3.51 and ensured that the TOE could not connect to the server. • The evaluator observed the packet capture and ensured that the client does not complete the TLS handshake due to “certificate unknown” message. • The evaluator further observed the logs on the TOE located at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs to confirm that the TOE could not connect to the server as The Subject Alternative Name '*.local' is not allowed according to the strict hostname verification policy. CN: • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator created a server certificate that contains a wildcard in the left-most label immediately preceding the public suffix (*.local). • The evaluator uploaded the server key and the certificate on the server and ensured it contains a wildcard in the left-most label immediately preceding the public suffix (*.local) in the CN field. • The evaluator configured the database server to leverage the uploaded server certificate and the key for TLS handshake. • The TOE automatically configures references identifiers based on the FQDN or IP address configured used to specify the TLS server. The evaluator attempted a connection to the server with identifier “sqlserver.acumensec.local” resolved to 10.1.3.51 and ensured that the TOE could not connect to the server. • The evaluator observed the packet capture and ensured that the client does not complete the TLS handshake due to “certificate unknown” message. • The evaluator further observed the logs on the TOE located at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs to confirm that the TOE could not connect to the server as The Common Alternative Name '*.local' is not allowed according to the strict hostname verification policy.

Pass/Fail with Explanation	Pass. The evaluator verified that the TLS connection fails when the server presented a certificate with wildcard in the left-most label of the presented identifier immediately preceding the public suffix while the reference identifier configured on the TOE contain two left most labels.

6.13.22 FCS_TLSC_EXT.1.3 Test #1a

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that a server using a certificate with a valid certification path successfully connects. **TD0513 has been applied.**
Expected Results	As a part of FIA_X509_EXT.1.1 Test #1, • The evaluator attempted a connection from the TOE (TLS client) to the MySQL server which is presenting a certificate with a valid certificate path and confirmed that the TLS connection successfully established as the resource test was successful. • The evaluator observed the packet capture on the server and confirmed that the TLS handshake was successful with server presenting a certificate with a valid certification path. (Added a Note in FIA_X509_EXT.1.1 Test #1 which satisfied the current requirement)
Pass/Fail with Explanation	Pass. This test is covered as a part of FIA_X509_EXT.1.1 Test #1 where the evaluator demonstrated that a server using a certificate with a valid certification path results in a successful connection.

6.13.23 FCS_TLSC_EXT.1.3 Test #1b

Item	Data
Test Assurance Activity	The evaluator shall modify the certificate chain used by the server in test 1a to be invalid and demonstrate that a server using a certificate without a valid certification path to a trust store element of the TOE results in an authentication failure. **TD0513 has been applied.**
Expected Results	As a part of FIA_X509_EXT.1.1 Test #1, • The evaluator observed the packet capture on the server and confirmed that the TLS handshake was not successful as one of the issuing certificates in the certificate path is not a CA certificate due to which the TOE was unable to construct a valid certificate path which resulted in Certificate Unknown error returned to the server. • The evaluator further observed the debug logs on the TOE at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs to confirm that the TOE was unable to construct a valid chain and the certification path could not be validated as the certificate presented was not a CA certificate. (Added a Note in FIA_X509_EXT.1.1 Test #1 which satisfied the current requirement)
Pass/Fail with Explanation	Pass. This test is covered as a part of FIA_X509_EXT.1.1 Test #1 where the evaluator demonstrated that modifying the certificate chain used by the server to be invalid resulted in an authentication failure as the TOE was unable to construct a valid certificate path reason being one of the issuing certificates in the certificate path is not a CA certificate.

6.13.24 FCS_TLSC_EXT.1.3 Test #1c

Item	Data
Test Assurance Activity	[conditional]: **If the TOE trust store can be managed**, the evaluator shall modify the trust store element used in Test 1a to be untrusted and demonstrate that a connection attempt from the same server used in Test 1a results in an authentication failure. **TD0513 has been applied.**
Expected Test Results	As a part of FIA_X509_EXT.1.1 Test #1, • The evaluator observed the packet capture on the server and confirmed that the TLS handshake was not successful as one of the issuing certificates in the certificate path is not present in the trust store due to which the TOE was unable to construct a valid certificate path which resulted in Certificate Unknown error returned to the server. • The evaluator further observed the debug logs on the TOE at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs to confirm that the TOE was unable to construct a valid chain as no issuer certificate in the trust store or the certification path was found. (Added a Note in FIA_X509_EXT.1.1 Test #1 which satisfied the current requirement)
Pass/Fail with Explanation	Pass. This test is covered as a part of FIA_X509_EXT.1.1 Test #1 where the evaluator demonstrated that modifying the trust store element to be untrusted (by deleting the Intermediate certificate from the trust store) and attempting a connection from the server resulted in an authentication failure as the TOE was unable to construct a valid certificate path reason being one of the issuing certificates in the certificate path is not present in the TOE's trust store.

6.13.25 FCS_TLSC_EXT.1.3 Test #2

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that using an invalid certificate (unless excepted) results in the function failing as follows, unless excepted: Test 2: The evaluator shall demonstrate that a server using a certificate which has been revoked results in an authentication failure.
Expected Results	As a part of FIA_X509_EXT.1.1 Test #3, • The evaluator observed the packet capture on the server and verified that the client fetched the CRLs required to validate the certificates and ensured to return a fatal alert to the server as the server certificate was revoked. • The evaluator further observed the debug logs on the TOE at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs to confirm that the Certificate 82:F7:34:04:5D:C4:24:C3:64:0E:A1:16:2E:16:2B:04:0D:32:CB:C2 has been revoked by CRL at 'http://10.1.3.51/ICA2.crl' which corresponds to the server certificate. (Added a Note in FIA_X509_EXT.1.1 Test #3 which satisfied the current requirement)
Pass/Fail with Explanation	Pass. This test is covered as a part of FIA_X509_EXT.1.1 Test #3 where the evaluator demonstrated that a server using a certificate that was revoked resulted in an authentication failure.

6.13.26 FCS_TLSC_EXT.1.3 Test #3

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that using an invalid certificate (unless excepted) results in the function failing as follows, unless excepted: Test 3: The evaluator shall demonstrate that a server using a certificate which has passed its expiration date results in an authentication failure.
Expected Results	As a part of FIA_X509_EXT.1.1 Test #2, • The evaluator observed the packet capture on the server and confirmed that the TLS handshake was not successful with certificate unknown alert returned by the TOE. • The evaluator further observed the debug logs on the TOE at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs to confirm that the server certificate expired on 20210212050000GMT+00:00 which indicates the function failing. (Added a Note in FIA_X509_EXT.1.1 Test #2 which satisfied the current requirement)
Pass/Fail with Explanation	Pass. This test is covered as a part of FIA_X509_EXT.1.1 Test #2 where the evaluator demonstrated that a server using a certificate which has passed its expiration date results in an authentication failure.

6.13.27 FCS_TLSC_EXT.1.3 Test #4

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that using an invalid certificate (unless excepted) results in the function failing as follows, unless excepted: Test 4: The evaluator shall demonstrate that a server using a certificate which does not have a valid identifier results in an authentication failure.
Expected Results	As a part of FCS_TLSC_EXT.1.2 Test #2, • The evaluator observed the packet capture and ensured that the client does not complete the TLS handshake due to “certificate unknown” message returned by the TOE. • The evaluator further observed the logs on the TOE located at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs to confirm that the TOE could not connect to the server as there was no subject alternative name found matching IP address 10.1.3.51. (Added a Note in FCS_TLSC_EXT.1.2 Test #2 which satisfied the current requirement)
Pass/Fail with Explanation	Pass. This test is covered as a part of FCS_TLSC_EXT.1.2 Test #2 where the evaluator demonstrated that a server using a certificate which does not have a valid identifier in the SAN results in an authentication failure.

6.13.28 FCS_TLSC_EXT.2.1 Test #1

Item	Data
Test Assurance Activity	The evaluator shall establish a connection to a server that is not configured for mutual authentication (i.e. does not send Server’s Certificate Request (type 13) message). The evaluator observes negotiation of a TLS channel and confirms that the TOE did not send Client’s Certificate message (type 11) during handshake.
Test Steps	Note: As per the TSS, The TOE supports TLS mutual authentication for FTP/s, AS2, and HTTPS connections. The evaluator configured the TOE as a TLS client to establish a connection with the HTTPS server used for Remote File transfers. User File transfers via

	HTTPS on the GoAnywhere MFT is configured as Projects. The HTTPS server for these projects is linked via the server resource shown below • The evaluator created a client certificate for the TOE that was signed by the RootCA. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator created a key vault named "HTTPS" in encryption > Key Management system > Add key Valult. • The evaluator uploaded the client certificate in the key vault on the TOE. • The evaluator configured the TOE to reach 10.1.3.51 as the HTTPS server on port 445 in Resources > HTTPS Servers. • The evaluator selected the uploaded client certificate to be used for client authentication which ensured that mutual authentication must be performed as these options must be selected if client authentication is required. Note: The evaluator has chosen the tool as a remote server hence the evidence is limited to showing the successfully connection between the client and server. It is important to note this HTTPS connection is used for Remote file transfers on the TOE as stated in the TSS. • The evaluator used the "acumen-tlsc-pkg" tool to ensure that the server does not perform mutual authentication (i.e., does not send Server's Certificate Request (type 13) message) to establish a TLS connection with the TOE. The evaluator observed that the TLS connection was successful with application data received as seen in the tool output. • The evaluator observed the packet capture between the TOE and the server and ensured that the server did not send Server's Certificate Request (type 13) message and the TOE did not send the Client's Certificate message (type 11) during handshake.
Pass/Fail with Explanation	Pass. The evaluator observed the negotiation of a TLS channel and confirmed that the TOE did not send Client's Certificate message (type 11) during handshake when the server is not configured for mutual authentication.

6.13.29 FCS_TLSC_EXT.2.1 Test #2

Item	Data
Test Assurance Activity	The evaluator shall establish a connection to a server with a shared trusted root that is configured for mutual authentication (i.e. it sends Server's Certificate Request (type 13) message). The evaluator observes negotiation of a TLS channel and confirms that the TOE responds with a non-empty Client's Certificate message (type 11) and Certificate Verify (type 15) message.
Test Steps	Note: As per the TSS, The TOE supports TLS mutual authentication for FTP/s, AS2, and HTTPS connections. The evaluator configured the TOE as a TLS client to establish a connection with the HTTPS server used for Remote File transfers. User File transfers via HTTPS on the GoAnywhere MFT is configured as Projects. The HTTPS server for these projects is linked via the server resource shown below • The evaluator created a client certificate for the TOE that was signed by the RootCA. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate.

	• The evaluator created a key vault named "HTTPS" in encryption > Key Management system > Add key Valult. • The evaluator uploaded the client certificate in the key vault on the TOE. • The evaluator configured the TOE to reach 10.1.3.51 as the HTTPS server on port 445 in Resources > HTTPS Servers. • The evaluator selected the uploaded client certificate to be used for client authentication which ensured that mutual authentication must be performed as these options must be selected if client authentication is required. Note: The evaluator has chosen the tool as a remote server hence the evidence is limited to showing the successfully connection between the client and server. It is important to note this HTTPS connection is used for Remote file transfers on the TOE as stated in the TSS. • The evaluator used the "acumen-tlsc-pkg" tool to ensure that the server supports mutual authentication. The evaluator observed that the TLS connection was successful with mutual authentication and the application data received as seen in the tool output. • The evaluator observed the packet capture between the TOE and the server and ensured that the server sent a Server's Certificate Request (type 13) message confirms that the TOE responds with a non-empty Client's Certificate message (type 11) and Certificate Verify (type 15) message.
Pass/Fail with Explanation	Pass. The evaluator observed the negotiation of a TLS channel and confirmed that the TOE responds with a non-empty Client's Certificate message (type 11) and Certificate Verify (type 15) message when the server is configured for mutual authentication.

6.13.30 FCS_TLSC_EXT.5.1 Test #1

Item	Data
Test Assurance Activity	The evaluator shall configure a server to perform key exchange using each of the TOE's supported curves and/or groups. The evaluator shall verify that the TOE successfully connects to the server.
Test Steps	• The evaluator imported the RootCA_ec certificate into the keyvault of the TOE that signed the ec server certificate. • The evaluator created a server ec certificate using the XCA tool. • The evaluator uploaded the server certificate and the server key to the Database server. • The evaluator used the "acumen-tlsc-mysql" tool to attempt a TLS connection using each of the supported elliptic curves. The evaluator observed that the TLS connection was successful using secp256r1 and secp384r1 and the application data received as seen in the tool output. • The evaluator observed the packet capture and ensured that the TLS handshake was successful using secp256r1 curve. • The evaluator observed the packet capture and ensured that the TLS handshake was successful using secp384r1 curve.
Pass/Fail with Explanation	Pass. The evaluator verified that the TOE successfully connects to the server with each of its supported curves when the server is configured to perform key exchange using each of the TOE's supported curves and/or groups.

6.14 PKG_TLSS (Windows)

6.14.1 FCS_TLSS_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	The evaluator shall establish a TLS connection using each of the cipher suites specified by the requirement. This connection may be established as part of the establishment of a higher-level protocol, e.g., as part of an EAP session. It is sufficient to observe the successful negotiation of a cipher suite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic in an attempt to discern the cipher suite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).

Test Steps	• The evaluator created a server certificate that was signed by the Root CA using the XCA tool.
	• The evaluator ensured that the server certificate was signed by the RootCA.
	• The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate.
	• The evaluator loaded the server certificate (https_server.pem) along with the key in pem format to the TOE's system keyvault.
	• The evaluator configured the TOE's Administration server in System > Admin server where the port was set 8001 in General.
	• The evaluator selected the TLS protocol, supported cipher suites and the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and saved the configuration.
	• The evaluator used the openssl s_client resource to establish the TLS connection with each of the cipher suite supported by the TOE as server and verified the connection to be successful.
	  ▪ TLS_RSA_WITH_AES_128_CBC_SHA
	  ▪ TLS_RSA_WITH_AES_256_CBC_SHA
	  ▪ TLS_RSA_WITH_AES_128_CBC_SHA256
	  ▪ TLS_RSA_WITH_AES_256_CBC_SHA256
	  ▪ TLS_RSA_WITH_AES_128_GCM_SHA256
	  ▪ TLS_RSA_WITH_AES_256_GCM_SHA384
	  ▪ TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
	  ▪ TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
	• The evaluator observed the packet capture and ensured that the TLS connection was successful where the server responded with the cipher suite configured on the client side.
	• The evaluator created a server ec certificate that was signed by the Root CA_ec using the XCA tool.

	• The evaluator ensured that the server ec certificate was signed by the RootCA_ec.
	• The evaluator imported the RootCA_ec certificate into the key vault of the TOE that signed the server certificate.
	• The evaluator loaded the server certificate (https_server_ec.pem) along with the key in pem format to the TOE's system keyvault.
	• The evaluator configured the TOE's Administration server in System > Admin server where the port was set 8001 in General.
	• The evaluator selected the TLS protocol, supported cipher suites and the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and saved the configuration.
	• The evaluator used the openssl s_client resource to establish the TLS connection with each of the cipher suite supported by the TOE as server and verified the connection to be successful.
	  ▪ TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
	  ▪ TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
	  ▪ TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
	  ▪ TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
	• The evaluator observed the packet capture and ensured that the TLS connection was successful where the server responded with the cipher suite configured on the client side.

Pass/Fail with Explanation	Pass. The TOE successfully connected with all claimed algorithms. This meets testing requirements.

6.14.2 FCS_TLSS_EXT.1.1 Test #2

Item	Data
Test Assurance Activity	The evaluator shall send a Client Hello to the server with a list of cipher suites that does not contain any of the cipher suites in the server's ST and verify that the server denies the connection. Additionally, the evaluator shall send a Client Hello to the server containing only the TLS_NULL_WITH_NULL_NULL cipher suite and verify that the server denies the connection.
Test Steps	• The evaluator created a server certificate that was signed by the Root CA using the XCA tool. • The evaluator ensured that the server certificate was signed by the RootCA. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator loaded the server certificate (https_server.pem) along with the key to the TOE's system keyvault. • The evaluator configured the TOE's Administration server in System > Admin server where the port was set 8001 in General. • The evaluator selected the TLS protocol, supported cipher suites and the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and saved the configuration. • The evaluator used the "acumen-tlss-pkg" tool to send a Client Hello to the server with a cipher suite that is not present in the list of the cipher suites claimed in the

	server's ST and verified that the server denies the connection. Additionally, the evaluator sent a Client Hello to the server using the tool containing only the TLS_NULL_WITH_NULL_NULL cipher suite and verified that the server denied the connection. • The evaluator observed the packet capture (packet 1-2) and ensured that when the client hello was sent to the server with a cipher suite (TLS_RSA_WITH_NULL_MD5) that is not present in the list of the cipher suites claimed in the server's ST, the server denied the connection due to Handshake failure. • The evaluator observed the packet capture (packet 3-4) and ensured that when the client hello was sent to the server with a TLS_NULL_WITH_NULL_NULL cipher suite, the server denied the connection due to Handshake failure.
Pass/Fail with Explanation	Pass. The TOE rejected a connection when the Client Hello consists of a cipher not claimed in the ST or a NULL cipher. This meets testing requirements.

6.14.3 FCS_TLSS_EXT.1.1 Test #3

Item	Data
Test Assurance Activity	If **RSA key exchange is used in one of the selected ciphersuites**, the evaluator shall use a client to send a properly constructed Key Exchange message with a modified EncryptedPreMasterSecret field during the TLS handshake. The evaluator shall verify that the handshake is not completed successfully, and no application data flows.
Test Steps	• The evaluator created a server certificate that was signed by the Root CA using the XCA tool. • The evaluator ensured that the server certificate was signed by the RootCA. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator loaded the server certificate (https_server.pem) along with the key to the TOE's system keyvault. • The evaluator configured the TOE's Administration server in System > Admin server where the port was set 8001 in General. • The evaluator selected the TLS protocol, supported cipher suites and the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and saved the configuration. • The evaluator used the "acumen-tlss-pkg" as a TLS client to send a properly constructed Key Exchange message with a modified EncryptedPreMasterSecret field during the TLS handshake and verified that the handshake is not completed successfully, and no application data flows as per the tool output. • The evaluator observed the packet capture on the client to ensure that the handshake is not completed successfully, and no application data flows when the TLS client sent a properly constructed Key Exchange message with a modified EncryptedPreMasterSecret field during the TLS handshake.

Pass/Fail with Explanation	Pass. The TOE rejects the connection when the client sends a properly constructed key exchange message but a modified EncryptedPreMasterSecret field. This meets the testing requirements.

6.14.4 FCS_TLSS_EXT.1.1 Test #4.1

TD0469 removes this test.

6.14.5 FCS_TLSS_EXT.1.1 Test #4.2

Item	Data
Test Assurance Activity	Modify a byte in the data of the client's Finished handshake message and verify that the server rejects the connection and does not send any application data.
Test Steps	• The evaluator created a server certificate that was signed by the Root CA using the XCA tool. • The evaluator ensured that the server certificate was signed by the RootCA. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator loaded the server certificate (https_server.pem) along with the key to the TOE's system keyvault. • The evaluator configured the TOE's Administration server in System > Admin server where the port was set 8001 in General. • The evaluator selected the TLS protocol, supported cipher suites and the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and saved the configuration. • The evaluator used the acumen-tlss-pkg tool as a TLS client to connect to the server with a modified client's Finished handshake message and verified that the server rejects the connection due to "DECRYPT _ERROR" alert as seen in the tool output. • The evaluator observed the packet capture on the client and ensured that the server rejects the connection due to "DECRYPT _ERROR" alert after the client attempted to connect to the server with a modified client's Finished handshake message.
Pass/Fail with Explanation	Pass. The TOE successfully rejects the connection after receiving the finished message before receiving the change cipher spec message. This meets testing requirements.

6.14.6 FCS_TLSS_EXT.1.1 Test #4.3

Item	Data
Test Assurance Activity	Demonstrate that the TOE will not resume a session for which the client failed to complete the handshake (independent of TOE support for session resumption): **Test 4.3 [conditional]:** If the TOE does not support session resumption based on session IDs according to RFC4346 (TLS1.1) or RFC5246 (TLS1.2) or session tickets according to RFC5077, the evaluator shall perform the following test: a) The evaluator shall send a Client Hello with a zero-length session identifier and with a SessionTicket extension containing a zero-length ticket.

	b) The evaluator shall verify the server does not send a NewSessionTicket handshake message (at any point in the handshake). c) The evaluator shall verify the Server Hello message contains a zero-length session identifier or passes the following steps: Note: The following steps are only performed if the ServerHello message contains a non-zero length SessionID. d) The evaluator shall complete the TLS handshake and capture the SessionID from the ServerHello. e) The evaluator shall send a ClientHello containing the SessionID captured in step d). This can be done by keeping the TLS session in step d) open or start a new TLS session using the SessionID captured in step d). f) The evaluator shall verify the TOE (1) implicitly rejects the SessionID by sending a ServerHello containing a different SessionID and by performing a full handshake (as shown in Figure 1 of RFC 4346 or RFC 5246), or (2) terminates the connection in some way that prevents the flow of application data. **TD0588 Applied**
Test Steps	• The evaluator created a server certificate that was signed by the Root CA using the XCA tool. • The evaluator ensured that the server certificate was signed by the RootCA. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator loaded the server certificate (https_server.pem) along with the key to the TOE's system keyvault. • The evaluator configured the TOE's Administration server in System > Admin server where the port was set 8001 in General. • The evaluator selected the TLS protocol, supported cipher suites and the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and saved the configuration. • The evaluator used the acumen-tlss-pkg tool to generate a Fatal alert during the TLS handshake from the client before the client sends a ChangeCipherSpec message and then sent a Client Hello with the session identifier from the previous incomplete session and verified that the server does not resume the previous dead session. The connection succeeded as per the tool output as the TOE does not set a session ID which implies that there was no dead session. • The evaluator observed the packet capture (packet 1-5) and observed the Fatal alert during the TLS handshake from the client before the client sends a ChangeCipherSpec message. • The evaluator then observed the packet capture (packet 6-13) to ensure that the connection succeeded as that the TOE does not set a session ID which implies that there was no dead session. • The evaluator ensured that a client hello was sent with a zero-length session identifier and with a SessionTicket extension containing a zero-length ticket as

	highlighted below (which implies that the client is not specifying a session to resume). • The evaluator also verified through the packet capture that the server did not send a NewSessionTicket handshake message (at any point in the handshake) which implies that the server does not support Session tickets. • The evaluator verified that the Server Hello message contains a zero-length session identifier as seen in packet 2 which confirms that the TOE does not support session resumption using session IDs.
Pass/Fail with Explanation	Pass. The TOE does not send a session ID when attempted to connect with the previous incomplete session which implies there are no dead sessions. The TOE sends a Server Hello message containing a zero-length session identifier in response to a Client Hello with a zero-length session identifier which implies that the TOE does not support session resumption using session IDs. The TOE sends oes not send a NewSessionTicket handshake message (at any point in the handshake) which implies that the TOE does not support session resumption using Session tickets. This meets the testing requirements.

6.14.7 FCS_TLSS_EXT.1.1 Test #4.4

Item	Data
Test Assurance Activity	Send a message consisting of random bytes from the client after the client has issued the ChangeCipherSpec message and verify that the server denies the connection.
Test Steps	• The evaluator created a server certificate that was signed by the Root CA using the XCA tool. • The evaluator ensured that the server certificate was signed by the RootCA. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator loaded the server certificate (https_server.pem) along with the key to the TOE's system keyvault. • The evaluator configured the TOE's Administration server in System > Admin server where the port was set 8001 in General. • The evaluator selected the TLS protocol, supported cipher suites and the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and saved the configuration. • The evaluator used the acumen-tlss-pkg tool to attempt a connection to the TOE with a TLS connection and sent a message ("this is a garbled message") from the client after the client has issued the ChangeCipherSpec message and verified that a fatal alert was returned by the server. • The evaluator observed the packet capture and ensured that the client sent a random data which appears as application data before sending the finished message and ensured that the server returned an alert message.
Pass/Fail with Explanation	Pass. The TOE successfully rejects a connection when receiving a garbled message after the ChangeCipherSpec message. This meets testing requirements.

6.14.8 FCS_TLSS_EXT.1.2 Test #1

Item	Data
Test Assurance Activity	The evaluator shall send a Client Hello requesting a connection with version SSL 2.0 and verify that the server denies the connection. The evaluator shall repeat this test with SSL 3.0 and TLS 1.0, and TLS 1.1 if it is selected.
Test Steps	• The evaluator created a server certificate that was signed by the Root CA using the XCA tool. • The evaluator ensured that the server certificate was signed by the RootCA. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator loaded the server certificate (https_server.pem) along with the key to the TOE's system keyvault. • The evaluator configured the TOE's Administration server in System > Admin server where the port was set 8001 in General. • The evaluator selected the TLS protocol, supported cipher suites and the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and saved the configuration. • The evaluator used the acumen-tlss-pkg tool to initiate the TLS connection with SSL 3.0, SSL 2.0, TLS 1.0, TLS 1.1 and verified that the server returned a fatal alert for each non-supported TLS versions. • The evaluator observed the packet capture on the client and ensured that the server returned a fatal alert when the client attempted a connection with SSL 3.0 version. • The evaluator observed the packet capture on the client and ensured that the server reset the connection with a FIN packet and a reset packet when the client attempted a connection with SSL 2.0 version. • The evaluator observed the packet capture on the client and ensured that the server returned a fatal alert when the client attempted a connection with TLS 1.1 version. • The evaluator observed the packet capture on the client and ensured that the server returned a fatal alert when the client attempted a connection with TLS 1.0 version.
Pass/Fail with Explanation	Pass. The evaluator ensured that the server returned a fatal alert or reset packet when the client attempted to connect with an unsupported TLS version in the client hello. This meets the testing requirements.

6.14.9 FCS_TLSS_EXT.1.3 Test #1

Item	Data
Test Assurance Activity	Note that this testing can be accomplished in conjunction with other testing activities. For each of the following tests, determining that the size matches the expected size is sufficient. **Test 1:** [conditional] If **RSA-based key establishment is selected**, the evaluator shall configure the TOE with a certificate containing a supported RSA size and attempt a connection. The evaluator shall verify that the size used matches that which is configured and that the

	connection is successfully established. The evaluator shall repeat this test for each supported size of RSA-based key establishment.
Test Steps	• The evaluator created a server certificate containing a supported RSA size of 2048 bit that was signed by the Root CA using the XCA tool. • The evaluator ensured that the server certificate was containing the RSA size of 2048 bit and was signed by the RootCA. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator loaded the server certificate (https_server_2048.pem) along with the key to the TOE's system keyvault. • The evaluator configured the TOE's Administration server in System > Admin server where the port was set 8001 in General. • The evaluator configured the TLS protocol, supported cipher suites and the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and saved the configuration. • The evaluator initiated a TLS connection using the openssl s_client resource to the TOE and ensured that the TLS handshake was successful with RSA key size of 2048 bit. • The evaluator observed the packet capture on the client and verified that the connection established with RSA size used (256*8=2048 bit). • The evaluator created a server certificate containing a supported RSA size of 3072 bit that was signed by the Root CA using the XCA tool. • The evaluator ensured that the server certificate was containing the RSA size of 3072 bit and was signed by the RootCA. • The evaluator loaded the server certificate (https_server_3072.pem) along with the key to the TOE's system keyvault. • The evaluator configured the TLS protocol, supported cipher suites and the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and saved the configuration. • The evaluator initiated a TLS connection using the openssl s_client resource to the TOE and ensured that the TLS handshake was successful with RSA key size of 3072 bit. • The evaluator observed the packet capture on the client and verified that the connection established with RSA size used (384*8=3072 bit). • The evaluator created a server certificate containing a supported RSA size of 4096 bit that was signed by the Root CA using the XCA tool. • The evaluator ensured that the server certificate was containing the RSA size of 4096 bit and was signed by the RootCA. • The evaluator loaded the server certificate (https_server_4096.pem) along with the key to the TOE's system keyvault. • The evaluator configured the TLS protocol, supported cipher suites and the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and saved the configuration.

	• The evaluator initiated a TLS connection using the openssl s_client resource to the TOE and ensured that the TLS handshake was successful with RSA key size of 4096 bit. • The evaluator observed the packet capture on the client and verified that the connection established with RSA size used (512*8=4096 bit).
Pass/Fail with Explanation	Pass. The evaluator verified that the TLS connection is successfully established with each supported RSA key size of 2048,3072 and 4096 bits. This meets the testing requirements.

6.14.10 FCS_TLSS_EXT.1.3 Test #3

Item	Data
Test Assurance Activity	Note that this testing can be accomplished in conjunction with other testing activities. For each of the following tests, determining that the size matches the expected size is sufficient. [conditional] If **ECDHE ciphers are selected**, the evaluator shall attempt a connection using an ECDHE ciphersuite with a supported curve. The evaluator shall verify that the key agreement parameters in the Key Exchange message are the ones configured. The evaluator shall repeat this test for each supported elliptic curve.
Test Steps	• The evaluator created a server certificate that was signed by the Root CA using the XCA tool. • The evaluator ensured that the server certificate was signed by the RootCA. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator loaded the server certificate (https_server.pem) along with the key to the TOE's system keyvault. • The evaluator configured the TOE's Administration server in System > Admin server where the port was set 8001 in General. • The evaluator selected the TLS protocol, supported cipher suites and the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and saved the configuration. • The evaluator attempted a connection to the TOE using an ECDHE cipher suite (ECDHE-RSA-AES128-GCM-SHA256) with supported EC-DH curve (secp256r1) and verified that the TLS connection was successful. • The evaluator observed the packet capture on the client to ensure that TLS connection was successfully established with the TOE using an ECDHE cipher suite (ECDHE-RSA-AES128-GCM-SHA256) with supported EC-DH curve (secp256r1) and also confirmed that key agreement parameters in the key exchange are the ones configured. • The evaluator attempted a connection to the TOE using an ECDHE cipher suite (ECDHE-RSA-AES128-GCM-SHA256) with supported EC-DH curve (secp384r1) and verified that the TLS connection was successful. • The evaluator observed the packet capture on the client to ensure that TLS connection was successfully established with the TOE using an ECDHE cipher suite (ECDHE-RSA-AES128-GCM-SHA256) with supported EC-DH curve (secp384r1) and

	also confirmed that key agreement parameters in the key exchange are the ones configured.
Pass/Fail with Explanation	Pass. The evaluator verified that the TOE establishes a successful TLS connection with all supported EC-DH curves. This meets the testing requirements.

6.14.11 FCS_TLSS_EXT.2.2 Test #1

Item	Data
Test Assurance Activity	The evaluator shall configure the server to send a certificate request to the client. The client shall send a certificate list structure which has a length of zero. The evaluator shall verify that the handshake is not finished successfully, and no application data flows.
Test Steps	• The evaluator created a server certificate that was signed by the Root CA using the XCA tool. • The evaluator ensured that the server certificate was signed by the RootCA. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator loaded the server certificate (https_server.pem) along with the key to the TOE's system keyvault. • The evaluator configured the TOE's Administration server in System > Admin server where the port was set 8001 in General. • The evaluator selected the TLS protocol, supported cipher suites, the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and also configured the server to send a client request by selecting the Client authentication as required and saved the configuration. • The evaluator created a client certificate (10.1.3.51_httpsclient.crt) and the client key (client_key.pem) using XCA tool. • The evaluator used the acumen-tlss-pkg tool as a client to initiate a TLS connection and send a certificate list structure which has a length of zero. The evaluator ensured with the tool output that a fatal alert was returned by the server. • The evaluator observed the packet capture and ensured that the TLS handshake did not finish successfully when the client sent a certificate list structure which has a length of zero.
Pass/Fail with Explanation	Pass. The TOE does not establish a TLS connection with the client when the client sends a certificate list structure which has a length of zero. This meets the testing requirements.

6.14.12 FCS_TLSS_EXT.2.2 Test #2

Item	Data
Test Assurance Activity	The evaluator shall configure the server to send a certificate request to the client. The client shall send no client certificate message, and instead send a client key exchange message in an attempt to continue the handshake. The evaluator shall verify that the handshake is not finished successfully, and no application data flows.
Test Steps	• The evaluator created a server certificate that was signed by the Root CA using the XCA tool.

	• The evaluator ensured that the server certificate was signed by the RootCA. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator loaded the server certificate (https_server.pem) along with the key to the TOE's system keyvault. • The evaluator configured the TOE's Administration server in System > Admin server where the port was set 8001 in General. • The evaluator selected the TLS protocol, supported cipher suites, the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and also configured the server to send a client request by selecting the Client authentication as required and saved the configuration. • The evaluator created a client certificate (10.1.3.51_httpsclient.crt) and the client key (client_key.pem) using XCA tool. • The evaluator used the acumen-tlss-pkg tool to attempt a TLS connection with the server by not sending a client certificate message, and instead send a client key exchange message in an attempt to continue the handshake. The evaluator verified that a fatal alert was returned by the server. • The evaluator observed the packet capture and ensured that the TLS handshake did not finish successfully when the client does not send a client certificate message, and instead send a client key exchange message in an attempt to continue the handshake.
Pass/Fail with Explanation	Pass. The evaluator verified that the TLS handshake did not finish successfully and the server returned a fatal alert when the client does not send a client certificate message, and instead send a client key exchange message in an attempt to continue the handshake. This meets the testing requirements.

6.14.13 FCS_TLSS_EXT.2.2 Test #3

Item	Data
Test Assurance Activity	The evaluator shall configure the server to send a certificate request to the client without the supported_signature_algorithm used by the client's certificate. The evaluator shall attempt a connection using the client certificate and verify that the handshake is not finished successfully, and no application data flows.
Test Steps	• The evaluator created a server certificate that was signed by the Root CA using the XCA tool. • The evaluator ensured that the server certificate was signed by the RootCA. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator loaded the server certificate (https_server.pem) along with the key to the TOE's system keyvault. • The evaluator configured the TOE's Administration server in System > Admin server where the port was set 8001 in General. • The evaluator selected the TLS protocol, supported cipher suites, the server certificate that must be used for TLS handshake which was previously uploaded

	on the TOE and also configured the server to send a client request by selecting the Client authentication as required and saved the configuration. • The evaluator created a client certificate (10.1.3.51_httpsclient.crt) and the client key (client_key.pem) using XCA tool. • The evaluator used the acumen-tlss-pkg tool to attempt a TLS connection with server and modify the signature algorithm used by the client's certificate to an unsupported signature algorithm (RSA_MD5) and verified that a fatal alert was returned by the server. • The evaluator observed the packet capture on the client and ensured that the TLS handshake did not finish successfully when the server received a client certificate with unsupported signature algorithm.
Pass/Fail with Explanation	Pass. The evaluator verified that the TLS handshake did not finish successfully when the server received a client certificate with unsupported signature algorithm.

6.14.14 FCS_TLSS_EXT.2.2 Test #4

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that using a certificate without a valid certification path results in the function failing. Using the administrative guidance, the evaluator shall then load a certificate or certificates needed to validate the certificate to be used in the function and demonstrate that the function succeeds. The evaluator then shall delete one of the certificates and show that the function fails.
Test Steps	• The evaluator created a RootCA certificate that signed the server certificate (https_server) and also created a RootCA_client certificate that signed the client certificate using the XCA tool. • The evaluator ensured that the server certificate was signed by the RootCA. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator loaded the server certificate (https_server.pem) along with the key to the TOE's system keyvault. • The evaluator configured the TOE's Administration server in System > Admin server where the port was set 8001 in General. • The evaluator selected the TLS protocol, supported cipher suites, the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and saved the configuration. • The evaluator confirmed that the RootCA_client certificate that signed the client certificate was not uploaded to the TOE's system keyvault. • The evaluator uploaded the client certificate to the client VM and ensured that the certificate was signed by the RootCA_client certificate. • The evaluator initiated the TLS connection using the client certificate that was signed by the RootCA_client certificate which was not present in the TOE's trust store (system keyvault) and ensured that the TLS connection did not succeed due to certificate unknown error.

	• The evaluator observed the packet capture to ensure that the TLS handshake was not successful as a certificate unknown alert was returned by the server after the client sent the certificate. • Note: A TLS handshake is considered to be successful when the server can validate the client certificate presented in the TLS handshake with the Client's Root certificate imported in the server's trust store. This results in exchange of application data packets. • The evaluator further observed the logs located at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs to ensure that the TLS handshake did not complete as the TOE was unable to construct a valid chain and no issuer certificate for the client certificate in the certificate was found. • The evaluator then loaded the RootCA_client certificate authority needed to validate the client certificate to the TOE's trust store (system keyvault) in Encryption > Key Management services. • The evaluator confirmed that the certificate authority was imported successfully. • The evaluator initiated the TLS connection using the client certificate that was signed by the RootCA_client certificate which was imported in the TOE's trust store (system keyvault) and ensured that the TLS connection established with SSL session observed in the terminal output with no errors observed validating the client certificate presented. • The evaluator observed the packet capture on the client VM to confirm that the TLS connection established with Application Data sent to the server indicating that the Handshake was successful. • Note: A TLS handshake is considered to be successful when the server can validate the client certificate presented in the TLS handshake with the Client's Root certificate imported in the server's trust store. This results in exchange of application data packets. • The evaluator then deleted the RootCA_client certificate authority from the TOE's certificate trust store (system keyvault) that signed the client's certificate. • The evaluator confirmed that the RootCA_client certificate authority was not present in the TOE's trust store. • The evaluator initiated the TLS connection using the client certificate that was signed by the RootCA_client certificate which was not present in the TOE's trust store (system keyvault) and ensured that the TLS connection did not complete due to certificate unknown error. • The evaluator observed the packet capture to ensure that the TLS handshake was not successful as a certificate unknown alert was returned by the server after the client sent the certificate. • Note: A TLS handshake is considered to be successful when the server can validate the client certificate presented in the TLS handshake with the Client's Root certificate imported in the server's trust store. This results in exchange of application data packets.

	• The evaluator further observed the logs located at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs to ensure that the TLS handshake did not complete as the TOE was unable to construct a valid chain and no issuer certificate for the client certificate in the certificate was found.
Pass/Fail with Explanation	Pass. The evaluator verified that when the server was presented with a client certificate without having its issuer certificate in the TOE's trust store or the certification path, the server failed to validate the client certificate which resulted in function failing. This meets the testing requirements.

6.14.15 FCS_TLSS_EXT.2.2 Test #5

Item	Data
Test Assurance Activity	The aim of this test is to check the response of the server when it receives a client identity certificate that is signed by an impostor CA (either Root CA or intermediate CA). To carry out this test the evaluator shall configure the client to send a client identity certificate with an issuer field that identifies a CA recognised by the TOE as a trusted CA, but where the key used for the signature on the client certificate does not in fact correspond to the CA certificate trusted by the TOE (meaning that the client certificate is invalid because its certification path does not in fact terminate in the claimed CA certificate). The evaluator shall verify that the attempted connection is denied.
Test Steps	• The evaluator created a RootCA certificate authority with key RootCA_key that was used to sign the client identity certificate. • The evaluator verified the issuer filed to be CN=RootCA. • The evaluator uploaded the RootCA certificate authority to the TOE's trust store (system keyvault) and ensured that the TOE recognizes the CA with "RootCA" in the issuer field. • The evaluator created a RootCA_imposter certificate authority with the same issuer field "RootCA" that was previously trusted on the TOE but with a different key RootCA_imposter_key and used this certificate authority to sign the client identity certificate. • The evaluator verified the issuer filed to be CN=RootCA that is same as the one trusted by the TOE. • The evaluator uploaded the client identity certificate that was signed by the RootCA_imposter certificate authority to the TLS client VM and ensured it to have an issuer field "CN=RootCA" that identifies the RootCA recognised by the TOE as a trusted CA. • The evaluator initiated the TLS connection using the client identity certificate signed by the Imposter Certificate authority and ensure that the TLS handshake failed due to certificate unknown error returned by the server. • The evaluator observed the packet capture to ensure that the TLS handshake failed due to certificate unknown error returned by the server after receiving the client identity certificate that was signed by the imposter CA. • The evaluator observed the logs on the TOE at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs and ensured that the TLS handshake was not successful as the TOE was unable to construct a valid chain

	and further confirmed that the TOE found the trust anchor, but the certificate validation failed as the certificate does not verify with the supplied key.
Pass/Fail with Explanation	Pass. The evaluator ensured the TOE to respond with a certificate unknown error when it received a client identity certificate that is signed by an impostor CA. This meets the testing requirements.

6.14.16 FCS_TLSS_EXT.2.2 Test #6

Item	Data
Test Assurance Activity	The evaluator shall configure the client to send a certificate with the Client Authentication purpose in the extendedKeyUsage field and verify that the server accepts the attempted connection. The evaluator shall repeat this test without the Client Authentication purpose and shall verify that the server denies the connection. Ideally, the two certificates should be identical except for the Client Authentication purpose.
Test Steps	• The evaluator created a server certificate that was signed by the Root CA using the XCA tool. • The evaluator ensured that the server certificate was signed by the RootCA. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator loaded the server certificate (https_server.pem) along with the key to the TOE's system keyvault. • The evaluator configured the TOE's Administration server in System > Admin server where the port was set 8001 in General. • The evaluator selected the TLS protocol, supported cipher suites, the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and also configured the server to send a client request by selecting the Client authentication as required and saved the configuration. • The evaluator created a client certificate that was signed by the RootCA that was previously uploaded to the TOE's trust store using XCA tool. • The evaluator uploaded the certificate to the Client VM and ensured the client certificate that is being used for a TLS connection contain Client Authentication purpose in the extendedKeyUsage extension. • The evaluator initiated a connection using the openssl s_client resource using the client certificate uploaded and ensured that the TLS handshake was successful. • The evaluator observed the packet capture on the client and confirmed the client certificate to have Client Authentication purpose in the extendedKeyUsage extension and the TLS handshake was successful. • The evaluator created an identical certificate that is similar in structure using xca, the types of identifiers used, and the chain of trust but lacks the Client Authentication purpose in the extendedKeyUsage extension.

	• The evaluator uploaded the client certificate to the client VM and ensured it to be an identical certificate that is similar in structure, the types of identifiers used, and the chain of trust but lacks the Client Authentication purpose in the extendedKeyUsage extension. • The evaluator initiated a TLS connection using the certificate and ensured that the connection was not successful as the server returned a certificate unknown error. • The evaluator observed the packet capture on the client and ensured that TLS handshake was not successful and the server returned the unknown certificate error after receiving the client certificate that lacks Client Authentication purpose in the extendedKeyUsage extension. • The evaluator observed the logs on the TOE at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs and ensured the TLS handshake was not successful due to invalid client extended key usage.
Pass/Fail with Explanation	Pass. The TOE had a successful connection when the certificate did contain the client authentication purpose. The TOE also denied a connection when the certificate did not contain the client authentication purpose. This meets testing requirements.

6.14.17 FCS_TLSS_EXT.2.2 Test #7(a)

Item	Data
Test Assurance Activity	The evaluator shall perform the following modifications to the traffic: Configure the server to require mutual authentication and then modify a byte in the client's certificate. The evaluator shall verify that the server rejects the connection.
Test Steps	• The evaluator created a server certificate that was signed by the Root CA using the XCA tool. • The evaluator ensured that the server certificate was signed by the RootCA. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator loaded the server certificate (https_server.pem) along with the key to the TOE's system keyvault. • The evaluator configured the TOE's Administration server in System > Admin server where the port was set 8001 in General. • The evaluator selected the TLS protocol, supported cipher suites, the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and also configured the server to send for mutual authentication by selecting the Client authentication as required and saved the configuration. • The evaluator used the acumen-tlss-pkg tool to initiate a TLS connection to the server with the client certificate and modify the last byte of the client's certificate during the handshake and confirmed that a Fatal alert was returned by the server. • The evaluator observed the packet capture and ensured that the last byte in the client's certificate was modified from ad to 41 and ensured that a fatal alert "certificate unknown" was returned by the server.

Pass/Fail with Explanation	Pass. The evaluator ensured that the server returns a fatal alert to the client and the TLS handshake fails when it receives a client certificate with a modified byte.

6.14.18 FCS_TLSS_EXT.2.2 Test #7(b)

Item	Data
Test Assurance Activity	The evaluator shall perform the following modifications to the traffic: Configure the server to require mutual authentication and then modify a byte in the signature block of the client's Certificate Verify handshake message. The evaluator shall verify that the server rejects the connection.
Test Steps	• The evaluator created a server certificate that was signed by the Root CA using the XCA tool. • The evaluator ensured that the server certificate was signed by the RootCA. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator loaded the server certificate (https_server.pem) along with the key to the TOE's system keyvault. • The evaluator configured the TOE's Administration server in System > Admin server where the port was set 8001 in General. • The evaluator selected the TLS protocol, supported cipher suites, the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and also configured the server to send for mutual authentication by selecting the Client authentication as required and saved the configuration. • The evaluator used the acumen-tlss-pkg tool to initiate a TLS connection to the server with the client certificate and modify 8 bytes in the signature block of the client's Certificate Verify handshake message during the handshake and confirmed that a Fatal alert was returned by the server. • The evaluator observed the packet capture and ensured that the 8 bytes in the signature block of the client's Certificate Verify handshake message were modified as per the tool output and confirmed that a fatal alert was returned by the server.
Pass/Fail with Explanation	Pass. The evaluator ensured that the server returns a fatal alert to the client and the TLS handshake fails when it modified signature block in the client's Certificate Verify handshake message. This meets testing requirements.

6.14.19 FCS_TLSS_EXT.2.3 Test #1

Item	Data
Test Assurance Activity	The evaluator shall send a client certificate with an identifier that does not match any of the expected identifiers and verify that the server denies the connection. The matching itself might be performed outside the TOE (e.g. when passing the certificate on to a directory server for comparison).
Test Steps	Note: The TOE does the identifier check in the client certificate when the admin users authenticate using certificate to login to the TOE.

- Certificates authenticating Admin Users can be verified by the Admin User's username, email address, or both. The Admin User's username is checked against the Subject Distinguished Name (DN) common name (CN) for a match
- The evaluator created a server certificate that was signed by the Root CA using the XCA tool.
- The evaluator ensured that the server certificate was signed by the RootCA.
- The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate.
- The evaluator loaded the server certificate (https_server.pem) along with the key to the TOE's system keyvault.
- The evaluator configured the TOE's Administration server in System > Admin server where the port was set 8001 in General.
- The evaluator selected the TLS protocol, supported cipher suites, the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and also configured the server to send for mutual authentication by selecting the Client authentication as required and saved the configuration.
- The evaluator created a client certificate for the admin user "test" with an incorrect CN/SAN that was signed by the RootCA trusted on the TOE.
- The evaluator uploaded the certificate to the client VM in p12 format and ensured it to have an incorrect CN=test_incorrect.
- The evaluator created an admin user named "test" with authentication type set to certificate on the TOE and provided the SHA1 fingerprint of the client certificate that is being presented. The evaluator selected the expected identifier to be the username (test) in SAN/DN validation.
- The evaluator ensured that the admin user test was created with authentication type set to certificate and username was selected for SAN/DN validation.
- The evaluator installed the client certificate incorrect CN=test_incorrect in the Firefox browser that will be used to login to the TOE.
- The evaluator attempted to login to the TOE using the Firefox browser with client certificate installed and ensured the login attempt was not successful using the certificate with incorrect CN and the TOE reverted back requesting for authentication credentials.

Note: If a matching certificate is found during the connection, the Admin User will automatically authenticate. However, if a match is not found, the Admin User can still login to the Go Anywhere server with a username and password.

- The evaluator observed the logs on the TOE and ensured that the admin user "test" failed to login.
- The evaluator further observed the debug logs at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs and ensured that the digital certificate that was used for client authentication was invalid.
- The evaluator attempted to connect to the TOE with the certificate and verified that the certificate was deemed invalid as identifier in the CN did not match the expected identifier "username". However, the TLS connection is interpreted to be

	the application layer connection (i.e., administrator GUI connection) and reverting back to requesting for authentication credentials led to observe a successful TLS connection in the packet capture.
Pass/Fail with Explanation	Pass. The evaluator confirmed that when the client presented a certificate with an identifier that did not match the expected identifier by the server for authentication, the server deemed the certificate to be invalid and did not allow the client to authenticate using the certificate. This meets the testing requirements.

6.15 EP_SSHC (Windows)

6.15.1 FCS_COP.1(1) Test #1

Item	Data
Test Assurance Activity	*If perform encryption/decryption services is chosen, the evaluator shall verify that the TSS describes the counter mechanism including rationale that the counter values provided are unique.* *AES-CTR Tests:* • Test 1: Known Answer Tests (KATs) There are four Known Answer Tests (KATs) described below. For all KATs, the plaintext, IV, and ciphertext values shall be 128-bit blocks. The results from each test may either be obtained by the validator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation. To test the encrypt functionality, the evaluator shall supply a set of 10 plaintext values and obtain the ciphertext value that results from encryption of the given plaintext using a key value of all zeros and an IV of all zeros. Five plaintext values shall be encrypted with a 128-bit all zeros key, and the other five shall be encrypted with a 256-bit all zeros key. To test the decrypt functionality, the evaluator shall perform the same test as for encrypt, using 10 ciphertext values as input. To test the encrypt functionality, the evaluator shall supply a set of 10 key values and obtain the ciphertext value that results from encryption of an all zeros plaintext using the given key value and an IV of all zeros. Five of the key values shall be 128-bit keys, and the other five shall be 256-bit keys. To test the decrypt functionality, the evaluator shall perform the same test as

for encrypt, using an all zero ciphertext value as input.

To test the encrypt functionality, the evaluator shall supply the two sets of key values described below and obtain the ciphertext values that result from AES encryption of an all zeros plaintext using the given key values and an IV of all zeros. The first set of keys shall have 128 128-bit keys, and the second shall have 256 256-bit keys. Key_i in each set shall have the leftmost i bits be ones and the rightmost N-i bits be zeros, for i in [1, N]. To test the decrypt functionality, the evaluator shall supply the two sets of key and ciphertext value pairs described below and obtain the plaintext value that results from decryption of the given ciphertext using the given key values and an IV of all zeros. The first set of key/ciphertext pairs shall have 128 128-bit key/ciphertext pairs, and the second set of key/ciphertext pairs shall have 256 256-bit pairs. Key_i in each set shall have the leftmost i bits be ones and the rightmost N-i bits be zeros for i in [1, N]. The ciphertext value in each pair shall be the value that results in an all zeros plaintext when decrypted with its corresponding key.

To test the encrypt functionality, the evaluator shall supply the set of 128 plaintext values described below and obtain the two ciphertext values that result from encryption of the given plaintext using a 128-bit key value of all zeros and using a 256 bit key value of all zeros, respectively, and an IV of all zeros. Plaintext value i in each set shall have the leftmost bits be ones and the rightmost 128-i bits be zeros, for i in [1, 128]. To test the decrypt functionality, the evaluator shall perform the same test as for encrypt, using ciphertext values of the same form as the plaintext in the encrypt test as input.

- Test 2: Multi-Block Message Test
  The evaluator shall test the encrypt functionality by encrypting an i-block message where 1 less-than i less-than-or-equal to 10. For each i the evaluator shall choose a key, IV, and plaintext message of length i blocks and encrypt the message, using the mode to be tested, with the chosen key. The ciphertext shall be compared to the result of encrypting the same plaintext message with the same key and IV using a known good implementation. The evaluator shall also test the decrypt functionality by decrypting an i-block message where 1 less-than i less-than-or-equal to 10. For each i the evaluator shall choose a key and a ciphertext message of length i blocks and decrypt the message, using the mode to be tested, with the chosen key. The plaintext shall be compared to the result of decrypting the same ciphertext message with the same key using a known good implementation.

- Test 3: Monte-Carlo Test
  For AES-CTR mode perform the Monte Carlo Test for ECB Mode on the

	encryption engine of the counter mode implementation. There is no need to test the decryption engine. The evaluator shall test the encrypt functionality using 200 plaintext/key pairs. 100 of these shall use 128 bit keys, and 100 of these shall use 256 bit keys. The plaintext values shall be 128-bit blocks. For each pair, 1000 iterations shall be run as follows: For AES-ECB mode # Input: PT, Key for i = 1 to 1000: CT[i] = AES-ECB-Encrypt(Key, PT) PT = CT[i] • The ciphertext computed in the 1000th iteration is the result for that trial. This result shall be compared to the result of running 1000 iterations with the same values using a known good implementation.
Pass/Fail with Explanation	Pass. The testing requirements have been satisfied by validating each of the claimed cryptographic algorithms for conformance to the requirements specified in their respective standards

6.15.2 FCS_SSHC_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	The evaluator will, for each public key algorithm supported, show that the TOE supports the use of that public key algorithm to authenticate a user connection to an SSH server. Any configuration activities required to support this test shall be performed according to instructions in the guidance documentation.
Test Steps	• Attempt a connection to a remote SSH server configured with the TOE public key. **SSH-RSA** • Generate a ssh key pair. • Load the private key onto the TOE. • The corresponding public key is added to the authorized_keys file on the server. • Choose the authentication method as public key on the TOE. • Attempt a connection from the TOE to the server with user acumensec using an rsa private key. • Note the logs to verify that the user was successfully authenticated using ssh-rsa algorithm. • The evaluator observed the logs on the server to verify that the TOE uses ssh-rsa (userauth_pubkey) public key algorithm to authenticate the user 'acumensec' to an SSH server. • Verify with packet capture that the connection is successful.

	Rsa-sha2-256 • Choose the authentication method as public key on the TOE. • Attempt a connection from the TOE to the server with user acumensec using a rsa private key . Not the RSA signature algorithm Is now set to rsa-sha2-256. • The evaluator observed the logs on the server to verify that the TOE uses rsa-sha2-256 (userauth_pubkey) public key algorithm to authenticate the user 'acumensec' to an SSH server. • Verify with packet capture that is connection is successful. **Rsa-sha2-512** • Choose the authentication method as public key. • Attempt a connection from the TOE to the server with user acumensec using a rsa private key . The corresponding public key is added to the authorized_keys file on the server. • The evaluator observed the logs on the server to verify that the TOE uses rsa-sha2-512 (userauth_pubkey) public key algorithm to authenticate the user 'acumensec' to an SSH server. • Verify with packet capture that the connection is successful. **Ecdsa-sha2-nistp256** • Generate a ssh key pair. • Load the private key onto the TOE. • Choose the authentication method as public key and the Host Key Signature Algorithm as ecdsa-sha2-nistp256. • Attempt a connection from the TOE to the server with user acumensec using an ecdsa private key . The corresponding public key is added to the authorized_keys file on the server. • The evaluator observed the logs on the server to verify that the TOE uses ecdsa-sha2-nistp256 (userauth_pubkey) public key algorithm to authenticate the user 'acumensec' to an SSH server. • Verify with packet capture that the connection is successful. **Ecdsa-sha2-nistp384** • Generate a ssh key pair. • Load the private key onto the TOE. • Choose the authentication method as public key on the TOE. • Attempt a connection from the TOE to the server with user acumensec using an ecdsa private key . The corresponding public key is added to the authorized_keys file on the server. • The evaluator observed the logs on the server to verify that the TOE uses ecdsa-sha2-nistp384 (userauth_pubkey) public key algorithm to authenticate the user 'acumensec' to an SSH server. • Verify with packet capture that the connection is successful.
Pass/Fail with Explanation	Pass. The user connection from the TOE to the server is authenticated successfully using each of the public key algorithms.

6.15.3 FCS_SSHC_EXT.1.1 Test #2

Item	Data

Test Assurance Activity	[Conditional] Using the guidance documentation, the evaluator will configure the TOE to perform password-based authentication to an SSH server, and demonstrate that a user can be successfully authenticated by the TOE to an SSH server using a password as an authenticator. **TD0420 has been applied.**
Test Steps	• The evaluator configured the TOE to perform password-based authentication to an SSH server. • Attempt a connection to an SSH server and show the user authentication succeeds. • Verify with packet capture to ensure that the connection was successful.
Pass/Fail with Explanation	Pass. The user can be successfully authenticated from the TOE to the server using a valid password.

6.15.4 FCS_SSHC_EXT.1.2 Test #1

Item	Data
Test Assurance Activity	The evaluator will demonstrate that if the TOE receives a packet larger than that specified in this component, that packet is dropped.
Test Steps	• Connect to a remote server running a tool that will send a packet larger than the allow packet size. • The evaluator ensured that the TOE terminates the connection.
Pass/Fail with Explanation	Pass. The connection between the TOE and server was terminated after the TOE received a large packet.

6.15.5 FCS_SSHC_EXT.1.3 Test #1

Item	Data
Test Assurance Activity	The evaluator will establish an SSH connection using each of the encryption algorithms specified by the requirement. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test.
Test Steps	• Connect to a remote SSH server using each of the claimed encryption algorithms. **aes128-cbc** • Configure the server to allow only aes128-cbc algorithm. • Attempt a connection from the TOE to the server and verify the connection succeeds. • Verify through packet capture that the connection succeeds. **aes256-cbc** • Configure the server to allow only aes256-cbc algorithm. • Show the connection is successful. • Verify through packet capture the connection succeeds. **aes128-ctr** • Configure the server to allow only aes128-ctr algorithm. • Show the connection is successful. • Verify with packet capture that the connection succeeds. **aes256-ctr**

	• Configure the server to allow only aes256-ctr algorithm. • Show the connection is successful. • Verify with packet capture the connection succeeds.
Pass/Fail with Explanation	Pass. The TOE successfully connects to a remote SSH server using each of the claimed encryption algorithms.

6.15.6 FCS_SSHC_EXT.1.3 Test #2

Item	Data
Test Assurance Activity	The evaluator will configure an SSH server to only allow the 3des-cbc encryption algorithm and no other encryption algorithms. The evaluator will attempt to establish an SSH connection from the TOE to the SSH server and observe that the connection is rejected.
Test Steps	• Configure an SSH server to only allow the 3des-cbc encryption algorithm and no other encryption algorithms. • Attempt a connection to a remote server configured to use 3des-cbc encryption algorithm only. Show the TOE rejects the connection. • Verify with packet capture that the connection is rejected.
Pass/Fail with Explanation	Pass. The connection from the TOE to the server is rejected when attempted with 3des-cbc algorithm.

6.15.7 FCS_SSHC_EXT.1.4 Test #1

Item	Data
Test Assurance Activity	The evaluator will establish a SSH connection using each of the public key algorithms specified by the requirement to authenticate an SSH server to the TOE. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test.
Expected Test Results	As a part of FCS_SSHC_EXT.1.1 Test#1, • The evaluator for each public key algorithm supported, showed using debug logs that the TOE supports the use of that public key algorithm to authenticate a user connection to an SSH server. • The evaluator also verified the successful negotiation of the SSH connection with encrypted packets exchanged between the client and the server which implies that SSH server in response authenticated to the TOE for each of the public key algorithm used to authenticate the user to the SSH server.
Pass/Fail with Explanation	Pass. The test requirements were tested as a part of FCS_SSHC_EXT.1.1 Test#1.

6.15.8 FCS_SSHC_EXT.1.4 Test #2

Item	Data
Test Assurance Activity	The evaluator will configure an SSH server to only allow the ssh-dsa public key algorithm and no other public key algorithms. The evaluator will attempt to establish an SSH connection from the TOE to the SSH server and observe that the connection is rejected.
Test Steps	• Configure an SSH server to only allow the ssh-dsa public key algorithm.

	• Attempt a connection from the TOE to the server and show that the connection is rejected. • Verify through packet capture the connection is rejected.
Pass/Fail with Explanation	Pass. The connection from the TOE to the server is rejected when the server allows only ssh-dsa public key algorithm.

6.15.9 FCS_SSHC_EXT.1.5 Test #1

Item	Data
Test Assurance Activity	The evaluator will establish a SSH connection using each of the integrity algorithms, except "implicit", specified by the requirement. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test. **TD0446 has been applied.**
Test Steps	• Attempt a connection from the TOE to the server using each of the claimed integrity algorithms. **Hmac-sha1** • Configure the server to allow only hmac-sha1 integrity algorithm. • Attempt a connection from the TOE to the server and verify the connection succeeds. • Verify through the packet capture that the connection is successful. **Hmac-sha1-96** • Configure the server to allow only hmac-sha1-96 integrity algorithm. • Attempt a connection from the TOE to the server and verify the connection succeeds. • Verify through the packet capture that the connection is successful. **Hmac-sha2-256** • Configure the server to allow only hmac-sha2-256 integrity algorithm. • Attempt a connection from the TOE to the server and verify the connection succeeds. • Verify through the packet capture that the connection is successful.
Pass/Fail with Explanation	Pass. The connection from the TOE to the server is successful using each of the claimed integrity algorithms.

6.15.10 FCS_SSHC_EXT.1.5 Test #2

Item	Data
Test Assurance Activity	The evaluator will configure an SSH server to only allow the “none” MAC algorithm. The evaluator will attempt to connect from the TOE to the SSH server and observe that the attempt fails. Note: To ensure the proposed MAC algorithm is used, the evaluator shall ensure a non-aes*- gcm@openssh.com encryption algorithm is negotiated while performing this test. **TD0446 has been applied.**

Test Steps	• Attempt to connect to a remote SSH server configured to only support 'none' MAC algorithm using the acumen-sshc tool. • Show that the TOE rejects the connection • Verify through the packet capture that the connection was not successful.
Pass/Fail with Explanation	Pass. The connection from the TOE to the server is rejected when attempted to connect with the "none" mac algorithm.

6.15.11 FCS_SSHC_EXT.1.5 Test #3

Item	Data
Test Assurance Activity	Test 3: The evaluator will configure an SSH server to only allow the hmac- md5 MAC algorithm. The evaluator will attempt to connect from the TOE to the SSH server and observe that the attempt fails. Note: To ensure the proposed MAC algorithm is used, the evaluator shall ensure a non-aes*- gcm@openssh.com encryption algorithm is negotiated while performing this test. **TD0446 has been applied.**
Test Steps	• Configure an SSH server to only allow the hmac- md5 MAC algorithm. • Attempt a connection from the TOE to a server which only allows hmac-md5 MAC algorithm. Show that the connection is rejected. • The evaluator observed the packet capture to ensure that the connection was not successful.
Pass/Fail with Explanation	Pass. The connection from the TOE to the server was rejected when the server allows only hmac-md5 algorithm.

6.15.12 FCS_SSHC_EXT.1.6 Test #1

Item	Data
Test Assurance Activity	The evaluator will configure an SSH server to permit all allowed key exchange methods. The evaluator will attempt to connect from the TOE to the SSH server using each allowed key exchange method and observe that each attempt succeeds.
Test Steps	• The evaluator configured the acumen-sshc tool to be an SSH server waiting for SSH connections on all allowed key exchange methods. The evaluator attempted a connection from the TOE to the server using each of the claimed key exchange methods. **Dh-group14** • The evaluator observed the TOE output and ensured that the SSH connection was successful. • The evaluator observed the packet capture and ensured that the SSH connection was successful using diffie-hellman-group14-sha1 key exchange methods. **Ecdh-sha2-nistp256** • The evaluator observed the TOE output and ensured that the SSH connection was successful. • The evaluator observed the packet capture and ensured that the SSH connection was successful using ecdh-sha2-nistp256 key exchange methods.

	Ecdh-sha2-nistp384 • The evaluator observed the TOE output and ensured that the SSH connection was successful. • The evaluator observed the packet capture and ensured that the SSH connection was successful using ecdh-sha2-nistp384 key exchange methods. **Ecdh-sha2-nistp521** • The evaluator observed the TOE output and ensured that the SSH connection was successful. • The evaluator observed the packet capture and ensured that the SSH connection was successful using ecdh-sha2-nistp521 key exchange methods.
Pass/Fail with Explanation	Pass. The TOE accepts connections to a remote SSH server using each of the claimed key exchange algorithm.

6.15.13 FCS_SSHC_EXT.1.7 Test #1

Item	Data
Test Assurance Activity	The evaluator will configure the TOE to create a log entry when a rekey occurs. The evaluator will connect to the TOE with an SSH client and cause a rekey to occur according to the selection(s) in the ST, and subsequently the evaluator uses available methods and tools to verify that rekeying occurs. This could be done, e.g., by checking that a corresponding audit event has been generated by the TOE, if the TOE supports auditing of rekey events. **TD0331 has been applied.**
Test Steps	Time based rekey • The evaluator configured the maximum seconds before Rekey to 3600 seconds on the TOE system.properties file. • The evaluator configured the SSH server the TOE must reach to perform the test. • The evaluator created a project named 'sftp client' at Workflows -> Projects and set the delay to 61 minutes. • The evaluator executed the project successfully and obtained the log entry file. • The evaluator further checked the log entry file to ensure that initial key exchange occurred at 11:44 am. • The evaluator ensured that rekeying occurred (TOE sent SSH_MSG_KEXINIT initiating rekey as Key re-exchange is started by sending an SSH_MSG_KEXINIT packet) at 12:44 pm prior to reaching the limits in time (3600 seconds-time based rekey). Data Based Rekey • The evaluator configured the maximum number of bytes before Rekey to 1GB on the TOE at system.properties file. • The evaluator configured the SSH server the TOE must reach to perform the test. • The evaluator created a project named 'sftp client' at Workflows -> Projects and set the project to get a file of size 1GB from an SFTP/SSH server. • The evaluator executed the project successfully and obtained the log entry file. • The evaluator further checked the log entry file to ensure that initial key exchange was done between the TOE and the server at 9:00:44pm. • The evaluator ensured that rekeying occurred (TOE sent SSH_MSG_KEXINIT initiating rekey Key re-exchange is started by sending an SSH_MSG_KEXINIT packet) prior to transmitting the file size of 1GB at 9:01:53pm.

Pass/Fail with Explanation	Pass. The TOE rekeys as expected with respect to both time and data.

6.15.14 FCS_SSHC_EXT.1.8 Test #1

Item	Data
Test Assurance Activity	The evaluator will delete all entries in the TOE's list of recognized SSH server host keys and, if selected, all entries in the TOE's list of trusted certification authorities. The evaluator will initiate a connection from the TOE to an SSH server. The evaluator shall ensure that the TOE either rejects the connection or displays the SSH server's public key (either the key bytes themselves or a hash of the key using any allowed hash algorithm) and prompts the user to accept or deny the key before continuing the connection.
Test Steps	• The evaluator ensured that Implicit trust of SSH connections is not allowed by setting the radio button to "No". • The evaluator created an SSH server resource "test" with the server IP set to 10.1.3.51 and port number set to 22. • The TOE does-not store the recognized SSH server host keys but rather store the information on the resource for the SSH/SFTP server the user is connecting to, and the configuration takes place on the Connection tab of the SSH Server resource. The evaluator ensured that the host key was not configured in the SSH server resource. • The evaluator created a Workflow to initiate a connection to the SFTP/SSH server to get files from it and ensured that the connection was not successful. • The evaluator observed the audit logs generated to confirm that the SSH connection was not successful as the host key could not be verified by the TOE.
Pass/Fail with Explanation	Pass. The evaluator ensured that the TOE rejects the SSH connection as the host key could not be verified by the TOE as it was not configured in the TOE's list of recognized SSH server host keys.

6.15.15 FCS_SSHC_EXT.1.8 Test #2

Item	Data
Test Assurance Activity	The evaluator will add an entry associating a host name with a public key into the TOE's local database. The evaluator will replace, on the corresponding SSH server, the server's host key with a different host key. The evaluator will initiate a connection from the TOE to the SSH server using password-based authentication, shall ensure that the TOE rejects the connection, and shall ensure that the password was not transmitted to the SSH server (for example, by instrumenting the SSH server with a debugging capability to output received passwords).
Test Steps	• The evaluator added a public key into the TOE's local database. • The evaluator specified the fingerprint of the server's public key, which will be used to authenticate the server. The evaluator ensured that that SSH connection was successful.

	• The evaluator reconfigured the server such that the server host key is replaced with a new key. • The evaluator added a public key into the TOE's local database. • The evaluator attempted a connection from the TOE to the SSH server using password-based authenticaiton while the older fingerprint of the server's public key was configured and ensured that the connection failed. The evaluator further verified from the ouput that the TOE disconnected from the TOE prior to sending the password which implies that the TOE did not transmit the password to the server. • The evaluator observed the logs on the server and ensured that the SSH connection was not established due to "unknown host key". The evaluator further confirmed that the server received a disconnect from the TOE prior to sending the password which implies that the TOE did not transmit the password to the server. • The evaluator observed the packet capture to ensure that the SSH connection was not successful.
Pass/Fail with Explanation	Pass. The TOE rejects the connection to the server and does not send the password to the SSH server if the host key of the server is replaced with a different key.

6.16 EP_SSH (Windows)

6.16.1 FCS_SSHS_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	The evaluator will, for each public key algorithm supported, show that the TOE supports the use of that public key algorithm to authenticate a user connection from an SSH client. Any configuration activities required to support this test shall be performed according to instructions in the guidance documentation.
Test Steps	• Generate a ssh rsa key pair. • The evaluator observed the private key and public key generated. • Load the public key into the web user SSH keys on the TOE. • Set the web user authentication method to public key. • Set the web user Feature to SFTP. **SSH-RSA** • Choose the server public key signature algorithm to ssh-rsa. • Attempt a connection from the SSH client to the server using a public key to authenticate the user connection and ensure that the connection was successful. The evaluator observed the logs on the client to ensure that the server (TOE) accepted the public-key algorithm ssh-rsa (log: Server accepts key) to authenticate the user connection from the SSH client and the authentication succeeded. • Verify via packet capture that the connection is successful. **RSA-SHA2-256** • Choose the server public key signature algorithm to rsa-sha2-256. • Restart the SFTP service.

	• Attempt a connection from the SSH client to the server using a public key to authenticate the user connection and ensure that the connection was successful. The evaluator observed the logs on the client to ensure that the server (TOE) accepted the public-key algorithm rsa-sha2-256 (log: Server accepts key) to authenticate the user connection from the SSH client and the authentication succeeded. • Verify with wireshark capture that the connection was successful. **RSA-SHA2-512** • Choose the server public key signature algorithm to rsa-sha2-512 • Restart the SFTP service. • Attempt a connection from the SSH client to the server using a public key to authenticate the user connection and ensure that the connection was successful. The evaluator observed the logs on the client to ensure that the server (TOE) accepted the public-key algorithm rsa-sha2-512 (log: Server accepts key) to authenticate the user connection from the SSH client and the authentication succeeded. • Verify via packet capture that the connection is successful. **ECDSA-SHA2-NISTP256** • Generate a ssh ecdsa key pair. • The evaluator observed the private key and public key generated. • Load the public key into the web user SSH keys on the TOE. • Choose the server public key signature algorithm to ecdsa-sha2-nistp256 • Attempt a connection from the SSH client to the server using a public key to authenticate the user connection and ensure that the connection was successful. The evaluator observed the logs on the client to ensure that the server (TOE) accepted the public-key algorithm ecdsa-sha2-nistp256 (log: Server accepts key) to authenticate the user connection from the SSH client and the authentication succeeded. • Verify via packet capture to ensure that the SSH connection was successful. **ECDSA-SHA2-NISTP384** • Choose the server public key signature algorithm to ecdsa-sha2-nistp384 • Attempt a connection from the SSH client to the server using a public key to authenticate the user connection and ensure that the SSH was successful. The evaluator observed the logs on the client to ensure that the server (TOE) accepted the public-key algorithm ecdsa-sha2-nistp384 (log: Server accepts key) to authenticate the user connection from the SSH client and the authentication succeeded. • Verify with packet capture that the SSH connection was successful.
Pass/Fail with Explanation	Pass. The TOE supports each of the public key algorithm to authenticate a user connection.

6.16.2 FCS_SSHS_EXT.1.1 Test #2

Item	Data
Test Assurance Activity	The evaluator shall choose one public key algorithm supported by the TOE. The evaluator shall generate a new key pair for that algorithm without configuring the TOE to recognize the public key for authentication. The evaluator shall use an SSH client to

	attempt to connect to the TOE with the new key pair and demonstrate that authentication fails.
Test Steps	• Set the SSH user authentication type to public key • Create a new keypair without adding the public key to the TOE • Attempt to login using the newly created private key. • Show the connection is rejected and login is unsuccessful. • The evaluator observed the packet capture and ensured that the connection was not successful.
Pass/Fail with Explanation	Pass. The authentication from the client to the TOE fails when attempted to connect with an unknown key.

6.16.3 FCS_SSHS_EXT.1.1 Test #3

Item	Data
Test Assurance Activity	[Conditional] Using the guidance documentation, the evaluator will configure the TOE to perform password-based authentication on a client and demonstrate that a user can be successfully authenticated by the TOE using a password as an authenticator. **TD0420 has been applied.**
Test Steps	• Set the user's authentication type as Password based. • Attempt to login to the TOE using a valid username/password combination. • The evaluator entered the password. • The evaluator ensured that the connection was successful. • The evaluator ensured via packet capture that the SSH connection was successful.
Pass/Fail with Explanation	Pass. The TOE allows a user to successfully authenticate with valid login credentials.

6.16.4 FCS_SSHS_EXT.1.1 Test #4

Item	Data
Test Assurance Activity	[Conditional] The evaluator shall use an SSH client, enter an incorrect password to attempt to authenticate to the TOE, and demonstrate that the authentication fails. **TD0420 has been applied.**
Test Steps	• Attempt a connection to the TOE using an invalid password • Show the connection fails
Pass/Fail with Explanation	Pass. The SSH client fails to authenticate to the TOE with an incorrect password.

6.16.5 FCS_SSHS_EXT.1.2 Test #1

Item	Data
Test Assurance Activity	The evaluator will demonstrate that if the TOE receives a packet larger than that specified in this component, that packet is dropped.

Test Steps	• The evaluator configured the TOE as SFTP/SSH server with authentication type set to password. • The evaluator used the acumen-sshfix tool as an SSH client to send a packet larger than 65535 bytes and ensured that the packet is dropped. • The evaluator observed the packet capture and ensured that the connection was not successful.
Pass/Fail with Explanation	Pass. The TOE drops large packets that are received within an SSH session.

6.16.6 FCS_SSHS_EXT.1.3 Test #1

Item	
Test Assurance Activity	The evaluator will initiate an SSH connection using each of the encryption algorithms specified by the requirement. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test.
Test Steps	• Configure the SSH server on the TOE to enable the claimed encryption algorithms. • The evaluator initiated an SSH connection using aes128-cbc as the encryption algorithm and ensured that the connection established. • The evaluator observed the packet capture and ensured that the SSH connection was successful using aes128-cbc as the encryption algorithm. • The evaluator initiated an SSH connection using aes256-cbc as the encryption algorithm and ensured that the connection established. • The evaluator observed the packet capture and ensured that the SSH connection was successful using aes256-cbc as the encryption algorithm. • The evaluator initiated an SSH connection using aes128-ctr as the encryption algorithm and ensured that the connection established. • The evaluator observed the packet capture and ensured that the SSH connection was successful using aes128-ctr as the encryption algorithm. • The evaluator initiated an SSH connection using aes256-ctr as the encryption algorithm and ensured that the connection established. • The evaluator observed the packet capture and ensured that the SSH connection was successful using aes256-ctr as the encryption algorithm.
Pass/Fail with Explanation	Pass. The connection from a client to the TOE is successful using each of the claimed encryption algorithms.

6.16.7 FCS_SSHS_EXT.1.3 Test #2

Item	Data
Test Assurance Activity	The evaluator will configure an SSH client to only propose the 3des-cbc encryption algorithm and no other encryption algorithms. The evaluator will attempt to establish an SSH connection from the client to the TOE server and observe that the connection is rejected.
Test Steps	• The evaluator attempted to connect to the TOE using 3des-cbc encryption algorithm and no other encryption algorithms. • The evaluator verified that the connection is rejected as the there was no matching cipher found.

	• The evaluator observed the packet capture and ensured that the SSH connection was not successful.
Pass/Fail with Explanation	Pass. The connection from the client to the TOE is rejected when the client attempts to connect with 3des-cbc algorithm.

6.16.8 FCS_SSHS_EXT.1.4 Test #1

Item	Data
Test Assurance Activity	Using an appropriately configured client, the evaluator will establish an SSH connection using each of the public key algorithms specified by the requirement to authenticate. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test.
Expected Test Results	As a part of FCS_SSHS_EXT.1.1 Test#1, • The evaluator for each public key algorithm supported, showed using debug logs that the TOE supports the use of that public key algorithm to authenticate a user connection from an SSH client. • The evaluator also verified the successful negotiation of the SSH connection with encrypted packets exchanged between the client and the server which implies that SSH client in response authenticated to the TOE for each of the public key algorithm used to authenticate the user connection from an SSH client.
Pass/Fail with Explanation	Pass. The test requirements were tested as a part of FCS_SSHC_EXT.1.1 Test#1.

6.16.9 FCS_SSHS_EXT.1.4 Test #2

Item	Data
Test Assurance Activity	The evaluator will configure an SSH client to propose only the ssh-dsa public key algorithm and no other public key algorithms. Using this client, the evaluator will attempt to establish an SSH connection to the TOE and observe that the connection is rejected.
Test Steps	• The evaluator generated a ssh dsa key pair using ssh-keygen. • The evaluator ensured that the SSH key-pair was generated. • The evaluator uploaded the public-key on the TOE. • The evaluator ensures that the dsa public key was uploaded to the TOE with name dsakey. • The evaluator checked the server public-key algorithms and ensured that the ssh-dss (dsa) public key algorithm is not supported by the TOE. • The evaluator attempted to select the uploaded DSA key for the TOE to leverage it as the Host Key by the SFTP server, but the TOE rejected it displaying that there is no valid public key algorithm for the DSA Host key that was attempted to select. • The evaluator attempted a connection from the SSH client to the server using only the ssh-dss (dsa) public key algorithm to authenticate the user connection and observed that the connection was not successful. • The evaluator further verified the packet-capture to ensure that the TOE rejected the connection.

Pass/Fail with Explanation	Pass. The connection attempt between the client and the TOE fails when the public key algorithm is ssh-dsa.

6.16.10 FCS_SSHS_EXT.1.5 Test #1

Item	Data
Test Assurance Activity	Using an appropriately configured client, the evaluator will establish a SSH connection using each of the integrity algorithms, except "implicit", specified by the requirement. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test. **TD0446 has been applied.**
Test Steps	• The evaluator configured the TOE to support the claimed integrity algorithms. • The evaluator used the acumen-sshfix tool as an SSH client to establish an SSH connection using each of the supported integrity algorithms and ensured that the connection was successful. • The evaluator observed the packet capture to ensure that the SSH connection was successful using each of the claimed integrity algorithms.
Pass/Fail with Explanation	Pass. The TOE allows a successful connection from the client using each of the claimed integrity algorithms.

6.16.11 FCS_SSHS_EXT.1.5 Test #2

Item	Data
Test Assurance Activity	The evaluator will configure an SSH client to only allow the “none” MAC algorithm. Using this client, the evaluator will attempt to connect to the TOE and observe that the attempt fails. Note: To ensure the proposed MAC algorithm is used, the evaluator shall ensure a non-aes*- gcm@openssh.com encryption algorithm is negotiated while performing this test. **TD0446 has been applied.**
Test Steps	• The evaluator configured the TOE as an SSH server to support only claimed MAC algorithms. • The evaluator used the acumen-sshfix tool as an SSH client to establish an SSH connection using “none” as the integrity algorithm and ensured that the SSH negotiation failed. • The evaluator observed the packet capture to ensure that the SSH negotiation failed when “none” MAC algorithm was presented by the SSH client.
Pass/Fail with Explanation	Pass. The connection from a client to the server is rejected when using the none MAC algorithm.

6.16.12 FCS_SSHS_EXT.1.5 Test #3

Item	Data

Test Assurance Activity	The evaluator will configure an SSH client to only allow the hmac- md5 MAC algorithm. using this client, the evaluator will attempt to connect to the TOE and observe that the attempt fails Note: To ensure the proposed MAC algorithm is used, the evaluator shall ensure a non-aes*- gcm@openssh.com encryption algorithm is negotiated while performing this test. **TD0446 has been applied.**
Test Steps	• The evaluator attempted a connection to the TOE from a client using hmac-md5 MAC algorithm and ensured that the connection is rejected. • The evaluator observed the packet capture and ensured that connection attempt failed when the SSH client attempted using hmac- md5 MAC algorithm.
Pass/Fail with Explanation	Pass. The connection was rejected when a client attempted to connect with hmac-md5 MAC algorithm.

6.16.13 FCS_SSHS_EXT.1.6 Test #1

Item	Data
Test Assurance Activity	For each of the allowed key exchange methods, the evaluator will configure an SSH client to propose only it and attempt to connect to the TOE and observe that each attempt succeeds.
Test Steps	• The evaluator configured the TOE as an SFTP/SSH server using each of the claimed key exchange methods. • The evaluator used the acumen-sshsfix tool as an SSH client to propose each of the claimed key exchange methods. • The evaluator observed the tool output which indicated that SSH connection succeeded with each of the key exchange methods. (Diffie-hellman-group14-sha1, ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521) • The evaluator observed the packet capture which indicated that SSH connection succeeded with each of the key exchange methods.
Pass/Fail with Explanation	Pass. The TOE allows a client to connect using each of the claimed key exchange methods.

6.16.14 FCS_SSHS_EXT.1.6 Test #2

Item	Data
Test Assurance Activity	The evaluator shall configure an SSH client to only allow the diffiehellman-group1-sha1 key exchange. The evaluator shall attempt to connect from the SSH client to the SSH Server and observe that the attempt fails.
Test Steps	• The evaluator used the acumen-sshsfix tool as an SSH client to propose diffiehellman-group1-sha1 as the claimed key exchange method. • The evaluator observed the tool output and ensured that the connection was rejected.

	• The evaluator observed the packet capture and ensured that the attempt to connect from the SSH client to the SSH Server using diffiehellman-group1-sha1 as the claimed key exchange method failed.
Pass/Fail with Explanation	Pass. The connection to the TOE was unsuccessful when attempted to connect with an unsupported key exchange algorithm.

6.16.15 FCS_SSHS_EXT.1.7 Test #1

Item	Data
Test Assurance Activity	The evaluator will configure the TOE to create a log entry when a rekey occurs. The evaluator will connect to the TOE with an SSH client and cause a rekey to occur according to the selection(s) in the ST, and subsequently the evaluator uses available methods and tools to verify that rekeying occurs. This could be done, e.g., by checking that a corresponding audit event has been generated by the TOE, if the TOE supports auditing of rekey events. **TD0331 has been applied.**
Test Steps	• The evaluator configured the TOE to act as an SSH server with local IP address 10.1.3.50 and port 1214 with authentication type set to password. • The evaluator configured the maximum bytes before rekeying to 1 Gigabyte data in system.properties. • The evaluator used the acumen-sshs tool as an SSH client to connect to the TOE configured as an SSH server. The tool kept the connection alive until more than 1 Gigabyte is transferred using the key and ensured that the SSH connection rekeyed before 1 Gigabyte of data has been transmitted as seen in the tool output. • The evaluator configured the TOE to act as an SSH server with local IP address 10.1.3.50 and port 1214 with authentication type set to password. • The evaluator configured the maximum seconds before rekeying to 3600 seconds in system.properties. • The evaluator used the acumen-sshs tool as an SSH client to connect to the TOE configured as an SSH server. The tool kept the connection alive for more than 1 hour of time using the key and ensured that the SSH connection rekeyed before 1 hour of time has passed as seen in the tool output.
Pass/Fail with Explanation	Pass. The TOE rekeys as expected with respect to both time and data.

6.17 Static Analysis (Windows)

a. Static Analysis

6.17.1 FCS_STO_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	For all credentials for which the application **implements functionality**, the evaluator shall verify credentials are encrypted according to FCS_COP.1(1)
Pass/Fail with Explanation	Pass. The application shall not store any credentials in non-volatile memory.

6.17.2 FDP_DEC_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	**For Windows:** For Windows Universal Applications the evaluator shall check the WMAppManifest.xml file for a list of required hardware capabilities. The evaluator shall verify that the user is made aware of the required hardware capabilities when the application is first installed. This includes permissions such as ID_CAP_ISV_CAMERA, ID_CAP_LOCATION, ID_CAP_NETWORKING, ID_CAP_MICROPHONE, ID_CAP_PROXIMITY and so on. A complete list of Windows App permissions can be found at: • http://msdn.microsoft.com/en-US/library/windows/apps/jj206936.aspx For Windows Desktop Applications the evaluator shall identify in either the application software or its documentation the list of the required hardware resources **Non-applicable platforms removed.** **TD0434 has been applied.** **TD0515 has been applied.**
Pass/Fail with Explanation	Pass. The evaluator reviewed the section named "TOE access to platform resources" in the GoAnywhere MFT Guidance Document and confirmed that Network connectivity is the only hardware platform resource accessed by the TOE.

6.17.3 FDP_DEC_EXT.1.2 Test #1

Item	Data
Test Assurance Activity	**For Windows:** For Windows Universal Applications the evaluator shall check the WMAppManifest.xml file for a list of required capabilities. The evaluator shall identify the required information repositories when the application is first installed. This includes permissions such as ID_CAP_CONTACTS,ID_CAP_APPOINTMENTS,ID_CAP_MEDIALIB and so on. A complete list of Windows App permissions can be found at: • http://msdn.microsoft.com/en-US/library/windows/apps/jj206936.aspx For Windows Desktop Applications the evaluator shall identify in either the application software or its documentation the list of sensitive information repositories it accesses. **Non-applicable platforms removed.** **TD0515 has been applied.**

Pass/Fail with Explanation	Pass. The evaluator reviewed the section named “TOE access to platform resources” in the GoAnywhere MFT Guidance Document and confirmed that System logs are the only sensitive information repository accessed by the TOE.

6.17.4 FPT_AEX_EXT.1.2 Test #1

Item	Data
Test Assurance Activity	The evaluator shall verify that no memory mapping requests are made with write and execute permissions. The method of doing so varies per platform. **For Windows:** The evaluator shall use a tool such as Microsoft's BinScope Binary Analyzer to confirm that the application passes the NXCheck. The evaluator may also ensure that the /NXCOMPAT flag was used during compilation to verify that DEP protections are enabled for the application. **Non-applicable platforms removed.**
Test Steps	• Scan the TOE with Microsoft BinScope with NXCheck and confirm that the TOE passes the NXCheck.
Pass/Fail with Explanation	Pass. The TOE passes NXCheck. This meets testing requirement.

6.17.5 FPT_AEX_EXT.1.5 Test #1

Item	Data
Test Assurance Activity	The evaluator will inspect every native executable included in the TOE to ensure that stack-based buffer overflow protection is present. **For Windows:** Applications that run as Managed Code in the .NET Framework do not require these stack protections. Applications developed in Object Pascal using the Delphi IDE compiled with RangeChecking enabled comply with this element. For other code, the evaluator shall review the TSS and verify that the /GS flag was used during compilation. The evaluator shall run a tool like, BinScope, that can verify the correct usage of /GS. **For PE** , the evaluator will disassemble each and ensure the following sequence appears: mov rcx, QWORD PTR [rsp+(...)] xor rcx, (...) call (...) . **For ELF executables**, the evaluator will ensure that each contains references to the symbol **__stack_chk_fail**. Tools such as Canary Detector may help automate these activities. **Non-applicable platforms removed.**
Test Steps	• When the application is running, scan the TOE’s executable with Process Hacker 2. • Verify the correct usage of the /GS flag.

Pass/Fail with Explanation	Pass. The /GS flag was used during the TOE's compilation on Windows.

6.17.6 FPT_API_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	The evaluator shall then compare the list with the supported APIs (available through e.g. developer accounts, platform developer groups) and ensure that all APIs listed in the TSS are supported.
Test Steps	• The evaluator observed the Windows APIs list in the TSS section of the Security Target and observed that all the PlatformAPIs leverage NLS (Locale.nls), MUI (Kernel32.dll.mui) and DLL (comctl32.dll and others). • The evaluator then compared the list with the supported APIs available through platform developer documentation at https://docs.microsoft.com/en-us/windows/win32/apiindex/windows-api-list and ensured that all resources documented in the TSS were part of the platform documented APIs.
Pass/Fail with Explanation	Pass. The evaluator compared the list with the supported APIs available through platform developer documentation and ensured that all resources documented in the TSS were part of the platform documented APIs.

6.17.7 FPT_TUD_EXT.2.1 Test #1

Item	Data
Test Assurance Activity	The evaluator shall verify that application updates are distributed in the format supported by the platform. This varies per platform: **For Android:** The evaluator shall ensure that the application is packaged in the Android application package (APK) format. **For Windows:** The evaluator shall ensure that the application is packaged in the standard Windows Installer (.MSI) format, the Windows Application Software (.EXE) format signed using the Microsoft Authenticode process, or the Windows Universal Application package (.APPX) format. See https://msdn.microsoft.com/en-us/library/ms537364(v=vs.85).aspx for details regarding Authenticode signing. **For iOS:** The evaluator shall ensure that the application is packaged in the IPA format. **For Linux:** The evaluator shall ensure that the application is packaged in the format of the package management infrastructure of the chosen distribution. For example, applications running on Red Hat and Red Hat derivatives shall be packaged in RPM format. Applications running on Debian and Debian derivatives shall be packaged in DEB format. **For Solaris:** The evaluator shall ensure that the application is packaged in the PKG format. **For macOS:** The evaluator shall ensure that application is packaged in the DMG format, the PKG format, or the MPKG format. **Non-applicable platforms removed.**
Test Steps	• Confirm that the TOE's installation package comes in EXE format. Below screenshot confirms that the TOE installer and the TOE upgrader are packaged in the Standard Windows Application Software (.EXE) format.

	• The evaluator ensured using the following screenshots that the TOE installer and upgrader were signed using Microsoft Authentication Code process. • The Digital Signature Information confirms that the Digital Signature is OK, and the Signer is Help systems(The vendor of TOE). • The evaluator ensured using the following screenshot shows that the Code Signing certificate was issued to Help systems by DigiCert SHA2 Assured ID Code Signing CA.
Pass/Fail with Explanation	Pass. TOE's installation package is in the EXE format and is signed by Microsoft Authentication Code Process.

6.18 X509 (Windows)

6.18.1 FIA_X509_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: - The node certificate to be tested, - Two Intermediate CAs, and - The self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. The evaluator shall demonstrate that validating a certificate without a valid certification path results in the function failing, for each of the following reasons, in turn: • by establishing a certificate path in which one of the issuing certificates is not a CA certificate, • by omitting the basicConstraints field in one of the issuing certificates, • by setting the basicConstraints field in an issuing certificate to have CA=False, • by omitting the CA signing bit of the key usage field in an issuing certificate, and • by setting the path length field of a valid CA field to a value strictly less than the certificate path.

	The evaluator shall then establish a valid certificate path consisting of valid CA certificates and demonstrate that the function succeeds. The evaluator shall then remove trust in one of the CA certificates and show that the function fails. **TD0668 has been applied.**
Test Steps	• The evaluator created a chain of four certificates using the XCA tool. • The evaluator makes necessary changes to this certificate database as per the test requirements. Attempt a connection from the TOE to the server in which one of the issuing certificates is not a CA certificate. **Note: The following test performed satisifies the FCS_TLSC_EXT.1.3 Test #1b requirement where the evaluator demonstrated that modifying the certificate chain used by the server to be invalid resulted in an authentication failure as the TOE was unable to construct a valid certificate path reason being one of the issuing certificates in the certificate path is not a CA certificate.** • The evaluator used the XCA tool to ensure that one of the issuing certificates (ICA2_notCAcert) in the certificate path is not a CA certificate by transforming the original ICA2 issuing certificate to non-CA certificate. • The evaluator uploaded the Self signed CA certificate to the TOE's trust store (system keyvault). • The evaluator uploaded the certificate in pem chain format containing both the ICA1 and ICA2_notCAcert to the mysql directory of the database server. • The evaluator ensured that the certificate in pem chain format contains both ICA1 and ICA2_notCAcert. • The evaluator uploaded the server certificate 10.1.3.51.crt and the server key 10.1.3.51_key.pem to the mysql directory of the database server. • The evaluator configured the MySQL database server to leverage the uploaded server certificates and the server key for the TLS handshake with the client. • The evaluator configured the TOE to reach the Database server at Resources > Database Servers where the JDBC Driver was set to mariadb.jdbc .driver as the server was a MariaDB SQL server and provided the necessary URL information to reach the database server along with username and password of the database created on the server. WADATA is the database created on the MySQL server for the TOE on the Windows Platform • The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources created and confirmed that the client could not connect to the server as the server returned a certificate_unknown error to the client. • The evaluator observed the packet capture on the server and confirmed that the TLS handshake was not successful as one of the issuing certificates in the certificate path is not a CA certificate due to which the TOE was unable to construct a valid certificate path which resulted in Certificate Unknown error returned to the server. • The evaluator further observed the debug logs on the TOE at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs to confirm that the TOE was

unable to construct a valid chain and the certification path could not be validated as the certificate presented was not a CA certificate.

Attempt a connection from the TOE to the server by omitting the basicConstraints field in one of the issuing certificates presented by the server.

- The evaluator used the XCA tool to ensure that one of the issuing certificates (ICA2_nbc) in the certificate path does not have the basicConstraints by transforming the original ICA2 issuing certificate to an ICA_nbc omitting the basicConstraints field.
- The evaluator ensured that the Self signed CA certificate is present in the TOE's trust store (system keyvault).
- The evaluator uploaded the certificate in pem chain format containing both the ICA1 and ICA2_nbc to the mysql directory of the database server.
- The evaluator ensured that the certificate in pem chain format contains both ICA1 and ICA2_nbc.
- The evaluator ensured that the server certificate 10.1.3.51.crt and the server key 10.1.3.51_key.pem is present in the mysql directory of the database server.
- The evaluator configured the MySQL database server to leverage the uploaded server certificates and the server key for the TLS handshake with the client.
- The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the client could not connect to the server as the server returned a certificate_unknown error to the client.
- The evaluator observed the packet capture on the server and confirmed that the TLS handshake was not successful. Note that the ICA2 certificate does not contain basicConstraints in the extension field.
- The evaluator further observed the debug logs on the TOE at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs to confirm that the TOE was unable to construct a valid chain and the certification path could not be validated as the Intermediate certificate lacks basic constraints.

Attempt a connection from the TOE to the server by setting the basicConstraints field in an issuing certificate to have CA=false

- The evaluator ensured that the Self signed CA certificate is present in the TOE's trust store (system keyvault).
- The evaluator exported the ICA2.crt file from the entire certificate chain that was created using XCA tool.
- The evaluator used the acumen x509-mod tool to modify the original ICA2.crt certificate file and output a modified ICA2_fbc.crt certificate file with BasicConstrants field set to false as per the test requirement. The evaluator then verified that the modified certificate has the correct subject and was signed by the correct certificate authority that created using XCA tool.
- The evaluator viewed the modified certificate ICA2_fbc.crt
- The evaluator created a single PEM encoded file with ICA1 and ICA2_fbc that can be presented to the client for certificate path validation.
- The evaluator verified that the pem encoded certificate file created previously have the BasicConstraints field set to false.

- The evaluator ensured that the server certificate 10.1.3.51.crt and the server key 10.1.3.51_key.pem is present in the mysql directory of the database server.
- The evaluator configured the MySQL database server to leverage the uploaded server certificates and the server key for the TLS handshake with the client.
- The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the client could not connect to the server as the server returned a certificate_unknown error to the client.
- The evaluator observed the packet capture on the server and confirmed that the TLS handshake was not successful. Note that the ICA2 certificate has the basicConstraints in the extension field set to false.
- The evaluator further observed the debug logs on the TOE at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs to confirm that the TOE was unable to construct a valid chain and the certification path could not be validated as the certificate presented was not a CA certificate.

Attempt a connection from the TOE to the server by omitting a CA signing bit of the key usage field

- The evaluator used the XCA tool to ensure that one of the issuing certificates (ICA2_Nosigbit) in the certificate path does not have a CA signing bit in the key usage field by transforming the original ICA2 issuing certificate to ICA2_Nosigbit certificate.
- The evaluator ensured that the Self signed CA certificate is present in the TOE's trust store (system keyvault).
- The evaluator uploaded the certificate encoded in pem chain format containing 10.1.3.51.crt, ICA1.crt and ICA2_Nosigbit.crt to the mysql directory of the database server.
- The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the client could not connect to the server as the server returned a certificate_unknown error to the client.
- The evaluator used the acumen-tlsc-mysql tool as a TLS server waiting for connection on IP address 10.1.3.51 and port 3307 and observed a fatal alert : certificate_unknown error when the client attempted a TLS connection.
- The evaluator observed the packet capture on the server and confirmed that the TLS handshake was not successful. Note that the ICA2 certificate has the Certificate signing bit set to false.
- The evaluator further observed the debug logs on the TOE at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs to confirm that the TOE was unable to construct a valid chain and the certification path could not be validated as the issuer certificate keyusage extension is critical and does not permit key signing.

Attempt a connection from the TOE to the server by setting the path length field of a valid CA field to a value strictly lesser than the certificate path

- Set the pathlength field of RootCA certificate to 1
- Set pathlength field of IntCA1 certificate and IntCA2 certificate to zero

- The evaluator used the XCA tool to ensure that one of the issuing certificates (ICA1) in the certificate path has the path length field set to a value 0 that is strictly lesser than the certificate path. i.e., a CA with a path length constraint of zero cannot have any subordinate CAs. However, the ICA1 has a subordinate ICA2 while the path length is set to 0.
- The evaluator ensured that the Self signed CA certificate is present in the TOE's trust store (system keyvault).
- The evaluator uploaded the ICA1_pathlen.crt to the server and ensured that the path length is set 0.
- The evaluator uploaded a pem encoded chain consisting of ICA1_pathlen.crt and ICA2.crt to the mysql directory of the database server.
- The evaluator ensured that the server certificate 10.1.3.51.crt and the server key 10.1.3.51_key.pem is present in the mysql directory of the database server.
- The evaluator configured the MySQL database server to leverage the uploaded server certificates and the server key for the TLS handshake with the client.
- The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the client could not connect to the server as the server returned a certificate_unknown error to the client.
- The evaluator observed the packet capture on the server and confirmed that the TLS handshake was not successful. Note that the ICA1 certificate has the path length set to 0.
- The evaluator further observed the debug logs on the TOE at C:\ProgramFiles\HelpSystems\GoAnywhere\userdata\logs to confirm that the TOE was unable to construct a valid chain and the certification path could not be validated as the max path length is not greater than zero.

The evaluator shall then establish a valid certificate path consisting of valid CA certificates and demonstrate that the function succeeds.

Note: The following test performed satisifies the FCS_TLSC_EXT.1.3 Test #1 requirement where the evaluator demonstrated that a server using a certificate with a valid certification path establishes a successful TLS handshake.

- The evaluator created a chain of four certificates using the XCA tool:
- The node certificate to be tested : 10.1.3.51 is the database server certificate loaded to the TLS server
- Two Intermediate Cas : ICA1 and ICA2 are the intermediate certificate authorities which are loaded to the TLS server. The ICA2.pem certificate file consists of both the ICA1 and ICA2 certificates.
- The self-signed Root CA : CA is the self-signed CA certificate which is loaded to the TOE or the TLS client to the database server.
- The evaluator ensured that the Self signed CA certificate is present in the TOE's trust store (system keyvault).
- The evaluator uploaded the certificate in pem chain format containing both the ICA1 and ICA2 to the mysql directory of the database server.
- The evaluator ensured that the server certificate 10.1.3.51.crt and the server key 10.1.3.51_key.pem is present in the mysql directory of the database server.

	• The evaluator configured the MySQL database server to leverage the uploaded server certificates and the server key for the TLS handshake with the client. • The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the TLS connection successfully established as the resource test was successful. • The evaluator observed the packet capture on the server and confirmed that the TLS handshake was successful with server presenting a certificate with a valid certification path. Attempt a connection from the TOE to the server by removing trust in one of the CA. **Note: The following test performed satisifies the FCS_TLSC_EXT.1.3 Test #1c requirement where the evaluator demonstrated that modifying the trust store element to be untrusted (by deleting the Intermediate certificate from the trust store) and attempting a connection from the server resulted in an authentication failure as the TOE was unable to construct a valid certificate path reason being one of the issuing certificates in the certificate path is not present in the TOE's trust store.** • The evaluator then removed the trust ICA1 from the pem encoded chain and presented only the ICA2 certificate that signed the server certificate from the server side. • The evaluator ensured that only the Self signed CA certificate is present in the TOE's trust store (system keyvault) and modified the trust store to confirm that the ICA1 was not present in the trust store which makes it untrusted. • The evaluator configured the MySQL database server to leverage the uploaded server certificate, server key and present only the ICA2 certificate in the capath for the TLS handshake with the client. • The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the client could not connect to the server as the server returned a certificate_unknown error to the client. • The evaluator observed the packet capture on the server and confirmed that the TLS handshake was not successful. Note that the ICA1 certificate was not presented by the server in the certificate path. • The evaluator further observed the debug logs on the TOE at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs to confirm that the TOE was unable to construct a valid chain as no issuer certificate in the certification path was found.
Pass/Fail with Explanation	Pass. The TOE will not validate a certificate without a valid certification path but it will accept that same certificate when it has the valid Certificate chain. This meets the testing requirements.

6.18.2 FIA_X509_EXT.1.1 Test #2

Item	Data
Test Assurance Activity	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules.

	If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: - The node certificate to be tested, - Two Intermediate CAs, and - The self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. The evaluator shall demonstrate that validating an expired certificate results in the function failing. **TD0668 has been applied.**
Test Steps	**Note: The following test performed satisifies the FCS_TLSC_EXT.1.3 Test #3 requirement where the evaluator demonstrated that server using a certificate which has passed its expiration date results in an authentication failure which was confirmed through packet capture where the client could not connect to the server as the server returned a certificate_unknown error to the client and also confirmed through TOE logs that the certificate expired on 20210212050000GMT+00:00. which corresponds to the server certificate.** • The evaluator created a chain of four certificates using the XCA tool. • The evaluator used the XCA tool to create an expired certificate that expired on February 12, 2021, 12:00:00 AM EST. • The evaluator ensured that the Self signed CA certificate is present in the TOE's trust store (system keyvault). • The evaluator uploaded the certificate in pem chain format containing both the ICA1 and ICA2 to the mysql directory of the database server. • The evaluator ensured that the server certificate expired on February 12, 2021, 12:00:00 AM EST and uploaded the expired certificate to the mysql directory of the database server. • The evaluator checked the current date and time on the database server and ensured that the certificate expired as per the current time. • The evaluator checked the current date and time on the TOE Platform and ensured that the certificate expired as per the current time. • The evaluator configured the MySQL database server to leverage the uploaded expired server certificate, ICA2.pem and the server key for the TLS handshake with the client. • The evaluator configured the TOE to reach the Database server at Resources > Database Servers where the JDBC Driver was set to mariadb.jdbc .driver as the server was a MariaDB SQL server and provided the necessary URL information to reach the database server along with username and password of the database created on the server. **WADATA** is the database created on the MySQL server for the TOE on the Windows Platform. • The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the client could not connect to the server as the server returned a certificate_unknown error to the client.

	• The evaluator observed the packet capture on the server and confirmed that the TLS handshake was not successful. • The evaluator further observed the debug logs on the TOE at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs to confirm that the certificate expired on 20210212050000GMT+00:00.
Pass/Fail with Explanation	Pass. The TOE does not validate an expired certificate and the TLS connection failed. This meets the testing requirements.

6.18.3 FIA_X509_EXT.1.1 Test #3

Item	Data
Test Assurance Activity	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: - The node certificate to be tested, - Two Intermediate CAs, and - The self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. The evaluator shall test that the TOE can properly handle revoked certificates – conditional on whether CRL, OCSP, or OCSP Stapling or OCSP Multi-stapling is selected; if multiple methods are selected, then the following tests shall be performed for each method: o The evaluator shall test revocation of the node certificate. o The evaluator shall also test revocation of an intermediate CA certificate (i.e. the intermediate CA certificate should be revoked by the root CA), if intermediate CA certificates are supported. If OCSP stapling per RFC6066 is the only supported revocation method, this test is omitted. The evaluator shall ensure that a valid certificate is used, and that the validation function succeeds. The evaluator then attempts the test with a certificate that has been revoked (for each method chosen in the selection) to ensure when the certificate is no longer valid that the validation function fails. **TD0668 has been applied.**
Test Steps	**Note: The following test performed satisifies the FCS_TLSC_EXT.1.3 Test #2 and FIA_X509_EXT.2.2 Test #2 requirement where the evaluator demonstrated that the server using a certificate which has been revoked results in an authentication failure which was confirmed through packet capture where the client could not connect to the server as the server returned a certificate_unknown error to the client and also confirmed through TOE logs that the Certificate 82:F7:34:04:5D:C4:24:C3:64:0E:A1:16:2E:16:2B:04:0D:32:CB:C2 has been revoked by CRL at 'http://10.1.3.51/ICA2.crl' which corresponds to the server certificate.**

	• The evaluator shall test revocation of the node certificate. • The evaluator created a chain of four certificates using the XCA tool: • The node certificate to be tested : 10.1.3.51 is the database server certificate loaded to the TLS server • Two Intermediate Cas : ICA1 and ICA2 are the intermediate certificate authorities which are loaded to the TLS server. The ICA2.pem certificate file consists of both the ICA1 and ICA2 certificates. • The self-signed Root CA : CA is the self-signed CA certificate which is loaded to the TOE's trust store. • The evaluator used the XCA tool and revoked the server certificate. • The evaluator generated CRLs using the xca tool. • The evaluator uploaded the CRLs in the html directory of the web server and ensured that there are no revoked certificates in CA.crl, ICA1.crl and have the server certificate as revoked in ICA2.crl • The evaluator ensured that the Self signed CA certificate is present in the TOE's trust store (system keyvault). • The evaluator uploaded the certificate in pem chain format containing both the ICA1 and ICA2 to the mysql directory of the database server. • The evaluator uploaded the revoked server certificate 10.1.3.51.crt and the server key 10.1.3.51_key.pem in the mysql directory of the database server. • The evaluator configured the MySQL database server to leverage the uploaded server certificates and the server key for the TLS handshake with the client. • The evaluator ensured to have the CRL check enabled for server certificates and specified the web server's URL information to fetch all the CRLs from the web server. • The evaluator configured the TOE to reach the Database server at Resources > Database Servers where the JDBC Driver was set to mariadb.jdbc .driver as the server was a MariaDB SQL server and provided the necessary URL information to reach the database server along with username and password of the database created on the server. WADATA is the database created on the MySQL server for the TOE on the Windows Platform. • The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the client could not connect to the server as the server returned a certificate_unknown error to the client. • The evaluator observed the packet capture on the server and verified that the client fetched the CRLs required to validate the certificates and ensured to return a fatal alert to the server. • The evaluator further observed the debug logs on the TOE at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs to confirm that the Certificate 82:F7:34:04:5D:C4:24:C3:64:0E:A1:16:2E:16:2B:04:0D:32:CB:C2 has been revoked by CRL at 'http://10.1.3.51/ICA2.crl' which corresponds to the server certificate. • The evaluator used the XCA tool and unrevoked the server certificate. The evaluator then revoked the ICA2 certificate that was signed by its root certificate authority ICA1.

	• The evaluator generated CRLs using the xca tool. • The evaluator uploaded the CRLs in the html directory of the web server and ensured that there are no revoked certificates in CA.crl, ICA2.crl and have the ICA2 certificate as revoked in ICA1.crl. • The evaluator restarted the apache2 webserver to ensure the TOE fetches the updated CRLs. • The evaluator configured the MySQL database server to leverage the uploaded server certificates and the server key for the TLS handshake with the client. • The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the client could not connect to the server as the server returned a certificate_unknown error to the client. • The evaluator observed the packet capture on the server and verified that the client fetched the CRLs required to validate the certificates and ensured to return a fatal alert to the server. • The evaluator further observed the debug logs on the TOE at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs to confirm that the Certificate 85:0D:CD:92:6E:04:CA:1E:43:D9:D1:76:41:61:A6:B6:30:61:BF:74 has been revoked by CRL at 'http://10.1.3.51/ICA1.crl' which corresponds to the ICA2 certificate that was signed by its root authority ICA1. • The evaluator shall ensure that a valid certificate is used, and that the validation function succeeds. • The evaluator used the XCA tool and unrevoked the ICA2 certificate and ensured that there are no revoked certificates in the chain. • The evaluator generated CRLs using the xca tool. • The evaluator uploaded the CRLs in the html directory of the web server and ensured that there are no revoked certificates in CA.crl, ICA1.crl, ICA2.crl. • The evaluator restarted the apache2 webserver to ensure the TOE fetches the updated CRLs. • The evaluator configured the MySQL database server to leverage the uploaded server certificates and the server key for the TLS handshake with the client. • The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the client was able to successfully communicate with the database server. • The evaluator observed the packet capture on the server and verified that the client fetched the CRLs required to validate the certificates and ensured that the TLS handshake was successful.
Pass/Fail with Explanation	Pass. The evaluator ensured when the node certificate or intermediate CA certificate is revoked and no longer valid, then the TLS handshake fails, and the validation function fails. This meetings the testing requirements.

6.18.4 FIA_X509_EXT.1.1 Test #4

Item	Data
Test Assurance Activity	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1.

	The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: - The node certificate to be tested, - Two Intermediate CAs, and - The self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. Test 4: If any OCSP option is selected, the evaluator shall **ensure the TSF has no other source of revocation information available and** configure the OCSP server or use a man-in-the-middle tool to present **an OCSP response signed by** a certificate that does not have the OCSP signing purpose and **which is the only source of revocation status information advertised by the CA issuing the certificate being validated. The evaluator shall** verify that validation of the OCSP response fails **and that the TOE treats the certificate being checked as invalid and rejects the connection**. If CRL is selected, the evaluator shall **likewise** configure the CA **to be the only source of revocation status information, and** to sign a CRL with a certificate that does not have the cRLsign key usage bit set. **The evaluator shall** verify that validation of the CRL fails **and that the TOE treats the certificate being checked as invalid and rejects the connection**. ***Note: The intent of this test is to ensure a TSF does not trust invalid revocation status information. A TSF receiving invalid revocation status information from the only advertised certificate status provider should treat the certificate whose status is being checked as invalid. This should generally be treated differently from the case where the TSF is not able to establish a connection to check revocation status information, but it is acceptable that the TSF ignore any invalid information and attempt to find another source of revocation status (another advertised provider, a locally configured provider, or cached information) and treat this situation as not having a connection to a valid certificate status provider.*** **TD0668 has been applied.**
Test Steps	• The evaluator created a chain of four certificates using the XCA tool: The node certificate to be tested : 10.1.3.51 is the database server certificate loaded to the TLS server Two Intermediate Cas : ICA1 and ICA2_noCRLsig are the intermediate certificate authorities which are loaded to the TLS server. The ICA2_noCRLsig.pem certificate file consists of both the ICA1 and ICA2_noCRLsig certificates. The ICA2_noCRLsig certificate does not have the cRLsign key usage bit set. The self-signed Root CA : CA is the self-signed CA certificate which is loaded to the TOE or the TLS client to the database server. • The evaluator generated the CRLs using the XCA tool and ensured that the ICA2.crl was signed by the ICA2_noCRLsig certificate that does not have the cRLsign key usage bit set. • The evaluator uploaded the CRLs in the html directory of the web server and ensured that there are no revoked certificates in CA.crl, ICA1.crl, ICA2.crl.

	• The evaluator ensured that the Self signed CA certificate is present in the TOE's trust store (system keyvault). • The evaluator uploaded the certificate in pem chain format containing both the ICA1 and ICA2_noCRLsig to the mysql directory of the database server. The evaluator ensured the certificate does not have the cRLsign key usage bit set. • The evaluator uploaded the revoked server certificate 10.1.3.51.crt and the server key 10.1.3.51_key.pem in the mysql directory of the database server. • The evaluator configured the MySQL database server to leverage the uploaded server certificate and the server key for the TLS handshake with the client and specified the ICA2_noCRLsig.pem chain as the CA. • The evaluator ensured to have the CRL check enabled for server certificates and specified the web server's URL information to fetch all the CRLs from the web server. • The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the client could not connect to the server as the server returned a certificate_unknown error to the client. NOTE: When CRL check is enabled and the certificate signing the CRL does not have a crlSign bit enabled, the TOE fails to validate the CRL, and the TLS connection fails. • The evaluator observed the packet capture on the server and ensured verified that the client fetched the CRLs required to validate the certificates, but the client ensured to return a fatal alert to the server as the TOE fails to validate the CRL that was signed by the certificate which does not have the crlSign bit enabled. • The evaluator further observed the debug logs on the TOE at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs to confirm that the TLS handshake was not successful as the certificate's CA does not contain the crlSign bit.
Pass/Fail with Explanation	Pass. The TOE fails to validate the CRL when the certificate used to sign the CRL is missing the CRL signing purpose in the Key Usage. This meets the testing requirements.

6.18.5 FIA_X509_EXT.1.1 Test #5

Item	Data
Test Assurance Activity	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: - The node certificate to be tested, - Two Intermediate CAs, and - The self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created.

	The evaluator shall modify any byte in the first eight bytes of the certificate and demonstrate that the certificate fails to validate. (The certificate will fail to parse correctly.) **TD0668 has been applied.**
Test Steps	• The evaluator created a chain of four certificates using the XCA tool: The node certificate to be tested : 10.1.3.51 is the database server certificate presented during the TLS handshake. Two Intermediate Cas : ICA1 and ICA2 are the intermediate certificate authorities. The 10.1.3.51.pem certificate file consists of the ICA1, ICA2 and 10.1.3.51 certificates in pem encoded chain format. The self-signed Root CA : CA is the self-signed CA certificate which is loaded to the TOE's trust store. • The evaluator ensured that the Self signed CA certificate is present in the TOE's trust store (system keyvault). • The evaluator uploaded the server certificate which is encoded in pem chain format along with ICA1 and ICA2 certificates and the server key 10.1.3.51_key.pem in the mysql directory of the database server. • The evaluator used the "acumen-tlsc-mysql" tool as a server waiting for TLS connections on IP address 10.1.3.51 and port 3307 presenting the certificate chain 10.1.3.51.pem as the server certificate and the server key 10.1.3.51_key.pem with ID 15 that corresponds to the current test. The evaluator observed the tool output and ensured that a fatal alert was returned to the server after the server presented a certificate with first 8 bytes modified. • The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the client could not connect to the server as the server returned a certificate_unknown error to the client. • The evaluator observed the packet capture on the server and ensured that the server presented a certificate with the first 8 modified while the client returned a fatal alert to the server.
Pass/Fail with Explanation	Pass. The evaluator modified the first eight bytes of the certificate being presented by the server and ensured that the certificate fails to validate, and the TLS handshake fails. This meets the testing requirements.

6.18.6 FIA_X509_EXT.1.1 Test #6

Item	Data
Test Assurance Activity	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: - The node certificate to be tested, - Two Intermediate CAs, and - The self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created.

	The evaluator shall modify any byte in the last byte of the certificate and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.) **TD0668 has been applied.**
Test Steps	• The evaluator created a chain of four certificates using the XCA tool: The node certificate to be tested : 10.1.3.51 is the database server certificate presented during the TLS handshake. Two Intermediate Cas : ICA1 and ICA2 are the intermediate certificate authorities. The 10.1.3.51.pem certificate file consists of the ICA1, ICA2 and 10.1.3.51 certificates in pem encoded chain format. The self-signed Root CA : CA is the self-signed CA certificate which is loaded to the TOE's trust store. • The self-signed Root CA : CA is the self-signed CA certificate which is loaded to the TOE or the TLS client to the database server. • The evaluator ensured that the Self signed CA certificate is present in the TOE's trust store (system keyvault). • The evaluator uploaded the server certificate 10.1.3.51.pem which is encoded in pem chain format along with ICA1 and ICA2 certificates and the server key 10.1.3.51_key.pem in the mysql directory of the database server. • The evaluator used the "acumen-tlsc-mysql" tool as a server waiting for TLS connections on IP address 10.1.3.51 and port 3307 presenting the certificate chain 10.1.3.51.pem as the server certificate and the server key 10.1.3.51_key.pem with ID 16 that corresponds to the current test. The evaluator observed the tool output and ensured that a fatal alert was returned to the server after the server presented a certificate with the last byte modified. • The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the client could not connect to the server as the server returned a certificate_unknown error to the client. • The evaluator observed the packet capture on the server and ensured that the server presented a certificate with the last byte modified while the client returned a fatal alert to the server.
Pass/Fail with Explanation	Pass. The evaluator modified the last byte of the certificate and demonstrated that the certificate fails to validate. This meets the testing requirements.

6.18.7 FIA_X509_EXT.1.1 Test #7

Item	Data
Test Assurance Activity	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: - The node certificate to be tested, - Two Intermediate CAs, and

	- The self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. The evaluator shall modify any byte in the public key of the certificate and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.) **TD0668 has been applied.**
Test Steps	• The evaluator created a chain of four certificates using the XCA tool: The node certificate to be tested : 10.1.3.51 is the database server certificate presented during the TLS handshake. Two Intermediate Cas : ICA1 and ICA2 are the intermediate certificate authorities. The 10.1.3.51.pem certificate file consists of the ICA1, ICA2 and 10.1.3.51 certificates in pem encoded chain format. • The self-signed Root CA : CA is the self-signed CA certificate which is loaded to the TOE's trust store. • The self-signed Root CA : CA is the self-signed CA certificate which is loaded to the TOE or the TLS client to the database server. • The evaluator ensured that the Self signed CA certificate is present in the TOE's trust store (system keyvault). • The evaluator uploaded the server certificate 10.1.3.51.pem which is encoded in pem chain format along with ICA1 and ICA2 certificates and the server key 10.1.3.51_key.pem in the mysql directory of the database server. • The evaluator used the "acumen-tlsc-mysql" tool as a server waiting for TLS connections on IP address 10.1.3.51 and port 3307 presenting the certificate chain 10.1.3.51.pem as the server certificate and the server key 10.1.3.51_key.pem with ID 17 that corresponds to the current test. The evaluator observed the tool output and ensured that a fatal alert was returned to the server after the server presented a certificate with a modified public key. • The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the client could not connect to the server as the server returned a certificate_unknown error to the client. • The evaluator observed the packet capture on the server and ensured that the server presented a certificate with the public key modified while the client returned a fatal alert to the server.
Pass/Fail with Explanation	Pass. The evaluator modified 8 bytes in the public key of the server certificate and demonstrated that the certificate fails to validate. This meets the testing requirements.

6.18.8 FIA_X509_EXT.1.1 Test #8a

Item	Data
Test Assurance Activity	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates:

	- The node certificate to be tested, - Two Intermediate CAs, and - The self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. **(Conditional on support for EC certificates as indicated in FCS_COP.1(3)).** The evaluator shall establish a valid, trusted certificate chain consisting of an EC leaf certificate, an EC Intermediate CA certificate not designated as a trust anchor, and an EC certificate designated as a trusted anchor, where the elliptic curve parameters are specified as a named curve. The evaluator shall confirm that the TOE validates the certificate chain. **TD0668 has been applied.**
Test Steps	• The evaluator established a valid, trusted certificate chain consisting of an EC leaf server certificate (10.1.3.51_ec), an EC Intermediate CA certificate (ICA_ec) not designated as a trust anchor, and an EC certificate (CA_ec) designated as a trusted anchor, where the elliptic curve parameters are specified as a named curve using the XCA tool. • The evaluator ensured that the CA_ec certificate was configured with the EC elliptic curve parameters specified as a named curve. • The evaluator uploaded the ICA_ec certificate to the mysql directory of the database server and ensured that it was configured with the EC elliptic curve parameters specified as a named curve. • The evaluator uploaded the 10.1.3.51_ec.crt certificate to the mysql directory of the database server and ensured that it was configured with the EC elliptic curve parameters specified as a named curve. • The evaluator ensured that the Self signed CA_ec certificate is present in the TOE's trust store (system keyvault). • The evaluator configured the MySQL database server to leverage the uploaded server certificate, ICA_ec and the ec server key for the TLS handshake with the client. • The evaluator configured the TOE to reach the Database server at Resources > Database Servers where the JDBC Driver was set to mariadb.jdbc .driver as the server was a MariaDB SQL server and provided the necessary URL information to reach the database server along with username and password of the database created on the server. WADATA is the database created on the MySQL server for the TOE on the Windows Platform. • The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the client successfully communicated with the data base server with the EC certificates configured. • The evaluator observed the packet capture and confirmed that the TOE validates the EC certificate chain where the elliptic curve parameters are specified as a named curve.
Pass/Fail with Explanation	Pass. The TOE validates an EC certificate chain where the EC elliptic curve parameters are specified as a Named curve.

6.18.9 FIA_X509_EXT.1.1 Test #8b

Item	Data
Test Assurance Activity	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: - The node certificate to be tested, - Two Intermediate CAs, and - The self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. **(Conditional on support for EC certificates as indicated in FCS_COP.1(3)).** The evaluator shall replace the intermediate certificate in the certificate chain for Test 8a with a modified certificate, where the modified intermediate CA has a public key information field where the EC parameters uses an explicit format version of the Elliptic Curve parameters in the public key information field of the intermediate CA certificate from Test 8a, and the modified Intermediate CA certificate is signed by the trusted EC root CA, but having no other changes. The evaluator shall confirm the TOE treats the certificate as invalid. **TD0668 has been applied.**
Test Steps	• The evaluator established a valid, trusted certificate chain consisting of an EC leaf server certificate (10.1.3.51_ec), an EC Intermediate CA certificate (ICA_ec) not designated as a trust anchor, and an EC certificate (CA_ec) designated as a trusted anchor, where the elliptic curve parameters are specified as a named curve using the XCA tool. • The evaluator ensured that the CA_ec certificate was configured with the EC elliptic curve parameters specified as a named curve. • The evaluator uploaded the ICA_ec certificate to the mysql directory of the database server and ensured that it was configured with the EC elliptic curve parameters specified as a named curve. • The evaluator uploaded the 10.1.3.51_ec.crt certificate to the mysql directory of the database server and ensured that it was configured with the EC elliptic curve parameters specified as a named curve. • The evaluator ensured that the Self signed CA_ec certificate is present in the TOE's trust store (system keyvault). • The evaluator used the acumen x509-mod tool to modify the original ICA_ec.crt certificate file where the elliptic curve parameters are specified as a named curve and output a modified ICA_ec_mod.crt certificate file that has a public key information field where the EC parameters uses an explicit format version of the Elliptic Curve parameters in the public key information field as per the test requirement. The evaluator then verified that the modified certificate has the correct subject and was signed by the correct certificate authority that created using XCA tool.

	• The evaluator ensured the modified Intermediate CA certificate uses an explicit format version of the Elliptic Curve parameters in the public key information field. • The evaluator configured the MySQL database server to leverage the uploaded server certificate, ICA_ec_mod and the ec server key for the TLS handshake with the client. • The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the client could not connect to the server as the server returned a certificate_unknown error to the client. • The evaluator observed the packet capture on the server and confirmed that the TLS handshake was not successful. • The evaluator further observed the debug logs on the TOE at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs to confirm that only named elliptic curves are allowed by the TOE while the certificate for 'ICA' contains an implicit or specified curve.
Pass/Fail with Explanation	Pass. The evaluator confirmed that the TOE treats the intermediate CA that has a public key information field where the EC parameters uses an explicit format version of the Elliptic Curve parameters in the public key information field as invalid. This meets the testing requirements.

6.18.10 FIA_X509_EXT.1.2 Test #1

Item	Data
Test Assurance Activity	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: - The node certificate to be tested, - Two Intermediate CAs, and - The self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. The evaluator shall ensure that the certificate of at least one of the CAs in the chain does not contain the basicConstraints extension. The evaluator shall confirm that validation of the certificate path fails: (i) as part of the validation of the peer certificate belonging to this chain; and/or (ii) when attempting to add the CA certificate without the basicConstraints extension to the TOE's trust store. **TD0495 has been applied.**
Test Steps	• The evaluator created a chain of four certificates using the XCA tool. • The evaluator used the XCA tool to ensure that one of the issuing certificates (ICA2_nbc) in the certificate path does not have the basicConstraints by

	transforming the original ICA2 issuing certificate to a ICA_nbc omitting the basicConstraints field. • The evaluator ensured that the Self signed CA certificate is present in the TOE's trust store (system keyvault). • The evaluator uploaded the certificate in pem chain format containing both the ICA1 and ICA2_nbc to the mysql directory of the database server. • The evaluator ensured that the certificate in pem chain format contains both ICA1 and ICA2_nbc. • The evaluator ensured that the server certificate 10.1.3.51.crt and the server key 10.1.3.51_key.pem is present in the mysql directory of the database server. • The evaluator configured the MySQL database server to leverage the uploaded server certificates and the server key for the TLS handshake with the client. • The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the client could not connect to the server as the server returned a certificate_unknown error to the client. • The evaluator observed the packet capture on the server and confirmed that the TLS handshake was not successful. Note that the ICA2 certificate does not contain basicConstraints in the extension field. • The evaluator further observed the debug logs on the TOE at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs to confirm that the TOE was unable to construct a valid chain and the certification path could not be validated as the Intermediate certificate lacks basic constraints.
Pass/Fail with Explanation	Pass. The TOE fails to validate a certificate with no basicConstraints and rejects it. This meets the testing requirements.

6.18.11 FIA_X509_EXT.1.2 Test #2

Item	Data
Test Assurance Activity	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: - The node certificate to be tested, - Two Intermediate CAs, and - The self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. The evaluator shall ensure that the certificate of at least one of the CAs in the chain has the CA flag in the basicConstraints extension not set (or set to FALSE). The evaluator shall confirm that validation of the certificate path fails (i) as part of the validation of the peer certificate belonging to this chain; and/or

	(ii) when attempting to add the CA certificate with the CA flag not set (or set to FALSE) in the basicConstraints extension to the TOE's trust store **TD0495 has been applied.**
Test Steps	• The evaluator created a chain of four certificates using the XCA tool. • The evaluator ensured that the Self signed CA certificate is present in the TOE's trust store (system keyvault). • The evaluator exported the ICA2.crt file from the entire certificate chain that was created using XCA tool. • The evaluator used the acumen x509-mod tool to modify the original ICA2.crt certificate file and output a modified ICA2_fbc.crt certificate file with BasicConstrants field set to false as per the test requirement. The evaluator then verified that the modified certificate has the correct subject and was signed by the correct certificate authority that created using XCA tool. • The evaluator viewed the modified certificate ICA2_fbc.crt • The evaluator created a single PEM encoded file with ICA1 and ICA2_fbc that can be presented to the client for certificate path validation. • The evaluator verified that the pem encoded certificate file created previously have the BasicConstraints field set to false. • The evaluator ensured that the server certificate 10.1.3.51.crt and the server key 10.1.3.51_key.pem is present in the mysql directory of the database server. • The evaluator configured the MySQL database server to leverage the uploaded server certificates and the server key for the TLS handshake with the client. • The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the client could not connect to the server as the server returned a certificate_unknown error to the client. • The evaluator observed the packet capture on the server and confirmed that the TLS handshake was not successful. Note that the ICA2 certificate has the basicConstraints in the extension field set to false. • The evaluator further observed the debug logs on the TOE at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs to confirm that the TOE was unable to construct a valid chain and the certification path could not be validated as the certificate presented was not a CA certificate.
Pass/Fail with Explanation	Pass. CA Certificates with the basicConstraints flag set to false are rejected by the TOE. This meets the testing requirements.

6.18.12 FIA_X509_EXT.1.2 Test #3

TD0495 removes this test.

6.18.13 FIA_X509_EXT.2.2 Test #1

Item	Data
Test Assurance Activity	The evaluator shall perform the following test for each trusted channel: **Test 1:** The evaluator shall demonstrate that using a valid certificate that requires certificate validation checking to be performed in at least some part by communicating with a non-TOE IT entity.

	The evaluator shall then manipulate the environment so that the TOE is unable to verify the validity of the certificate and observe that the action selected in FIA_X509_EXT.2.2 is performed. If the selected action is administrator-configurable, then the evaluator shall follow the operational guidance to determine that all supported administrator-configurable options behave in their documented manner.
Test Steps	**TOE as client:** • The evaluator created a chain of four certificates using the XCA tool. • The evaluator uploaded the CRLs in the html directory of the web server and ensured that there are no revoked certificates in CA.crl, ICA1.crl, ICA2.crl. • The evaluator ensured that the Self signed CA certificate is present in the TOE's trust store (system keyvault). • The evaluator uploaded the certificate in pem chain format containing both the ICA1 and ICA2 to the mysql directory of the database server. • The evaluator ensured that the server certificate 10.1.3.51.crt and the server key 10.1.3.51_key.pem is present in the mysql directory of the database server. • The evaluator ensured to have the CRL check enabled for server certificates and specified the web server's URL information to fetch all the CRLs from the web server. • The evaluator configured the TOE to reach the Database server at Resources > Database Servers where the JDBC Driver was set to mariadb.jdbc .driver as the server was a MariaDB SQL server and provided the necessary URL information to reach the database server along with username and password of the database created on the server. WADATA is the database created on the MySQL server for the TOE on the Windows Platform. • The evaluator configured the MySQL database server to leverage the uploaded server certificates and the server key for the TLS handshake with the client. • The evaluator attempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the TLS connection successfully established as the resource test was successful. • The evaluator observed the packet capture on the server and verified that the client fetched the CRLs required to validate the certificates and ensured that the TLS handshake was successful. • The evaluator then manipulated the environment by shutting down the apache2 webserver on the non-TOE IT entity which the TOE is communicating with to verify the validity of the certificate. • The evaluator reattempted a connection from the TOE (TLS client) to the MySQL server using the TOE's resources that was previously created and confirmed that the TLS connection did not establish as the client returned a fatal alert "certificate unknown" to the server. • The evaluator observed the packet capture on the server and verified that the client was unable to fetch the CRLs required to validate the certificates and confirmed that the TLS connection did not establish. • The evaluator further observed the debug logs on the TOE at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs to confirm the error was that

	the client was unable to fetch the Certificate Revocation List from URL 'http://10.1.3.51/CA.crl. **TOE as server:** • The evaluator used the XCA tool to create the required certificates. • The evaluator generated CRLs using the XCA tool. • The evaluator uploaded the CRLs in the html directory of the web server and ensured that there are no revoked certificates in CA.crl, ICA1_client.crl and have the server certificate as revoked in ICA2_client.crl • The evaluator ensured that the Self signed CA certificate is present in the TOE's trust store (system keyvault). • The evaluator uploaded the certificate in pem chain format containing both the ICA1_client and ICA2_client to the client VM. • The evaluator uploaded the server certificate 10.1.3.51_client.pem and the server key 10.1.3.51_client_key.pem in the Client VM. • The evaluator ensured to have the CRL check enabled for client certificates and specified the web server's URL information to fetch all the CRLs from the web server. • The evaluator ensured that the CA and ICA certificates that signed the server_x509 certificate are uploaded to the TOE' trust store. • The evaluator configured the TOE to use the server_x509 certificate for TLS connection used for remote administration. The evaluator also ensured that the client authentication is required. • The evaluator attempted a connection from the VM (TLS client) to the remote Admin server using the openssl s_client and confirmed that the client could connect to the server. • The evaluator observed the packet capture on the server and verified that the TOE fetched the CRLs required to validate the client certificates and ensured to establish a connection with the TOE. • The evaluator then manipulated the environment by shutting down the apache2 webserver on the non-TOE IT entity which the TOE is communicating with to verify the validity of the certificate. • The evaluator attempted a connection from the VM (TLS client) to the remote Admin server using the openssl s_client and confirmed that the client could not connect to the server. • The evaluator observed the packet capture on the client and verified that the server was unable to fetch the CRLs required to validate the certificates and confirmed that the TLS connection did not establish. • The evaluator further observed the debug logs on the TOE at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs to confirm the error was that the client was unable to fetch the Certificate Revocation List from URL 'http://10.1.3.51/CA.crl.
Pass/Fail with Explanation	Pass. The evaluator demonstrated that certificate validation checking was performed by communicating with a non-TOE IT entity during the TLS handshake. The evaluator then manipulated the environment so that the TOE is unable to verify the validity of the certificate and observed that when the client could not establish a connection to

	determine the validity of a certificate, the client did not accept the certificate. This meets the testing requirements.

6.18.14 FIA_X509_EXT.2.2 Test #2

Item	Data
Test Assurance Activity	The evaluator shall perform the following test for each trusted channel: **Test 2:** The evaluator shall demonstrate that an invalid certificate that requires certificate validation checking to be performed in at least some part by communicating with a non-TOE IT entity cannot be accepted.
Expected Results	As a part of FIA_X509_EXT.1.1 Test #3, • The evaluator observed the packet capture on the server and verified that the client fetched the CRLs required to validate the server certificate by communicating with a non-TOE IT entity (CRL web server) and ensured to return a fatal alert to the server as the server certificate was revoked. • The evaluator further observed the debug logs on the TOE at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs to confirm that the Certificate 82:F7:34:04:5D:C4:24:C3:64:0E:A1:16:2E:16:2B:04:0D:32:CB:C2 has been revoked by CRL at 'http://10.1.3.51/ICA2.crl' which corresponds to the server certificate. (Added a Note in FIA_X509_EXT.1.1 Test #3 which satisfied the current requirement)
Pass/Fail with Explanation	Pass. This test is performed in conjunction with FIA_X509_EXT.1.1 Test #3 where the evaluator demonstrated that the validation check of the certificate was performed by communicating with a non-TOE IT entity (CRL web server) and ensured that it cannot be accepted as the certificate was deemed invalid (revoked).

6.18.15 FCS_HTTPS_EXT.1.3 /Client

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that using a certificate without a valid certification path results in the selected action in the SFR. If "notify the user" is selected in the SFR, then the evaluator shall also determine that the user is notified of the certificate validation failure. Using the administrative guidance, the evaluator shall then load a certificate or certificates to the Trust Anchor Database needed to validate the certificate to be used in the function and demonstrate that the function succeeds. The evaluator then shall delete one of the certificates, and show that again, using a certificate without a valid certification path results in the selected action in the SFR, and if "notify the user" was selected in the SFR, the user is notified of the validation failure
Test Steps	• The evaluator created the necessary certificates to perform the test using the XCA tool. • The evaluator created key vault named “https_client” that is required to setup the certificates that must be used by the TOE in Encryption > Key Management Services > Add Key vault. • The evaluator ensured that the key vault was created. • The evaluator uploaded the CA_client certificate to the TOE’S trust store (system keyvault) where the trusted root certificates are stored.

	• The evaluator uploaded the client_windows.pem certificate in pem encoded format into the keyvault that was previously created on the TOE which will be presented as a client certificate during HTTPS/TLS handshake. • The evaluator ensured that the client certificate was installed on the TOE. • The evaluator added the HTTPS server resource where the details regarding the HTTPS server with which client have to communicate is configured. The evaluator specified the server host IP address as 10.1.3.51. • The evaluator set the server port to 443. • The evaluator selected the key vault "https_client" that was previously created and the client_windows certificate that was uploaded on the TOE. • Note: The evaluator observed that Key Management system will be used for validating the HTTPS server's identity. • The evaluator saved the HTTPS server resource created. • The evaluator ensured that the CA_server certificate authority that signed the server certificate was not present in the TOE's trust store (Encryption > Key Management Services > System keyvault) where the trust certificates are stored. • The evaluator created a project in Workflows and configured the project to leverage the HTTPS resource that was previously created. The evaluator added a POST function under the HTTPS resource to send a text file to the HTTPS server. • The evaluator uploaded the server certificate and the key that was created on the Web server that is being used as a HTTPS server. • The evaluator ensured that the server certificate was signed by the CA_Server certificate authority. • The evaluator configured the server to use the uploaded certificate and key for TLS/HTTPS handshake and wait for a TLS connection on IP address 10.1.3.51 and port 443. • The evaluator initiated a TLS connection the send a file to the HTTPS server and ensured that the TLS/HTTPS connection failed due to certificate unknown error returned by the client to the server. • The evaluator observed the logs on the TOE to ensure that the TLS/HTTPS handshake was not successful due to certificate unknown error. • The evaluator further observed the stack trace to ensure that the TLS handshake failed as the TOE could not find the issuer certificate for the server certificate in certificate path. • The evaluator observed the packet capture on the server and ensured that the client returned a fatal alert: Certificate unknown to the server. • The evaluator then uploaded the trusted CA_server certificate that signed the server certificate to the Trust store (system keyvault). • The evaluator attempted the HTTPS/TLS connection to send the text file to the HTTPS server and ensured that the project executed the task with no errors. • The evaluator observed the logs and ensured that the task was successfully executed. • The evaluator observed the packet capture on the server to ensure that the TLS/HTTPS handshake with the HTTPS server was successful.

	• The evaluator then deleted the CA_server certificate authority that signed the server certificate. • The evaluator ensured that the CA_server certificate authority is not present in the TOE's trust store. (system keyvault) • The evaluator initiated a TLS connection the send a file to the HTTPS server and ensured that the TLS/HTTPS connection failed due to certificate unknown error returned by the client to the server. • The evaluator observed the logs on the TOE to ensure that the TLS/HTTPS handshake was not successful due to certificate unknown error. • The evaluator further observed the stack trace to ensure that the TLS handshake failed as the TOE could not find the issuer certificate for the server certificate in certificate path. • The evaluator observed the packet capture on the server and ensured that the client returned a fatal alert: Certificate unknown to the server.
Pass/Fail with Explanation	Pass. The HTTPS connection succeeds only when the TOE can successfully validate the certificate chain. This meets the test requirements.

6.18.16 FCS_HTTPS_EXT.2/HTTPS with Mutual authentication

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that using a certificate without a valid certification path results in the selected action in the SFR. Using the administrative guidance, the evaluator shall then load a certificate or certificates to the Trust Anchor Database needed to validate the certificate to be used in the function and demonstrate that the function succeeds. The evaluator then shall delete one of the certificates, and show that again, using a certificate without a valid certification path results in the selected action in the SFR.
Test Steps	• The evaluator created a RootCA certificate that signed the server certificate (https_server) and also created a RootCA_client certificate that signed the client certificate using the XCA tool. • The evaluator ensured that the server certificate was signed by the RootCA. • The evaluator imported the RootCA certificate into the key vault of the TOE that signed the server certificate. • The evaluator loaded the server certificate (https_server.pem) along with the key to the TOE's system keyvault. • The evaluator configured the TOE's Administration server in System > Admin server where the port was set 8001 in General. • The evaluator selected the TLS protocol, supported cipher suites, the server certificate that must be used for TLS handshake which was previously uploaded on the TOE and saved the configuration. • The evaluator confirmed that the RootCA_client certificate that signed the client certificate was not uploaded to the TOE's system keyvault. • The evaluator uploaded the client certificate to the client VM and ensured that the certificate was signed by the RootCA_client certificate. • The evaluator initiated the TLS connection using the client certificate that was signed by the RootCA_client certificate which was not present in the TOE's trust

	store (system keyvault) and ensured that the TLS connection did not succeed due to certificate unknown error. • The evaluator observed the packet capture to ensure that the TLS handshake was not successful due to certificate unknown alert returned by the server after the client sent the certificate. • The evaluator further observed the logs located at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs to ensure that the TLS handshake was unsuccessful as the TOE was unable to construct a valid chain and no issuer certificate for the client certificate in the certificate was found. • The evaluator then loaded the RootCA_client certificate authority needed to validate the client certificate to the TOE's trust store (system keyvault) in Encryption > Key Management services. • The evaluator confirmed that the certificate authority was imported successfully. • The evaluator initiated the TLS connection using the client certificate that was signed by the RootCA_client certificate which was imported in the TOE's trust store (system keyvault) and ensured that the TLS connection established successfully. • The evaluator observed the packet capture on the client VM to confirm that the TLS connection established successfully. • The evaluator then deleted the RootCA_client certificate authority from the TOE's certificate trust store (system keyvault) that signed the client's certificate. • The evaluator confirmed that the RootCA_client certificate authority was not present in the TOE's trust store. • The evaluator initiated the TLS connection using the client certificate that was signed by the RootCA_client certificate which was not present in the TOE's trust store (system keyvault) and ensured that the TLS connection did not succeed due to certificate unknown error. • The evaluator observed the packet capture to ensure that the TLS handshake was not successful due to certificate unknown alert returned by the server after the client sent the certificate. • The evaluator further observed the logs located at C:\Program Files\HelpSystems\GoAnywhere\userdata\logs to ensure that the TLS handshake was unsuccessful as the TOE was unable to construct a valid chain and no issuer certificate for the client certificate in the certificate was found.
Pass/Fail with Explanation	Pass. The evaluator verified that when the server was presented with a client certificate without having its issuer certificate in the TOE's trust store or the certification path resulted in failure to validate the certificate. This meets the testing requirements.

7 Security Assurance Requirements

7.1 AGD_OPE.1 Operational User Guidance

7.1.1 AGD_OPE.1

7.1.1.1 AGD_OPE.1 Guidance 1

Objective	If cryptographic functions are provided by the TOE, the operational guidance shall contain instructions for configuring the cryptographic engine associated with the evaluated configuration of the TOE. It shall provide a warning to the administrator that use of other cryptographic engines was not evaluated nor tested during the CC evaluation of the TOE.
Evaluator Findings	The evaluator examined the section titled **'Enabling FIPS 140-2 mode'** in the AGD to verify that it contains instructions for configuring the cryptographic engine associated with the evaluated configuration of the TOE. Upon investigation, the evaluator found that the AGD provides instructions on enabling FIPS 140-2 mode which utilizes the GoAnywhere MFT Bouncy Castle FIPS Java API cryptographic library version 1.0.2. This library implements all the cryptographic algorithms required for SSH and TLS, drawing entropy from the platform RBG. The evaluator also examined the section titled **'Enabling FIPS 140-2 mode'** in the AGD to verify that it provides a warning to the administrator that use of other cryptographic engines was not evaluated nor tested during the CC evaluation of the TOE. Upon investigation, the evaluator found that the AGD states that the Federal Information Processing Standard (FIPS) is a set of requirements used by the US Federal Government and agencies/companies that do business with them to ensure all sensitive data is encrypted with approved encryption algorithms (ciphers). GoAnywhere provides a FIPS 140-2 Compliance Mode and when enabled, it only permits the use of FIPS 140-2 compliant ciphers for encrypting the data. The Administrator must ensure that the FIPS 140-2 mode is always enabled to implement only evaluated encryption algorithms as other cryptographic engines were not evaluated or tested during the Common Criteria evaluation of the product.
Verdict	Pass

7.1.1.2 AGD_OPE.1 Guidance 2

Objective	The documentation must describe the process for verifying updates to the TOE by verifying a digital signature – this may be done by the TOE or the underlying platform. The evaluator shall verify that this process includes the following steps: • Instructions for obtaining the update itself. This should include instructions for making the update accessible to the TOE (e.g., placement in a specific directory). • Instructions for initiating the update process, as well as discerning whether the process was successful or unsuccessful. This includes generation of the digital signature. The TOE will likely contain security functionality that does not fall in the scope of evaluation under this PP. The operational guidance shall make it

	clear to an administrator which security functionality is covered by the evaluation activities.
Evaluator Findings	The evaluator examined the section titled **'Secure Updates'** in the AGD to verify that it describes the process for verifying updates to the TOE by verifying a digital signature. Upon investigation, the evaluator found that the AGD states that updates to the TOE are digitally signed and verified by the platform (Windows Installer or RPM Package manager) prior to installation. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

7.2 AGD_PRE.1 Preparative Procedures

7.2.1 AGD_PRE.1

7.2.1.1 AGD_PRE.1 Guidance 1

Objective	As indicated in the introduction above, there are significant expectations with respect to the documentation—especially when configuring the operational environment to support TOE functional requirements. The evaluator shall check to ensure that the guidance provided for the TOE adequately addresses all platforms claimed for the TOE in the ST.
Evaluator Findings	The evaluator examined the section titled **'Evaluated Configuration'** in the AGD to verify that it adequately addresses all platforms claimed for the TOE in the ST. Upon investigation, the evaluator found that the AGD states that the TOE has been evaluated on the following host platforms: • CentOS 7 on ESXi 6.7 with Intel Xeon E5-4620v4 (Broadwell) • Windows Server 2016 on ESXi 6.7 with Intel Xeon E5-4620v4 (Broadwell) Note: The TOE is the application software only. The host platforms are not part of the evaluation. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

7.3 ALC Assurance Activities

7.3.1 ALC_CMC.1

7.3.1.1 ALC_CMC.1 TSS 1

Objective	The evaluator shall check the ST to ensure that it contains an identifier (such as a product name/version number) that specifically identifies the version that meets the requirements of the ST.
Evaluator Findings	The evaluator examined the section titled **'TOE Summary Specification'** in the Security Target to verify that the ST contains an identifier (such as a product name/version number) that specifically identifies the version that meets the requirements of the ST. Upon investigation, the evaluator found that the ST states that the Configuration

	Management (CM) documents describe how the consumer identifies the evaluated TOE. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

7.3.1.2 ALC_CMC.1 TSS 2

Objective	If the vendor maintains a web site advertising the TOE, the evaluator shall examine the information on the web site to ensure that the information in the ST is sufficient to distinguish the product.
Evaluator Findings	The evaluator examined the vendor web site to ensure that the information in the ST is sufficient to distinguish the product. Upon investigation, the evaluator found that the company website advertises the TOE in a manner which sufficiently distinguishing the product. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

7.3.1.3 ALC_CMC.1 Guidance 1

Objective	Further, the evaluator shall check the AGD guidance to ensure that the version number is consistent with that in the ST.
Evaluator Findings	The evaluator examined the title page in the AGD to verify that the version number is consistent with that in the ST. Upon investigation, the evaluator found that the AGD states that the TOE is version 6.8 and is consistent with the version in the ST. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

7.3.2 ALC_CMS.1

7.3.2.1 ALC_CMS.1 Guidance 1

Objective	The evaluator shall ensure that the developer has identified (in guidance documentation for application developers concerning the targeted platform) one or more development environments appropriate for use in developing applications for the developer's platform. For each of these development environments, the developer shall provide information on how to configure the environment to ensure that buffer overflow protection mechanisms in the environment(s) are invoked (e.g., compiler flags). The evaluator shall ensure that this documentation also includes an indication of whether such protections are on by default, or have to be specifically enabled.
Evaluator Findings	The evaluator examined the section titled **'Operational Environment'** in the platform developer guidance documentation to verify that it identifies one or more development environments appropriate for use in developing applications for the developer's platform. For each of these development environments, the evaluator verified that the developer provides information on how to configure the environment to ensure that buffer overflow protection mechanisms in the environment(s) are

	invoked (e.g., compiler flags) and whether such protections are on by default. Upon investigation, the evaluator found that the guidance documentation identifies the operational environment and the configuration list in 'Table 1 IT Environment Components." In the section titled 'Other Assumptions' the AGD also states that the TOE on Windows Platform is composed of Java and native code. The native code implements stack-based buffer overflow protections, being compiled with the /GS flag on Windows. All Java objects are strictly typed with explicit sizes, so it is not possible to overflow a buffer in Java code. The TOE on Linux Platform is composed of Java code. All Java objects are strictly typed with explicit sizes, so it is not possible to overflow a buffer in Java code. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

7.3.2.2 ALC_CMS.1 Guidance 2

Objective	The evaluator shall ensure that the TSF is uniquely identified (with respect to other products from the TSF vendor), and that documentation provided by the developer in association with the requirements in the ST is associated with the TSF using this unique identification.
Evaluator Findings	The evaluator examined the section titled **'Purpose of this document'** in the AGD to verify that it is associated with the TSF using unique identification. Upon investigation, the evaluator found that the guidance documentation supports the usage of the unique identifier, 'Fortra's GoAnywhere Managed File Transfer v6.8' for this TOE and it is reflected across all documentation. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

7.3.3 ALC_TSU.1

7.3.3.1 ALC_TSU.1 TSS 1

Objective	The evaluator shall verify that the TSS contains a description of the timely security update process used by the developer to create and deploy security updates. The evaluator shall verify that this description addresses the entire application. The evaluator shall also verify that, in addition to the TOE developer's process, any third-party processes are also addressed in the description. The evaluator shall also verify that each mechanism for deployment of security updates is described.
Evaluator Findings	The evaluator examined the section titled **'TOE Summary Specification'** in the Security Target to verify that the TSS contains a description of the timely security update process that addresses the entire application (including third-party processes). Upon investigation, the evaluator found that the TSS states that Fortra uses various security tools to regularly scan the TOE throughout the development lifecycle. Vulnerability reports are submitted using a form on the Fortra website which is protected using HTTPS. This protects the confidentiality of the vulnerability report.

	The evaluator also examined the section titled **'TOE Summary Specification'** in the Security Target to verify that each mechanism for deployment of security updates is described. Upon investigation, the evaluator found that the TSS states that GoAnywhere Support and Development Teams collaborate to evaluate any reports of application vulnerabilities received. The teams work to understand the issue, understand the impact, and evaluate potential courses of action. After confirming and understanding the issue, the GoAnywhere team prepares the appropriate remediation for the problem. GoAnywhere Support and Development Teams collaborate to evaluate any reports of application vulnerabilities received. The teams work to understand the issue, understand the impact, and evaluate potential courses of action. After confirming and understanding the issue, the GoAnywhere team prepares the appropriate remediation for the problem. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

7.3.3.2 ALC_TSU.1 TSS 2

Objective	The evaluator shall verify that, for each deployment mechanism described for the update process, the TSS lists a time between public disclosure of a vulnerability and public availability of the security update to the TOE patching this vulnerability, to include any third-party or carrier delays in deployment. The evaluator shall verify that this time is expressed in a number or range of days.
Evaluator Findings	The evaluator examined the section titled **'TOE Summary Specification'** in the Security Target to verify that, for each deployment mechanism described for the update process, the TSS lists a time between public disclosure of a vulnerability and public availability of the security update to the TOE patching this vulnerability. Upon investigation, the evaluator found that the TSS states that Fortra strives to meet the following timelines for addressing vulnerabilities: • Zero-day and Critical Vulnerabilities: within a week • High Vulnerabilities: within 30 days • Medium and Low Vulnerabilities: 3-4 months • If the vulnerability is in a third-party library, then Fortra must wait for the library developers to address the issue, but Fortra will provide mitigation recommendations to minimize potential risk. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

7.3.3.3 ALC_TSU.1 TSS 3

Objective	The evaluator shall verify that this description includes the publicly available mechanisms (including either an email address or website) for reporting security issues related to the TOE. The evaluator shall verify that the description of this mechanism

	includes a method for protecting the report either using a public key for encrypting email or a trusted channel for a website.
Evaluator Findings	The evaluator examined the section titled **'TOE Summary Specification'** in the Security Target to verify that the TSS includes the publicly available mechanisms for reporting security issues related to the TOE, including a method for protecting the report. Upon investigation, the evaluator found that the TSS states that vulnerability reports are submitted using a form on the Fortra website which is protected using HTTPS. This protects the confidentiality of the vulnerability report. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

7.3.4 ATE_IND.1.2E Test 1

Objective	The evaluator shall prepare a test plan and report documenting the testing aspects of the system, including any application crashes during testing. The evaluator shall determine the root cause of any application crashes and include that information in the report. The test plan covers all of the testing actions contained in the [CEM] and the body of this PP's evaluation activities. While it is not necessary to have one test case per test listed in an evaluation activity, the evaluator must document in the test plan that each applicable testing requirement in the ST is covered. The test plan identifies the platforms to be tested, and for those platforms not included in the test plan but included in the ST, the test plan provides a justification for not testing the platforms. This justification must address the differences between the tested platforms and the untested platforms, and make an argument that the differences do not affect the testing to be performed. It is not sufficient to merely assert that the differences have no effect; rationale must be provided. If all platforms claimed in the ST are tested, then no rationale is necessary. The test plan describes the composition of each platform to be tested, and any setup that is necessary beyond what is contained in the AGD documentation. It should be noted that the evaluator is expected to follow the AGD documentation for installation and setup of each platform either as part of a test or as a standard pre-test condition. This may include special test drivers or tools. For each driver or tool, an argument (not just an assertion) should be provided that the driver or tool will not adversely affect the performance of the functionality by the TOE and its platform. This also includes the configuration of the cryptographic engine to be used. The cryptographic algorithms implemented by this engine are those specified by this PP and used by the cryptographic protocols being evaluated (e.g SSH). The test plan identifies high-level test objectives as well as the test procedures to be followed to achieve those objectives. These procedures include expected results. The test report (which could just be an annotated version of the test plan) details the

	activities that took place when the test procedures were executed, and includes the actual results of the tests. This shall be a cumulative account, so if there was a test run that resulted in a failure; a fix installed; and then a successful re-run of the test, the report would show a "fail" and "pass" result (and the supporting details), and not just the "pass" result.
Evaluator Findings	In support of the AAs in the PP, the evaluator created a test plan. This test plan includes an equivalency argument, a description of the test infrastructure (including the host platforms), each test case, and actual results for each test case. Based on these findings, this work unit is considered satisfied
Verdict	Pass

7.4 AVA_VAN.1 Vulnerability Survey

7.4.1 AVA_VAN.1

7.4.1.1 AVA_VAN.1 Activity 1

Objective	The evaluator shall generate a report to document their findings with respect to this requirement. This report could physically be part of the overall test report mentioned in ATE_IND, or a separate document. The evaluator performs a search of public information to find vulnerabilities that have been found in similar applications with a particular focus on network protocols the application uses and document formats it parses. The evaluator documents the sources consulted and the vulnerabilities found in the report. For each vulnerability found, the evaluator either provides a rationale with respect to its non-applicability, or the evaluator formulates a test (using the guidelines provided in ATE_IND) to confirm the vulnerability, if suitable. Suitability is determined by assessing the attack vector needed to take advantage of the vulnerability. If exploiting the vulnerability requires expert skills and an electron microscope, for instance, then a test would not be suitable and an appropriate justification would be formulated. **TD0554 has been applied.**
Evaluator Findings	The evaluator documented their analysis and testing of potential vulnerabilities with respect to this requirement. Public searches were performed against all keywords found within the Security Target and AGD that may be applicable to specific TOE components. This included protocols, TOE software version, and TOE hardware to ensure sufficient coverage under AVA. The evaluator searched the Internet for potential vulnerabilities in the TOE using the web sites listed below. The sources of the publicly available information are provided below. • https://www.goanywhere.com • http://nvd.nist.gov/ • http://www.us-cert.gov • http://www.securityfocus.com/ • https://www.cvedetails.com/

The evaluator performed the public domain vulnerability searches using the following key words. The search was performed on 01/15/2023 and then re-performed on 03/29/2023.

Component	CPE
fortra	cpe:2.3:a:fortra
helpsystems 6.6.0	cpe:2.3:a:helpsystems:boks:6.6.0:*:*:*:*:*:*:*
helpsystems 6.7.1	cpe:2.3:a:helpsystems:boks:6.7.1:*:*:*:*:*:*:*
helpsystems 6.8.7	cpe:2.3:a:goanywhere:mft:6.8.7:*:*:*:*:*:*:*
goanywhere	cpe:2.3:a:goanywhere:*:*:*:*:*:*:*
GoAnywhere MFT Bouncy Castle FIPS Java API	cpe:2.3:a:GoAnywhereMFTBouncyCastleFIPSJavaAPI:*:*:*:*:*:*:*
centos 7.0	cpe:2.3:o:centos:centos:7.0:*:*:*:*:*:*:*
intel xeon e5-4620 v4	cpe:2.3:h:intel:xeon_e5-4620_v4:-:*:*:*:*:*:*:*
Azul Zulu Java SE 8 Update 272	cpe:2.3:a:azul:zulu:8:update272:*:*:*:*:*:*
vmware esxi 6.7	cpe:2.3:o:vmware:esxi:6.7:-:*:*:*:*:*:*
all-themes-1.0.8.jar	cpe:2.3:a:all-themes-1.0.8.jar:*:*:*:*:*:*:*
apache tomcat 9:0:41	cpe:2.3:a:apache:tomcat:9.0.41:*:*:*:*:*:*:*
apache-mime4j-core-0.7.2	cpe:2.3:a:apache:mime4j:core-0.7.2:*:*:*:*:*:*:*
aws-java-sdk-cloudfront	cpe:2.3:a:amazon:aws_sdk_for_cloudfront:-:*:*:*:*:node.js:*:*
aws-java-sdk-core-1.11.631	cpe:2.3:a:amazon:aws_sdk_for_core:1.11.631:*:*:*:*:node.js:*:*
aws-java-sdk-kms-1.11.631	cpe:2.3:a:amazon:aws_sdk_for_kms:1.11.631:*:*:*:*:node.js:*:*
aws-java-sdk-s3-1.11.631	cpe:2.3:a:amazon:aws_s3_crypto_sdk:1:*:*:*:*:golang:*:*
aws-java-sdk-sts-1.11.631	cpe:2.3:a:amazon:aws_java_sdk_sts:1:*:*:*:*:golang:*:*
azure storage 5.5.0	cpe:2.3:a:azure:storage:5.5.0:*:*:*:*:*:*:*
apache batik 1.10	cpe:2.3:a:apache:batik:1.10:*:*:*:*:*:*:*
bouncy castle fips 1.0.2	cpe:2.3:a:bouncycastle:fips_java_api:1.0.2:*:*:*:*:*:*:*
bouncy castle mail fips 1.0.3	cpe:2.3:a:bouncycastle:mail:fips:1.0.3:*:*:*:*:*:*
bouncy castle pg fips 1.0.5	cpe:2.3:a:bouncycastle:pg:fips:1.0.5:*:*:*:*:*:*

	bouncy castle cryptography APIs	cpe:2.3:a:bouncycastle:legion-of-the-bouncy-castle-java-crytography-api:1.02:*:*:*:*:*:*:*
	bctls fips 1.0.10.3	cpe:2.3:a:bouncycastle:tls:fips:1.0.10.3:*:*:*:*:*:*
	bluesky 1.0.6	cpe:2.3:a:bluesky:1.0.6:*:*:*:*:*:*
	bsh-2.0b6	cpe:2.3:a:beanshell:beanshell:2.0:beta6:*:*:*:*:*:*
	chartcreator-1.2.0	cpe:2.3:a:chartcreator:1.2.0:*:*:*:*:*:*
	commons_beanutils 1.9.4	cpe:2.3:a:apache:commons_beanutils:1.9.4:*:*:*:*:*:*:*
	commons-cli 1.3.1	cpe:2.3:a:apache:commons-cli:1.3.1:*:*:*:*:*:*:*
	commons-codec 1.14	cpe:2.3:a:apache:commons-codec:1.14:*:*:*:*:*:*:*
	commons collections 3.2.2	cpe:2.3:a:apache:commons_collections:3.2.2:*:*:*:*:*:*:*
	commons collections 4.4.1	cpe:2.3:a:apache:commons_collections:4.4.1:*:*:*:*:*:*:*
	commons compress 1.19	cpe:2.3:a:apache:commons_compress:1.19:*:*:*:*:*:*:*
	commons configuration 1.7	cpe:2.3:a:apache:commons_configuration:1.7:*:*:*:*:*:*:*
	commons dbcp 1.3	cpe:2.3:a:apache:commons:dbcp:1.3:*:*:*:*:*:*
	commons digestoer 1.8.1	cpe:2.3:a:apache:commons:digester:1.8.1:*:*:*:*:*:*
	commons discovery 0.4	cpe:2.3:a:apache:commons:discovery:0.4:*:*:*:*:*:*
	commons-el	cpe:2.3:a:apache:commons:el:*:*:*:*:*:*
	commons fileupload 1.4	cpe:2.3:a:apache:commons_fileupload:1.4:*:*:*:*:*:*:*
	commons httpclient 3.0	cpe:2.3:a:apache:commons-httpclient:3.0:*:*:*:*:*:*:*
	commons-io 2.6	cpe:2.3:a:apache:commons-io:2.6:*:*:*:*:*:*:*
	commons-lang 2.1	cpe:2.3:a:apache:commons-lang:2.1:*:*:*:*:*:*:*
	commons-lang3 3.9	cpe:2.3:a:apache:commons-lang3:3.9:*:*:*:*:*:*:*
	commons logging 1.2	cpe:2.3:a:apache:commons-logging:1.2:*:*:*:*:*:*:*
	commons math3 3.6.1	cpe:2.3:a:apache:commons-math3:3.6.1:*:*:*:*:*:*:*
	commons-net-3.3.0	cpe:2.3:a:netcommons:netcommons:3.3.0:*:*:*:*:*:*:*
	commons-pool-1.6	cpe:2.3:a:apache:commons-pool:1.6:*:*:*:*:*:*:*
	commons-validator-1.5.0	cpe:2.3:a:apache:commons-validator:1.5.0:*:*:*:*:*:*:*
	commons-vfs2-2.1	cpe:2.3:a:apache:commons-vfs2:2.1:*:*:*:*:*:*:*

	cryptojce	cpe:2.3:o:cryptojce:*:*:*:*:*:*:*
	cryptojcommon	cpe:2.3:o:cryptojcommon:*:*:*:*:*:*:*
	css parser	cpe:2.3:a:horde:horde_css_parser:1.0.0:*:*:*:*:*:*:*
	curvesapi 1.0.6	cpe:2.3:o:curvesapi:1.0.6:*:*:*:*:*:*:*
	db2jcc	cpe:2.3:a:ibm:db2:11.1:*:*:*:*:*:*:*
	derby	cpe:2.3:a:apache:derby:-:*:*:*:*:*:*:*
	derby client	cpe:2.3:a:apache:derby:client:*:*:*:*:*:*:*
	ehcache-core-2.5.1	cpe:2.3:a:ehcache:core:2.5.1:*:*:*:*:*:*:*
	esapi-2.1.0.1	cpe:2.3:a:owasp:enterprise_security_api:2.1.0.1:*:*:*:*:*:*:*
	facestrace 0.9.0	cpe:2.3:a:facestrace:0.9.0:*:*:*:*:*:*:*
	face info set	cpe:2.3:a:faceinfoset:*:*:*:*:*:*:*
	font awesome 5.6.1	cpe:2.3:a:font:awesome:5.6.1:*:*:*:*:*:*:*
	gmbal-api-only	cpe:2.3:a:oracle:glassfish:-:*:*:*:*:*:*:*
	gson 2.2.4	cpe:2.3:a:gson:2.2.4:*:*:*:*:*:*:*
	guava 26.0	cpe:2.3:a:google:guava:26.0:*:*:*:*:*:*:*
	ha-api	cpe:2.3:a:ha:api:*:*:*:*:*:*:*
	httpclient 4.5.13	cpe:2.3:a:apache:httpclient:4.5.13:*:*:*:*:*:*:*
	httpcore 4.4.14	cpe:2.3:a:apache:httpcore:4.4.14:*:*:*:*:*:*:*
	icu4j-63.1	cpe:2.3:a:icu-project:international_components_for_unicode:63.1:*:*:*:*:c\/c\+\+:*:*
	ifxjdbc	cpe:2.3:a:ibm:informix_jdbc:*:*:*:*:*:*:*
	imagscalr-lib 4.2	cpe:2.3:a:imgscalr:lib:4.2:*:*:*:*:*:*:*
	ion-java-1.0.2	cpe:2.3:a:amazon:ion:1.02:*:*:*:*:node.js:*:*
	ipworkszip	cpe:2.3:a:ipworkszip:*:*:*:*:*:*:*
	itext 2.1.7	cpe:2.3:a:itextpdf:itext:2.1.7:*:*:*:*:*:*:*
	jackson annotations	cpe:2.3:a:fasterxml:jackson:2.10.0:*:*:*:*:*:*:*
	jackson core	cpe:2.3:a:fasterxml:jackson-core:*
	jackson databind 2.10.5	cpe:2.3:a:fasterxml:jackson-databind:2.10.5:*:*:*:*:*:*:*
	jakartha oro	cpe:2.3:a:jakartha:oro:*:*:*:*:*:*:*
	jasperreports 6.7.1	cpe:2.3:a:jaspersoft:jasperreports:6.7.1:*:*:*:*:*:*:*
	jasperreports-chart-themes 6.7.0	cpe:2.3:a:jaspersoft:jasperreports-chart-themes:6.7.0:*:*:*:*:*:*:*
	jasperreports-fonts 6.7.1	cpe:2.3:a:jaspersoft:jasperreports-fonts:6.7.1:*:*:*:*:*:*:*
	jasypt 1.9.2	cpe:2.3:a:jasypt_project:jasypt:1.9.2:*
	java jwt 3.3.0	cpe:2.3:a:java:jwt:3.3.0:*:*:*:*:*:*:*
	javax annotation	cpe:2.3:a:oracle:javax:annotation:*:*:*:*:*:*:*
	java xml soap	cpe:2.3:a:javax:xml:soap:*:*:*:*:*:*:*

	jaxb-api	cpe:2.3:o:jaxb-api:*:*:*:*:*:*:*
	jaxb core	cpe:2.3:o:jaxb-core:*:*:*:*:*:*:*
	jaxb-impl	cpe:2.3:o:jaxb-impl:*:*:*:*:*:*:*
	jaxb-jxc	cpe:2.3:o:jaxb-jxc:*:*:*:*:*:*:*
	jaxb-xjc	cpe:2.3:o:jaxb-xjc:*:*:*:*:*:*:*
	jaxws-rt	cpe:2.3:o:jaxws-rt:*:*:*:*:*:*:*
	jaxws-tools	cpe:2.3:o:jaxws-tools:*:*:*:*:*:*:*
	jcifs 1.3.18	cpe:2.3:o:jcifs:1.3.18:*:*:*:*:*:*:*
	jcmFIPS	cpe:2.3:a:oracle:jcmFIPS:*:*:*:*:*:*:*
	jcommon-1.0.10	cpe:2.3:a:oracle:jcommon:1.0.10:*:*:*:*:*:*:*
	jfreechat 1.0.19	cpe:2.3:a:oracle:jfreechart:1.0.19:*:*:*:*:*:*:*
	jgroups 4.1.2	cpe:2.3:a:jgroups:jgroup:4.1.2:*:*:*:*:*:*:*
	jmespath java 1.11.631	cpe:2.3:a:amazon:jmespath:java:1.11.631:*:*:*:*:*:*
	jms	cpe:2.3:a:jenkins:jms_messaging:1.1.1:*:*:*:*:jenkins:*:*
	jnq 1.3.6	cpe:2.3:o:jnq:1.3.6:*:*:*:*:*:*:*
	joda-time 2.2	cpe:2.3:o:joda-time:2.2:*:*:*:*:*:*:*
	jsch 0.1.54	cpe:2.3:o:jsch:0.1.54:*:*:*:*:*:*:*
	jsr181 api	cpe:2.3:a:jsr181:*:*:*:*:*:*
	jt400	cpe:2.3:a:jt400:*:*:*:*:*:*
	jTDS3	cpe:2.3:a:jTDS3:*:*:*:*:*:*
	jxl	cpe:2.3:a:jxl:*:*:*:*:*:*
	jzlib 1.1.2	cpe:2.3:a:jcraft:jzlib:1.1.2:*:*:*:*:*:*:*
	log4j 1.2	cpe:2.3:a:apache:log4j:1.2:-:*:*:*:*:*:*
	log4j-1.2-api-2.13.3	cpe:2.3:a:apache:log4j:2.13.3:rc1:*:*:*:*:*:*
	log4j-core 2.13.3	cpe:2.3:a:apache:log4j-core:2.13.3:rc1:*:*:*:*:*:*
	log4j-slf4j-impl-2.13.3	cpe:2.3:a:slf4j:slf4j-log4j-2:13.3:*:*:*:*:*:*:*
	lucene analyzers common 4.7.2	cpe:2.3:a:apache:lucene-analyzers:common:4.7.2*:*:*:*:*:*:*
	lucene codecs 4.7.2	cpe:2.3:a:apache:lucene-codecs:4.7.2*:*:*:*:*:*:*
	lucene core 4.7.2	cpe:2.3:a:apache:lucene-core:4.7.2*:*:*:*:*:*:*
	lucene-grouping 4.7.2	cpe:2.3:a:apache:lucene-grouping:4.7.2*:*:*:*:*:*:*
	lucene-queries 4.7.2	cpe:2.3:a:apache:lucene-queries:4.7.2*:*:*:*:*:*:*
	lucene-queryparser 4.7.2	cpe:2.3:a:apache:lucene-queryparser:4.7.2*:*:*:*:*:*:*
	management-api	cpe:2.3:a:management-api:*:*:*:*:*:*

	mariadb-java-client 1.7.1	cpe:2.3:a:mariadb-java-client:1.7.1:*:*:*:*:*:*:*
	maverick-legacy-server 1.7.34	cpe:2.3:a:maverick-legacy-server:1.7.34:*:*:*:*:*:*:*
	mimepull	cpe:2.3:a:mimepull:*:*:*:*:*:*:*
	mina-core 2.1.4	cpe:2.3:a:apache:mina:2.1.4:*:*:*:*:*:*:*
	msbase	cpe:2.3:a:msbase:*:*:*:*:*:*:*
	mssqlserver	cpe:2.3:a:mssqlserver:*:*:*:*:*:*:*
	msutil	cpe:2.3:a:msutil:*:*:*:*:*:*:*
	myfaces 2.2.12	cpe:2.3:a:apache:myfaces:2.2.12:*:*:*:*:*:*:*
	native-lib-loader 2.0.2	cpe:2.3:a:native-lib-loader:2.0.2:*:*:*:*:*:*:*
	netty 4.1.48	cpe:2.3:a:netty:netty:4.1.48:*:*:*:*:*:*:*
	not going to be common ssl 0.3.18	cpe:2.3:a:not-going-to-be-common:ssl:0.3.18*:*:*:*:*:*:*
	ojdbc5	cpe:2.3:a:ojdbc5:*:*:*:*:*:*:*
	opensaml 2.6.6	cpe:2.3:a:shibboleth:opensaml:2.6.6:*:*:*:*:*:*:*
	openws 1.5.4	cpe:2.3:a:shibboleth:openws:1.5.4:*:*:*:*:*:*:*
	oro 2.0.8	cpe:2.3:a:jahia:oro:2.0.8:*:*:*:*:*:*:*
	owasp sanitizer	cpe:2.3:a:owasp:json-sanitizer:*:*:*:*:*:*:*
	poi 4.1.1	cpe:2.3:a:apache:poi:4.1.1:*:*:*:*:*:*:*
	poi ooxml 4.1.1	cpe:2.3:a:apache:poi-ooxml:4.1.1:*:*:*:*:*:*:*
	poi ooxml schemas 4.4.1	cpe:2.3:a:apache:poi-ooxml-schemas:4.1.1:*:*:*:*:*:*:*
	policy	cpe:2.3:a:policy:*:*:*:*:*:*:*
	postgresql 42.2.14	cpe:2.3:a:postgresql:postgresql_jdbc_driver:42.2.14:*:*:*:*:*:*:*
	prettyfaces-jsf2 3.3.0	cpe:2.3:a:apache:prettyfaces-jsf2:3.3.0:*:*:*:*:*:*:*
	primefaces 7.0.14	cpe:2.3:a:primetek:primefaces:7.0.14:*:*:*:*:*:*:*
	primefaces-extensions 7.0.1	cpe:2.3:a:primetek:primefaces-extensions:7.0.1:*:*:*:*:*:*:*
	qname	cpe:2.3:a:qname:*:*:*:*:*:*:*
	resolver	cpe:2.3:a:resolver:jar:*:*:*:*:*:*:*
	saaj-impl	cpe:2.3:a:sun:saaj:impl:*:*:*:*:*:*:*
	sardine	cpe:2.3:a:sardine:*:*:*:*:*:*:*
	slf4j-api-1.7.25	cpe:2.3:a:qos:slf4j:1.7.25:*:*:*:*:*:*:*
	snmp4j 2.3.4	cpe:2.3:a:snmp4j:2.3.4:*:*:*:*:*:*:*
	spring beans 5.2.9	cpe:2.3:a:spring-beans:5.2.9:*:*:*:*:*:*:*
	spring context 5.2.9	cpe:2.3:a:spring-context:5.2.9:*:*:*:*:*:*:*
	spring core 5.2.9	cpe:2.3:a:spring-core:5.2.9:*:*:*:*:*:*:*
	sqljdbc4	cpe:2.3:a:microsoft:sqljdbc4*:*:*:*:*:*:*

sslj	cpe:2.3:a:sslj:*:*:*:*:*:*:*
stax2-api	cpe:2.3:a:stax2-api:*:*:*:*:*:*:*
stax2-api-3.1.4	cpe:2.3:a:stax2-api:3.1.4:*:*:*:*:*:*:*
stax2-api-1.0.2	cpe:2.3:a:stax2-api:1.0.2:*:*:*:*:*:*:*
stax-ex	cpe:2.3:a:stax-ex:*:*:*:*:*:*:*
streambuffer	cpe:2.3:a:streambuffer:*:*:*:*:*:*:*
taglibs-standard 1.2.3	cpe:2.3:a:apache:standard_taglibs:1.2.1:*:*:*:*:*:*:*
tinyradius 1.1.0	cpe:2.3:a:tinyradius:1.1.0:*:*:*:*:*:*:*
tomahawk20-1.1.14	cpe:2.3:a:apache:myfaces_tomahawk:1.1.14:*:*:*:*:*:*:*
unboundid-ldapsdk-4.0.11	cpe:2.3:a:pingidentity:ldapsdk:4.0.11:*:*:*:*:java:*:*
velocity-1.7	cpe:2.3:a:apache:velocity_engine:1.7:*:*:*:*:*:*:*
woodstox-core-asl	cpe:2.3:a:apache:woodstox-core-asl:*:*:*:*:*:*:*
woodstox-core-asl-4.4.1	cpe:2.3:a:apache:woodstox-core-asl:4.4.1:*:*:*:*:*:*:*
wsbuilder	cpe:2.3:a:apache:wsbuilder:*:*:*:*:*:*:*
wsdl4j	cpe:2.3:a:wsdl4j:*:*:*:*:*:*:*
xml-apis 1.3.04	cpe:2.3:a:xmlapis:1.3.04:*:*:*:*:*:*:*
xmlbeans 3.1.0	cpe:2.3:a:apache:xmlbeans:3.1.0:*:*:*:*:*:*:*
xmlgraphics commons 2.2	cpe:2.3:a:apache:xmlgraphics_commons:2.2:*:*:*:*:*:*:*
xmlsec 2.1.4	cpe:2.3:a:xmlseclibs_project:xmlseclibs:2.1.4:*:*:*:*:*:*:*
xmltooling 1.4.6	cpe:2.3:a:xmltooling_project:xmltooling:1.5.4:*:*:*:*:*:*:*
openjdk 1.8.0	cpe:2.3:a:oracle:openjdk:1.8.0:*:*:*:*:*:*:*
microsoft windows server 2016	cpe:2.3:o:microsoft:windows_server_2016:-:*:*:*:*:*:*:*

Based upon the analysis, any issues found were patched in the TOE version and prior versions, mitigating the risk factor. Details can be found in the separate Vulnerability Analysis document.

Based on these findings, this assurance activity is considered satisfied.

Verdict	Pass

7.4.1.2 AVA_VAN.1 Activity 2

Objective	**Conditional for Windows, Linux, macOS and Solaris:** The evaluator shall also run a virus scanner with the most current virus definitions against the application files and verify that no files are flagged as malicious. **TD0554 has been applied.**

Evaluator Findings	The evaluator documented their analysis and testing of potential malicious files with respect to this requirement. The evaluator performed the virus scans using ClamAV antivirus software on the Linux platform and Windows Defender Antivirus scanner on Windows platform with the latest virus definitions. The scan was performed on 01/05/2023. Based upon the analysis, no malicious files were identified. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

8 Conclusion

The testing shows that all test cases required for conformance have passed testing.

9 Appendix A: CAVP Certificate Table

The This section provides a table that lists all SFRs for which a CAVP certificate is claimed, the cryptographic operation, the NIST standard, the SFR supported, the CAVP algorithm list name and the CAVP Certificate number.

Table 1 - CAVP Table

SFR	Algorithm in ST	Implementation name	CAVP Alg.	CAVP Cert #
FCS_CKM.1	RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3	GoAnywhere MFT Bouncy Castle FIPS Java API	RSA KeyGen (n = 2048, 3072)	C1876
	ECC schemes using "NIST curves" [selection: P-256, P-384, P-521] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4	GoAnywhere MFT Bouncy Castle FIPS Java API	ECDSA KeyGen ECDSA KeyVer (Curve = P-256, P-384, P-521)	C1876
	FFC Schemes using Diffie-Hellman group 14 that meet the following: RFC 3526, Section 3	GoAnywhere MFT Bouncy Castle FIPS Java API	NIAP Policy Letter #5, Addendum #2, states "No NIST CAVP, CCTL must perform all assurance/evaluation activities".	Evaluator Affirmed.
FCS_CKM.2	RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1"	GoAnywhere MFT Bouncy Castle FIPS Java API	NIAP Policy Letter #5, Addendum #2, states "No NIST CAVP exists, must be described in TSS – See FIPS 140-2 I.G. D.4: Vendor Affirmation".	Evaluator Affirmed.
	Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 2, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"	GoAnywhere MFT Bouncy Castle FIPS Java API	KAS-ECC (Curve = P-256, P-384, P-521)	C1876
	Key establishment scheme using Diffie-Hellman group 14 that meets the following: RFC 3526, Section 3	GoAnywhere MFT Bouncy Castle FIPS Java API	NIAP Policy Letter #5, Addendum #2 does not provide any guidance for this selection.	Evaluator Affirmed.
FCS_COP.1/ DataEncryption	AES used in [**CBC, GCM**] mode and cryptographic key sizes [**128 bits, 256 bits**]	GoAnywhere MFT Bouncy	AES-CBC (128-bit, 256-bit)	C1876

SFR	Algorithm in ST	Implementation name	CAVP Alg.	CAVP Cert #
		Castle FIPS Java API	AES-GCM (128-bit, 256-bit)	
FCS_COP.1/ SigGen	For RSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3	GoAnywhere MFT Bouncy Castle FIPS Java API	RSA SigGen RSA SigVer (n = 2048, 3072)	C1876
	For ECDSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6 and Appendix D, Implementing "NIST curves" [**P-256, P-384, P-521**]; ISO/IEC 14888-3, Section 6.4	GoAnywhere MFT Bouncy Castle FIPS Java API	ECDSA SigGen ECDSA SigVer (Curve = P-256, P-384, P-521)	C1876
FCS_COP.1/ Hash	[**SHA-1, SHA-256, SHA-384, SHA-512**] and message digest sizes [**160, 256, 384, 512**] bits	GoAnywhere MFT Bouncy Castle FIPS Java API	SHA-1 SHA2-256 SHA2-384 SHA2-512	C1876
FCS_COP.1/ KeyedHash	[**HMAC-SHA-1, HMAC-SHA- 256, HMAC-SHA-384, HMAC-SHA-512**] and cryptographic key sizes [**256-bits, 160-bits, 384-bits, 512-bits**] and message digest sizes [**160, 384, 512**] bits	GoAnywhere MFT Bouncy Castle FIPS Java API	HMAC-SHA-1 HMAC-SHA2-256 HMAC-SHA2-384 HMAC-SHA2-512	C1876
FCS_RBG_EXT.1	**CTR_DRBG (AES)**	GoAnywhere MFT Bouncy Castle FIPS Java API	Counter DRBG (AES)	C1876

End of Document