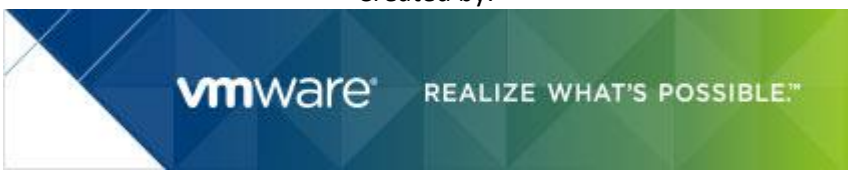
Common Criteria
NDcPP for NSX-T Data Center

AGD – Configuration Guidance
(Version 3.1, Date June-2022)

Created by:



[THE REMAINDER OF THIS PAGE IS INTENTIONALLY LEFT BLANK]

1 Overview

1.1 TOE Overview

The Target of Evaluation (TOE) is the VMware NSX-T Data Center, software-only network virtualization platform that programmatically provisions and manages virtual networks through software network devices and provides network overlay for virtual environments.

In much the same way the server virtualization programmatically creates, snapshots, deletes and restores software-based virtual machines (VMs), NSX-T network virtualization programmatically creates, and deletes software-based virtual networks.

With network virtualization, NSX-T reproduces Layer 2 through Layer 7 networking services (for example, bridging, switching, routing, firewalling, load balancer) in software. As a result, these services can be programmatically assembled to produce unique, isolated virtual networks in a matter of seconds. For more details regarding the TOE, refer to the Security Target.

1.2 TOE Description

The TOE is a VMware network device software product that provides and manages virtual networking components. The TOE is designed as a network virtualization platform, providing the ability to implement and virtualize networks across multiple ESXi nodes and virtual machines (VMs) while providing isolation and efficient use of network resources. This allows for implementation of wide array of various data center workloads or on-demand infrastructure.

For the purpose of testing of the identified TOE, the evaluated TOE configuration is the following:

VMware NSX-T Data Center 3.1 on hypervisor VMware ESXi 6.7 running Ubuntu 18.04 on Dell Power Edge R740 with Intel Xeon Gold 6230R Processor (Cascade Lake).

TOE supports both local and remote administration. The TOE provides a local console which supports CLI. REST API over TLS is the remote interface.

1.3 Assumptions

a

The following assumptions are made with regards to the setup, installation, and ongoing operation of the TOE:

- The TOE has not been compromised prior to installation
- The TOE is assumed to be physically protected in its operational environment and not subject to physical attacks
- The IT environment specified in the Security Target is assumed to be properly implemented by a trained, trusted, and competent administrator:
    - The administrator uses only designated interfaces to manage the TOE
    - The administrator regularly installs software updates per VMware's guidance
    - The administrator follows guidance throughout this document to properly maintain the TOE's security policies
- The IT environment prevents willfully negligent or hostile actions from an administrator

1.4 TOE Delivery

VMware provides customers with the ability to retrieve the NSX-T software package by accessing the public VMware site. The steps below describe how to retrieve the TOE:

1. Visit https://my.vmware.com
2. Log into account or register for one if accessing for the first time
3. Navigate to Products > All Products and Programs > select All Programs
    a. Scroll down to "Networking & Security"
    b. Select "View Download Components" for VMware NSX-T Data Center
4. Locate the version specified in the Security Target, and download the product
    a. Selecting "Read More" will provide additional information regarding the product such as a SHA-256 checksum that can be used for verification

2 Software Download, Installation And Update

2.1 Download

2.1.1 Download VMware NSX-T Data Center (UA and Edge ova)

1. Go to Download VMware NSX-T Data Center. Select release (eg. 3.X).
2. Under Product Downloads tab, locate VMware NSX-T Data Center 3.1.3, then click Go To Download.
3. Under Download Product page, locate NSX-T Manager for VMware ESXi, then, Download Now.
4. Similarly, locate NSX-T Edge for VMware ESXi, then click Download Now

The site provides a checksum to validate the package once it has been downloaded.

2.2 Install and Deploy

2.2.1 Deploy NSX-T UA (Unified Appliance) OVA Package

The NSX-T UA (Unified Appliance) can be deployed as a virtual machine on a vSphere ESXi 6.7 hypervisor. Refer to VMware Documentation NSX-T Data Center Installation Guide for detailed deployment workflow with vCenter Note that "software integrity check" checkbox must be selected during the deployment.
User may also deploy NSX-T UA or edge with VMWare command-line utility **ovftool**. This utility can be downloaded from VMWare customer portal. For example, https://my.vmware.com/web/vmware/details?downloadGroup=OVFTOOL420&productId=614. Execute the following commands to deploy UA:

```
ovftool --name=<UA's name appears in ESX host> \
        --X:injectOvfEnv \
        --X:logFile=ovftool.log \
        --allowExtraConfig \
        --datastore=<name of datastore in which the UA to be deployed> \
        --net:"Network 1"="<mgmt network in the ESX host>" \
```

```
    --noSSLVerify \
    --diskMode=thin \
    --deploymentOption=small \
    --overwrite \
    --powerOn \
    --prop:nsx_role="NSX Manager" \
    --prop:nsx_ip_0=<UA's mgmt intf IP. > \
    --prop:nsx_netmask_0=<UA's mgmt intf net mask. > \
    --prop:nsx_gateway_0=<UA's mgmt intf gateway. > \
    --prop:nsx_dns1_0=<DNS server. > \
    --prop:nsx_domain_0=<UA's search domain. > \
    --prop:nsx_ntp_0=<NTP server. > \
    --prop:nsx_swIntegrityCheck=True
    --prop:nsx_isSSHEnabled=True \
    --prop:nsx_allowSSHRootLogin=True \
    --prop:nsx_passwd_0=<UA's root password, use \ to precede special characters> \
    --prop:nsx_cli_passwd_0=<UA's admin password, use \ to precede special characters> \
    --prop:nsx_hostname=<DNS server. > \
<absolute path of the UA's ova template> \
    vi://root:<ESX host root password>@<ESX host IP>
```

Complete the following steps to create the topology:

1. Once the UA and edge are deployed, powered on, use the following API to retrieve the "detailed_cluster_status". The "overall_status" must be "STABLE" before proceeding to the next step
   GET /api/v1/cluster/status
   Then, import the NSX-T license key to the UA with the following API:
   POST /api/v1/licenses -d '{"license_key": "<license provided by VMware>"}'
   Obtain the thumbprint for port 443 with API:
   GET /api/v1/cluster/nodes/self

2. Log into the CLI of the edge node with user 'admin', execute:
   join management-plane <UA ip> username admin thumbprint <thumbprint obtained above>
   note down the edge node ID <node_id>
3. Create an overlay transport zone with API:
   POST /api/v1/transport-zones -d '{"host_switch_name": "<host switch name>", "transport_type": "OVERLAY", "display_name": "<TZ name>"}'
   and note down the transport zone uuid <tz_id>
4. Create an uplink profile with API
   POST /api/v1/host-switch-profiles -d '{"mtu": 1600, "teaming": {"policy": "FAILOVER_ORDER", "active_list": [{"uplink_name":"uplink-1", "uplink_type": "PNIC"}]},"transport_vlan": "0", "resource_type": "UplinkHostSwitchProfile", "display_name": "<profile name>"}'
   and note down the profile uuid <profile_id>
5. Create an IP pool with API:
   POST /api/v1/pools/ip-pools -d '{"display_name": "<ip pool name>", "subnets": [{"allocation_ranges": [{"start": "<starting addr>" ,"end": "<ending addr>"}], "cidr": "<subnet CIDR>", "gateway_ip": "<gateway IP>"}]}'
   and note down the IP pool uuid <pool_id>

6. Create an edge transport node with API:
   PUT /api/v1/transport-nodes/<node_id> -d '{"resource_type": "TransportNode", "display_name": "<name>", "id": "<node_id>", "node_id": "<node_id>", "host_switch_spec": { "resource_type": "StandardHostSwitchSpec", "host_switches": [{ "host_switch_profile_ids": [{"key": "UplinkHostSwitchProfile", "value": "<profile_id>"}], "host_switch_name": "<host switch name>", "pnics": [{ "device_name": "fp-eth1", "uplink_name": "uplink-1" }], "ip_assignment_spec": {"resource_type": "StaticIpPoolSpec", "ip_pool_id": "<pool_id>"}}]}, "transport_zone_endpoints": [{ "transport_zone_id": "<tz_id>"}], "_revision": 0}'
7. Obtain the ID associated with “nsx-default-edge-high-availability-profile” <cluster profile id>, with API:
   GET /api/v1/cluster-profiles
8. Create an edge cluster (of a single edge node) with API:
   POST /api/v1/edge-clusters -d '{"display_name": "<cluster name>", "cluster_profile_bindings": [{"profile_id": "<cluster profile id>", "resource_type": "EdgeHighAvailabilityProfile"}], "members": [{"transport_node_id": "<edge node id>"}]}'

2.2.2 Deploy NSX-T Edge OVA Package

The NSX-T Edge node can be deployed as a virtual machine on a vSphere ESXi 6.7 hypervisor. Refer to VMware Documentation NSX-T Data Center Installation Guide for detailed deployment workflow with vCenter Note that “software integrity check” checkbox must be selected during the deployment. Execute the following command to deploy edge:

```
ovftool --deploymentOption=medium \
    --name=<edge’s name appears in ESX host> \
    --X:injectOvfEnv \
    --X:logFile=ovftool.log \
    --allowExtraConfig \
    --datastore=<name of datastore in which the edge to be deployed> \
    --net:"Network 0"="<mgmt network in ESX host>" \
    --net:"Network 1"="<data network in ESX host>" \
    --net:"Network 2"="<data network in ESX host>" \
    --net:"Network 3"="<data network in ESX host>" \
    --noSSLVerify \
    --diskMode=thin \
    --powerOn \
    --prop:nsx_passwd_0=<edge’s root password, use \ to precede special characters> \
    --prop:nsx_cli_passwd_0=<edge’s admin password, use \ to precede special characters > \
    --prop:nsx_hostname=<edge’s host name> \
    --prop:nsx_gateway_0=<edge’s mgmt intf gateway. > \
    --prop:nsx_ip_0=<edge’s mgmt intf IP. > \
    --prop:nsx_netmask_0=<edge’s mgmt intf net mask. > \
    --prop:nsx_dns1_0=<DNS server. > \
    --prop:nsx_domain_0=<edge’s search domain. > \
    --prop:nsx_ntp_0=<NTP server. > \
    --prop:nsx_swIntegrityCheck=True
    --prop:nsx_isSSHEnabled=True \
    <absolute path of the edge’s ova template> \
```

vi://root:<ESX host root password>@<ESX host IP>

2.2.3 TOE Configuration

Complete the following steps to create the aforesaid topology:

9. Once the UA and edge are deployed, powered on, use the following API to retrieve the "detailed_cluster_status". The "overall_status" must be "STABLE" before proceeding to the next step:
   GET /api/v1/cluster/status
   Then, import the NSX-T license key to the UA with the following API:
   POST /api/v1/licenses -d '{"license_key": "<license provided by VMware>"}'
   Obtain the thumbprint for port 443 with API:
   GET /api/v1/cluster/nodes/self
10. In the console, log into the CLI of the edge node with user 'admin', execute:
    join management-plane <UA ip> username admin thumbprint <thumbprint obtained above>
    note down the edge node ID <node_id>
11. Create an overlay transport zone with API:
    POST /api/v1/transport-zones -d '{"host_switch_name": "<host switch name>", "transport_type": "OVERLAY", "display_name": "<TZ name>"}'
    and note down the transport zone uuid <tz_id>
12. Create an uplink profile with API
    POST /api/v1/host-switch-profiles -d '{"mtu": 1600, "teaming": {"policy": "FAILOVER_ORDER", "active_list": [{"uplink_name":"uplink-1", "uplink_type": "PNIC"}]},"transport_vlan": "0", "resource_type": "UplinkHostSwitchProfile", "display_name": "<profile name>"}'
    and note down the profile uuid <profile_id>
13. Create an IP pool with API:
    POST /api/v1/pools/ip-pools -d '{"display_name": "<ip pool name>", "subnets": [{"allocation_ranges": [{"start": "<starting addr>" ,"end": "<ending addr>"}], "cidr": "<subnet CIDR>", "gateway_ip": "<gateway IP>"}]}'
    and note down the IP pool uuid <pool_id>
14. Create an edge transport node with API:
    PUT /api/v1/transport-nodes/<node_id> -d '{"resource_type": "TransportNode", "display_name": "<name>", "id": "<node_id>", "node_id": "<node_id>", "host_switch_spec": { "resource_type": "StandardHostSwitchSpec", "host_switches": [{ "host_switch_profile_ids": [{"key": "UplinkHostSwitchProfile", "value": "<profile_id>"}], "host_switch_name": "<host switch name>", "pnics": [{ "device_name": "fp-eth1", "uplink_name": "uplink-1" }], "ip_assignment_spec": {"resource_type": "StaticIpPoolSpec", "ip_pool_id": "<pool_id>"}}]}, "transport_zone_endpoints": [{ "transport_zone_id": "<tz_id>"}], "_revision": 0}'
15. Obtain the ID associated with "nsx-default-edge-high-availability-profile" <cluster profile id>, with API:
    GET /api/v1/cluster-profiles
16. Create an edge cluster (of a single edge node) with API:
    POST /api/v1/edge-clusters -d '{"display_name": "<cluster name>", "cluster_profile_bindings": [{"profile_id": "<cluster profile id>", "resource_type": "EdgeHighAvailabilityProfile"}], "members": [{"transport_node_id": "<edge node id>"}]}'

2.3 Upgrade

Follow the procedures below to carry out the NSX upgrade. The below procedure is in line with the SFR FPT_TUD_EXT SFRs:

1. First obtain the manager upgrade bundle (MUB) from VMware, and place the MUB in the location accessible with a URL
2. Check Current NSX Version
   GET /node/version
3. Place the upgrade MUB to a proper location so that it has an accessible URL, such as http://<server>/MUB/VMware-NSX-upgrade-bundle-<version>.mub
4. Upload the MUB with
   POST /api/v1/upgrade/bundles -d '{"url": "<MUB URL>"}'
   Mark down the upload ID in the API output
5. Periodically check the upload status with
   GET /api/v1/upgrade/bundles/<upload ID>/upload-status
   Until you see: "percent" : 100.0, "status" : "SUCCESS", "detailed_status" : "Upgrade Bundle retrieved successfully"
   If you don't see these three conditions together after a prolong period (such as 30 minutes, depending on upload speed), contact VMware. If the upgrade bundle has an invalid signature, the upgrade process would fail in this step.
6. Accept upgrade end user agreement
   POST /api/v1/upgrade/eula/accept
7. Upgrade the upgrade coordinator and periodically check status until the upgrade coordinator is fully completed, such as "state" : "SUCCESS", "status" : "Upgrade-coordinator has been upgraded"
   POST /api/v1/upgrade?action=upgrade_uc
   GET /api/v1/upgrade/uc-upgrade-status
8. Review component upgrade status with
   GET /api/v1/upgrade/upgrade-unit-groups-status
   At this point the upgrade status for the NSX components should be NOT_STARTED
9. Start upgrade the system with
   POST /api/v1/upgrade/plan?action=start
10. Periodically check upgrade group status with the API in step 8, until the upgrade of one component reaches 100%, then use the following API to continue the upgrade for the next component
    POST /api/v1/upgrade/plan?action=continue
11. Repeat step 10 until all components, including MP (i.e. unified appliance), are upgraded. There would be a period of inaccessibility of UA while it is going through a shutdown cycle
12. Verify the version of the newly upgrade UA with API
    GET /node/version
13. Verify that “overall_upgrade_status”: “SUCCESS” is shown with API
    GET /api/v1/upgrade/status-summary

3 TOE Access

3.1 Configure REST API for TOE Remote Access

3.1.1 Configure REST API for Remote Access

No particular configuration is needed for REST API access. By default, the HTTP service in the NSX-T Unified Appliance is enabled.

3.2 TOE Remote Access via REST API

The most common REST API clients used are curl and postman. For this evaluation, the curl tool was used, although other clients may also be used. A typical NSX-T API command in curl is illustrated in the following example:

curl -u '<user:password>' -k -H 'Content-Type: application/json' -X <GET|POST|PUT|DELETE> https://<UA's IP addr or FQDN>/api/v1/<specific API> -d '<API body>'

Where the <API body> is in json format. For example, to create a transport zone:

curl –u ' admin:password ' -k –H 'Content-Type: application/json' -X POST https://nsx-ua.domain.com/api/v1/transport-zones -d '{"host_switch_name":"nsxswitch","transport_type":"OVERLAY","display_name":"TZ_name"}'

For a full NSX-T API guide, log into the web UI of the NSX-T UA. Click the (?) icon located at the upper right corner, and select "API Documentation".

4 In-Scope vs Out-of-Scope Components and Compliance Configuration

4.1 Non-TOE Hardware/Software/Firmware

The following components are not within the TOE Boundary and are located in the TOE environment:

- Syslog (Audit) Server (rsyslogd 8.20 was used in the evaluated configuration)
- VMware ESXi 6.7
- NSX Agent software (installed on the ESXi hypervisor)
- NSX and vSphere Distributed Switches (NVDS/VDS)
- Certificate Authority Server (CA) (XCA 2.1.0 was used in the evaluated configuration)

4.2 NSX-T Components and Features

4.2.1 In-Scope

The components that are in-scope for this evaluation include the NSX-T Unified Appliance and the NSX-T Edge.

4.2.2 Out-Of-Scope

The following features are not part of the core functionality, and are not included in the scope of the evaluation:

- Unified Appliance clustering is not restricted; however, it is not evaluated.
- Any integration and/or communication with authentication servers such as vIDM is not evaluated.
- The use of SNMPv3 for monitoring is not restricted; however, it is not evaluated.
- Synchronization with an external NTP server is not restricted; however, this functionality is not evaluated.
- The Log Insight interface (an alternative Audit Server) is not enabled by default and is not evaluated.
- CLI (using SSH communications) is not enabled by default and is not evaluated.
- The TOE's debug mode is not intended for normal use and is not evaluated.
- Public and Hybrid Cloud functionality is not enabled by default; and is not evaluated.
- Container functionality is not enabled by default; and is not evaluated.
- The intra-TOE TLS connection between the UA and the NSX-T Edge is not evaluated and an Edge platform residing on another ESXi is not evaluated.
- NSX-T Edge Data-plane Services (or NVDS/VDS) and NSX Agents are excluded from the TOE; and are not evaluated.
- vCenter Server

4.3 Cryptographic Protocols and Modules/Algorithms

4.3.1 In-Scope Cryptographic Protocols

TLS

The TOE uses TLS 1.2 default with approved cipher suites as specified in the Security Target, namely

- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288
- TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268
- TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288
- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268

NSX also supports TLS 1.1 between NSX API server and external API client. In this case users can select a supported TLS version by using API "PUT /api/v1/cluster/api-service".

NSX rejects TLS versions lower than TLS 1.1.

The TOE establishes session keys for TLS sessions using RSA with 2048 and 3072 bit keys and ECDHE curves secp256r1, secp384r1 and secp521r1. The TOE does not support configuration of key/curve used during key agreement.

The TOE does not support session resumption or session tickets.

4.3.2 In-Scope Cryptographic Modules/Algorithms

OpenSSL

The TOE uses VMware's OpenSSL FIPS Object Module for TLS/HTTPS, key stores, and trust stores. Details regarding the algorithms included in this module are noted in the TOE's Security Target along with the relevant cryptographic algorithm certificates.

Bouncy Castle

The TOE also includes VMware BC-FJA (Bouncy Castle FIPS Java API) module for Java based implementations of TLS/HTTPS, key stores, and trust stores. The Security Target includes details regarding the cryptographic algorithms supported by this module along with its CAVP certificates.

4.4 Compliance Configuration for TOE

4.4.1 Enable REST API Interface

The REST API is enabled by default; no specific command needed to enable REST API interface

4.4.2 Time Configuration

Use the following API to configure the date and time:
POST /api/v1/node?action=set_system_time -d '{"system_datetime": "<new date and time>"}'

For example:
POST /api/v1/node?action=set_system_time -d '{"system_datetime": "2020-12-07T17:25:00Z"}'

4.4.3 Automatic Logout for Inactive Remote and Local Authenticated Sessions

The default time period for the automatic logout of inactive authenticated remote http session is 1800 seconds. Users can change this value with API "PUT /api/v1/cluster/api-service", with the detailed steps below:

1. Use API "GET /api/v1/cluster/api-service" to retrieve the json output of the api-service setting. Save this output as a json file say setting.json
2. Edit the "session_timeout" field (in seconds) in the json file to a desired value.
3. Use API "PUT /api/v1/cluster/api-service –d @setting.json" to apply the desired value

The default time period for the automatic logout of inactive authenticated local (I.e. console) CLI session is 600 seconds. Users can change this value with API "PUT /api/v1/node", with the detailed steps below:

1. Use API "GET /api/v1/node" to retrieve the json output of the api-service setting. Save this output as a json file say setting.json
2. Edit the "cli_timeout" field (in seconds) in the json file to a desired value.
3. Use API "PUT /api/v1/node –d @setting.json" to apply the desired value

User Initiated Logout of the TOE

There is no need for explicit logout of the TOE using remote API call as each API call requires independent authentication.

The UI provides a logout button to terminate the http session. For CLI sessions, the user may terminate a session by typing 'exit'.

4.4.4 Login Banner Configuration

Use the following API to configure the login banner:
PUT /api/v1/node -d '{"motd": "<banner message>"}'

4.4.5 Authentication Policy Configuration

Users may configure TOE authentication with API "PUT /api/v1/node/aaa/auth-policy". Authentication policy includes:

- api_failed_auth_lockout_period – the time period the account remains locked out of the API (or HTTP) once such event occurs. Default value is 900 seconds.
- cli_failed_auth_lockout_period – the time period the account remains locked out of the CLI once such event occurs. Default value is 900 seconds.

- api_failed_auth_reset_period – in order to trigger an account lockout, all authentication failures must occur in this period. Default value is 900 seconds.
- api_max_auth_failures – the number of authentication failures that triggers an API lockout. Default value is 5.
- cli_max_auth_failures - the number of authentication failures that triggers an CLI lockout. Default value is 5.
- minimum_password_length – minimum number of characters in a password. Minimum number is 8.

Users may modify these values via REST API "PUT /api/v1/node/aaa/auth-policy –d '{"<parameter>": value}'

4.4.6 Other configurations

Refer to sections “Authentication”, "Audit Log Server” and “Self-Tests” for additional configurations which are required to place the TOE into the CC evaluated configuration.

5 Authentication

The following sections detail authentication methods and configurations employed by the TOE.

5.1 Password Configuration

NSX-T user password should comply with the following rules:
- minimum <8-20 configurable> characters in length
- minimum 1 uppercase character
- minimum 1 lowercase character
- minimum 1 numeric character
- minimum 1 special character including "!”, "@”, "#", “$”, “%”, "^", "&", "*”, "(“, “)"
- default password complexity rules as enforced by the Linux PAM module

Use API to configure NSX-T admin user password:
PUT /api/v1/node/users/10000 -d '{ "old_password": "<old_password>", "password": "<password>"}'

User password for edge node can be changed by edge CLI command set user admin password.

5.2 Key Configuration

The TOE accepts RSA keys of at least 2048 bits, in PEM format.

5.3 X.509 Certificate Configuration

Client and server (end entity) certificates for NSX-T shall comply with the following rules:
- The certificate shall not be expired
- The certificate shall be signed by a valid CA or intermediate CA
- The certificate shall have the X509v3 key usages of “Digital Signature, No Repudiation, Key Encipherment”

- The certificate shall have the X509v3 extended key usages of “TLS Web Server Authentication” and “TLS Web Client Authentication” depending on its use
- The certificate shall include reachable X509v3 CRL distribution points

5.3.1 X.509 Certificate Chain Configuration

When importing a certificate chain to NSX-T as a server (reverse proxy) certificate, users should concatenate all certificates from the end entity to the root CA. When importing a certificate chain to NSX-T as a client certificate (in the case of mutual authentication with remote log servers), users should concatenate the end entity and all intermediate CA certificates with root CA.

5.3.2 X.509 Reference ID Configuration

The TOE supports DNS name identifiers in SAN and CN according to RFC 6125.

5.3.3 Generate X.509 Certificate Signing Request

Use the following API to generate an X.509 CSR:

```
POST /api/v1/trust-management/csrs -d ‘
{
 "subject":{
  "attributes":[
   {"key":"CN","value":"<FQDN>"},
   {"key":"O","value":"<Organization>"},
   {"key":"OU","value":"<Organization Unit"},
   {"key":"C","value":"<Country>"},
   {"key":"ST","value":"<State>"},
   {"key":"L","value":"<Location>"}
  ]
 },
 "key_size":"<2048|3072>", "algorithm":"RSA"
}’
```

5.3.4 Authentication Failure Handling

In the event that an authentication failure occurs, the TOE does not accept the certificate, and rejects the connection. The administrator is required to check the certificates in order to address the issue for the connection to succeed. Audit logs will provide information regarding the failure that has occurred.

5.4 Security Administrator Role

The Security Administrator role is an administrator that includes all the privilege required to perform management tasks and functions.

5.4.1 Description of Security Administrator Role

The Security Administrator is able to perform various functions provided by the TOE. These functions include the following:

- Downloading updates from VMware, and performing a manual update
- Generating, deleting, importing/exporting cryptographic keys
- Configuring the minimum password length
- Changing their own password
- Setting the date and time
- Configuring inactivity times for administrative sessions
- Authenticating the TOE by validating the RSA host key pair
- Configuring audit functions and viewing audit records that are generated
- Configuring the access banner

5.4.2 Managing Security Administrator Role

The Security Administrator role is enabled by default and does not require any additional settings to configure. In order to use the functions provided to the Security Administrator, the individual must be properly authenticated using the correct credentials.

Administrator can use local console, even if locked out on the remote management interface via TLS due to unsuccessful logins. While administration via local console is limited, it provides continuous administrator to the TOE.

5.5 FIA_AFL.1 (Authentication Failure Management)

Administrator can configure the TOE so that 1-5 unsuccessful attempts to login to the TOE will lock the TOE for a duration of 900 seconds.

Administrator can use the following commands on both local console and the REST API as below to set maximum admin authorization failure to 5, upon which it will be locked out for 900 seconds:

```
curl -u admin:${PASSWORD} -i -k -X PUT https://$IP/api/v1/node/aaa/auth-policy -H "Content-
Type:application/json" -d ' { "_schema": "AuthenticationPolicyProperties", "_self": { "href":
"/node/aaa/auth-policy", "rel": "self" }, "api_failed_auth_lockout_period": 900,
"api_failed_auth_reset_period": 900, "api_max_auth_failures": 5,
"cli_failed_auth_lockout_period": 900, "cli_max_auth_failures": 5, "minimum_password_length":
16 }'
```

5.6 FIA_AFL.1 (Authentication Failure Management)

Whitelisting of the local addresses can be done as below:

```
curl -u admin:${PASSWORD} -i -k -X PUT https://$IP/api/v1/cluster/api-service -H "Content-
Type:application/json" -d '{ "global_api_concurrency_limit" : 199,
"basic_authentication_enabled": true, "cipher_suites": [ { "enabled": true, "name":
"TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384" }, { "enabled": true, "name":
"TLS_RSA_WITH_AES_256_GCM_SHA384" }, { "enabled": true, "name":
"TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256" }, { "enabled": true, "name":
"TLS_RSA_WITH_AES_128_GCM_SHA256" }, { "enabled": true, "name":
"TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384" }, { "enabled": true, "name":
"TLS_RSA_WITH_AES_256_CBC_SHA256" }, { "enabled": true, "name":
"TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA" }, { "enabled": true, "name":
"TLS_RSA_WITH_AES_256_CBC_SHA" }, { "enabled": true, "name":
"TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256" }, { "enabled": true, "name":
```

```
"TLS_RSA_WITH_AES_128_CBC_SHA256" }, { "enabled": true, "name":
"TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA" }, { "enabled": true, "name":
"TLS_RSA_WITH_AES_128_CBC_SHA" } ], "protocol_versions": [ { "enabled": true, "name":
"TLSv1.2" } ], "lockout_immune_addresses": [ ] }'
```

```
curl -u admin:${PASSWORD} -i -k -X PUT https://$IP/api/v1/cluster/api-service -H "Content-
Type:application/json" -d '{ "global_api_concurrency_limit" : 199,
"basic_authentication_enabled": true, "cipher_suites": [ { "enabled": true, "name":
"TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384" }, { "enabled": true, "name":
"TLS_RSA_WITH_AES_256_GCM_SHA384" }, { "enabled": true, "name":
"TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256" }, { "enabled": true, "name":
"TLS_RSA_WITH_AES_128_GCM_SHA256" }, { "enabled": true, "name":
"TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384" }, { "enabled": true, "name":
"TLS_RSA_WITH_AES_256_CBC_SHA256" }, { "enabled": true, "name":
"TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA" }, { "enabled": true, "name":
"TLS_RSA_WITH_AES_256_CBC_SHA" }, { "enabled": true, "name":
"TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256" }, { "enabled": true, "name":
"TLS_RSA_WITH_AES_128_CBC_SHA256" }, { "enabled": true, "name":
"TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA" }, { "enabled": true, "name":
"TLS_RSA_WITH_AES_128_CBC_SHA" } ], "protocol_versions": [ { "enabled": true, "name":
"TLSv1.2" } ], "lockout_immune_addresses": [ "1.1.1.1", "2.2.2.2" ] }'
```

Even if the administrator is locked out via REST API, admin can still administer the TOE via local interface.

6 Key destruction

All keys in the volatile memory are considered in use until the TOE is rebooted. All the keys in the non-volatile storage is considered in use until the TOE is re-installed. The below table enumerates the keys in TOE and how they are stored and destroyed upon no use:

Key	Key Size	Generation/Input	Output	Storage	Zeroization	Use
RSA Private Key	2048 - 3072 bits	Internally generated using OpenSSL, or input into system though REST API	Associated cert can be output, but not the private key itself.	File System, keystore	Overwritten by zeros before the disk block is re-allocated	Signature generation, decryption Used by calling application
RSA Public Key	2048 – 3072 bits	Internally generated using OpenSSL, or input into system through REST API	Output via REST API in plaintext (in the form of cert)	File System, keystore	Overwritten by zeros before the disk block is re-allocated	Signature verification, encryption Used by calling application
AES Key	128, 256 bits	Internally generated using API	Never	In RAM	Overwritten by zeros when deleted; or cleared when power cycled or host reboots	Encryption, Decryption

Key	Key Size	Generation/Input	Output	Storage	Zeroization	Use
AES GCM Key	128, 256 bits	Internally generated using API	Never	In RAM	Overwritten by zeros when deleted; or cleared when power cycled or host reboots	Encryption, Decryption, MAC Generation and Verification
HMAC key	160, 224, 256, 384 bits	Internally generated using API	Never	In RAM	Overwritten by zeros when deleted; or cleared when power cycled or host reboots	Message Authentication
CTR_DRBG CSPs	V (128 bits); Key (128, 190, 256 bits); Entropy Input	Internally generated using API	Never	In RAM	Overwritten by zeros when deleted; or cleared when power cycled or host reboots	Random Number Generation

7 Audit Log Storage/Server and Audit Log Records (Auditable Events)

7.1 Audit Log Server

7.1.1 Configure and Usage of Local Storage

Local storage for Java logs is configured in log4j2.xml for each Java web application, and for non-Java applications, the local storage are defined in /etc/rsyslog.d/ directory. In addition to the local daemon/application specific logs, the important INFO logs and all ERROR and above logs will also be forwarded to /var/log/syslog. Log rotation policies for each Java web application are defined in corresponding log4j2.xml, and the rest are defined in the /etc/logrotate.d/ directory.

For example, local storage and rotation policy for proton logs are configured in /opt/vmware/proton-tomcat/conf/log4j2.xml:

```
<RollingRandomAccessFile name="LOGFILE" append="true" fileName="/var/log/proton/nsxapi.log"
filePattern="/var/log/proton/nsxapi.%i.log.gz" >
  <PatternLayout pattern="%d{yyyy-MM-dd'T'HH:mm:ss.SSS'Z'}{UTC} %5p %t %logger{1} - %m%n"/>
  <Policies>
    <SizeBasedTriggeringPolicy size="200 MB" />
  </Policies>
  <DefaultRolloverStrategy max="10" fileIndex="min"/>
</RollingRandomAccessFile>
```

And the configurations for nsx-event.log can be found in /etc/rsyslog.d/08-nsx-event.conf and /etc/logrotate.d/nsx-event:

```
root@junjiex-svc:~# cat /etc/rsyslog.d/08-nsx-event.conf
if $structured-data contains 'eventId' then /var/log/nsx-event.log

root@junjiex-svc:~# cat /etc/logrotate.d/nsx-event
/var/log/nsx-event.log
{
    rotate 10
    size 25k
    missingok
    notifempty
    nocompress
    sharedscripts
    postrotate
        invoke-rc.d rsyslog rotate > /dev/null
        chown root:adm /var/log/nsx-event.log.2
    endscript
}
```

The TOE stores its own syslog events locally on the platform, and can offload events to an audit server protected by TLS. The TOE will stop logging new audit data if the audit local space is full. The TOE does not provide a method of clearing the locally stored records.

When connectivity with the audit server is interrupted, the TOE will continue to store syslog events locally. The TOE will transmit any locally stored contents when connectivity to the syslog server is restored. Otherwise, TOE transmits audit events to syslog server in real-time.

7.1.2 Configure Remote Audit Syslog Exporter

The configuration of a remote logger server depends on specific server type. The administrator may choose a standalone Linux server (Ubuntu for example) to simulate a remote logging server. Use openssl command to listen to port 6514 which NSX-T uses to make TLS connection with the remote logging server:

> openssl s_server -accept 6514 -CAfile <CA cert> [-cert_chain <cert chain>] -cert <server cert> -key <server key>

Remote audit syslog server can be configured via NSX REST API.

Example REST API call is below:

POST -d /api/v1/node/services/syslog/exporters '{"exporter_name": "<exporter's name>", "level": "INFO", "port": 6514, "protocol": "TLS", "server": "<log server's IP/FQDN>", "tls_ca_pem": <server CA>, "tls_client_ca_pem": <client CA>, "tls_cert_pem": <client cert>, "tls_key_pem": <client key>}'

Where the certificates and keys in the API should be in one continuous line, with line breaks are represented by '\n'.

7.1.3 Configure Communication Channel between TOE and Remote Audit Syslog Server

If TLS protocol is used for communication with the remote syslog server, the CA certificates for both server and client, the client certificate, and the client private key are needed to establish the TLS connection.

During the connection establishment, rsyslogd will first check if the encryption cipher and Elliptic Curves presented in the SERVER HELLO message is supported by NSX, then rsyslogd will check the certificate chain presented by the remote server against the server CA certificate to verify if it is trusted. It also performs CRL check to verify the server certificate has not been revoked. Once the server certificate is validated, the NSX node API would then perform the same validation steps on the client certificate against the client CA certificate. The connection will be established only if both server and client certificates are validated.

To disrupt the link between NSX-T and the remote logging server, the evaluator can disable NSX-T's eth0 interface, by Linux shell command:

ip link set eth0 down

and resume the interface by:

ip link set eth0 up

This should be done in the appliance consoles as REST API connection would be lost once eth0 is disabled. Note that by flapping the interface eth0, the default route in the UA would be lost, resulting in connectivity issue with

the external entities. Therefore a “**service networking restart**” is needed to restore the default route, so that the TLS connection could be deleted without error when the test is completed.

7.2 Audit Log Records

7.2.1 Audit Log Record Format

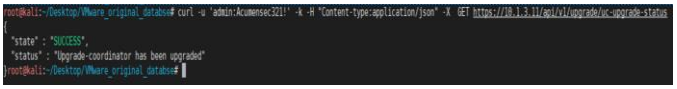
NSX uses RFC 5424 as the standard format for syslog: https://tools.ietf.org/html/rfc5424#section-6.

The RFC-5424 template can be roughly identified as follows:
<facility * 8 + priority>version UTC-TZ hostname APP-NAME procid MSGID [structured-data] msg

Below is a RFC-5424 compliant log:
<190>1 2016-03-15T22:53:00.114Z nsx-manager NSX 6151 FABRIC [nsx@6876 comp="nsx-manager" subcomp="manager" entId="2485b93d-4c1e-423e-8b5a-6952ee9c5dc9" \reqId="2dd35bba-cd25-4d33-a8eb-fe98e8b438e3"] Created Transport Zone/2485b93d-4c1e-423e-8b5a-6952ee9c5dc9

7.2.2 Audit Log Record Examples

Requirement	Auditable Events	Additional Audit Record Contents	Audit Sample audit records
FAU_GEN.1	None	None	
FAU_GEN.2	None	None	
FAU_STG_EXT.1	None	None	
FCS_CKM.1	None	None	
FCS_CKM.2	None	None	
FCS_CKM.4	None	None	
FCS_COP.1/DataEncryption	None	None	
FCS_COP.1/SigGen	None	None	
FCS_COP.1/Hash	None	None	
FCS_COP.1/KeyedHash	None	None	
FCS_RBG_EXT.1	None	None	
FCS_TLSC_EXT.2	None	None	
FCS_TLSS_EXT.1	Failure to establish a TLS Session	Reason for failure	INFO | jvm 1 | 2020/12/14 23:28:10 | https-jsse-nio-10.1.3.11-443-exec-2, fatal error: 40: Client requested protocol TLSv1 not enabled or not supported INFO | jvm 1 | 2020/12/14 23:28:10 | javax.net.ssl.SSLHandshakeException: Client requested protocol TLSv1 not enabled or not supported INFO | jvm 1 | 2020/12/14 23:28:10 | https-jsse-nio-10.1.3.11-443-exec-2, SEND TLSv1.2 ALERT: fatal, description = handshake_failure INFO | jvm 1 | 2020/12/14 23:28:10 | https-jsse-nio-10.1.3.11-443-exec-2, WRITE: TLSv1.2 Alert, length = 2 INFO | jvm 1 | 2020/12/14 23:28:10 | https-jsse-nio-10.1.3.11-443-exec-2, fatal: engine already closed. Rethrowing javax.net.ssl.SSLHandshakeException: Client requested protocol TLSv1 not enabled or not supported
FCS_TLSC_EXT.1	Failure to establish a TLS Session	Failure to establish a TLS Session	2020-12-15T20:27:33.993850+00:00 UA rsyslogd - - - SSL_ERROR_SSL Error in 'osslHandshakeCheck Client': 'error:00000001:lib(0):func(0):reason(1)(1)' with ret=-1 [v8.2002.0]

			2020-12-15T20:27:33.993965+00:00 UA rsyslogd - - - OpenSSL Error Stack: error:1425F102:SSL routines:ssl_choose_client_version:unsupported protocol [v8.2002.0] 2020-12-15T20:27:33.999414+00:00 UA rsyslogd - - - cannot connect to 10.1.1.205:6514: Connection refused [v8.2002.0 try https://www.rsyslog.com/e/2027]
FIA_AFL.1	Unsuccessful login attempts limit is met or exceeded.	Origin of the attempt (e.g., IP address).	2022-03-29T14:23:52.835Z UA NSX 19408 - [nsx@6876 comp="nsx-manager" errorCode="MP403" level="ERROR" subcomp="http"] The credentials were incorrect or the account specified has been locked.
FIA_PMG_EXT.1	None	None	
FIA_UIA_EXT.1	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address)	2020-12-10T19:52:24.303Z UA NSX 24992 - [nsx@6876 comp="nsx-manager" errorCode="MP60206" level="ERROR" subcomp="http"] pam_authenticate failed: Authentication failure rc: 7 2020-12-10T19:52:24.304Z UA NSX 24992 SYSTEM [nsx@6876 audit="true" comp="nsx-manager" level="INFO" subcomp="http"] UserName="admin@192.168.254.181", ModuleName="ACCESS_CONTROL", Operation="LOGIN", Operation status="failure" 2020-12-10T20:03:51.992Z UA NSX 24992 SYSTEM [nsx@6876 audit="true" comp="nsx-manager" level="INFO" subcomp="http"] UserName="admin@192.168.254.181", ModuleName="ACCESS_CONTROL", Operation="LOGIN", Operation status="success"
FIA_UAU_EXT.2	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address)	2020-12-10T19:52:24.303Z UA NSX 24992 - [nsx@6876 comp="nsx-manager" errorCode="MP60206" level="ERROR" subcomp="http"] pam_authenticate failed: Authentication failure rc: 7 2020-12-10T19:52:24.304Z UA NSX 24992 SYSTEM [nsx@6876 audit="true" comp="nsx-manager" level="INFO" subcomp="http"] UserName="admin@192.168.254.181", ModuleName="ACCESS_CONTROL", Operation="LOGIN", Operation status="failure" 2020-12-10T20:03:51.992Z UA NSX 24992 SYSTEM [nsx@6876 audit="true" comp="nsx-manager" level="INFO" subcomp="http"] UserName="admin@192.168.254.181", ModuleName="ACCESS_CONTROL", Operation="LOGIN", Operation status="success"
FIA_UAU.7	None	None	
FMT_MOF.1/ManualUpdate)	Any attempt to initiate a manual update	None	
FMT_MTD.1/CoreData	None	None	
FMT_SMF.1	All management activities of TSF data	None	<182>1 2020-12-10T13:28:05.481Z UA NSX 28842 - [nsx@6876 comp="nsx-manager" subcomp="cli" username="admin" level="INFO" audit="true"] CMD: set user admin password <password-obfuscated> (duration: 16.131s), Operation status: CMD_EXECUTED 2022-04-21T19:09:20.277Z UA2 NSX 17436 - [nsx@6876 comp="nsx-manager" subcomp="cli" username="admin" level="INFO" audit="true"] CMD: set cli-timeout 60 (duration: 1.068s), Operation status: CMD_EXECUTED 2020-12-03T14:58:15.530Z UA NSX 3184 - [nsx@6876 comp="nsx-manager" subcomp="cli" username="admin" level="INFO" audit="true"]

			CMD: set logging-server 10.1.1.205 proto tls level debug serverca RootCA.crt clientca RootCA2.crt certificate client.crt key client.pem (duration: 13.924s), Operation status: CMD_EXECUTED
FMT_SMR.2	None	None	
FPT_SKP_EXT.1	None	None	
FPT_APW_EXT.1	None	None	
FPT_TST_EXT.1	None	None	
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	None.	The log below shows the initiation of the update.  The log below shows a successful update.  The log below shows an unsuccessful update. 2020-12-14T14:40:49.880Z UA NSX 31590 SYSTEM [nsx@6876 comp="nsx-manager" level="INFO" subcomp="upgrade-coordinator"] upgradebundle status is http://10.1.1.205/VMware-NSX-upgrade-bundle-3.1.1.0.0.42176651.mub, FAILED, 100.0, signature_not_verified 2020-12-14T14:40:49.881Z UA NSX 31590 SYSTEM [nsx@6876 comp="nsx-manager" level="INFO" subcomp="upgrade-coordinator"] Status of upgrade bundle 3110042176651 is FAILED[100.0 %]-Signature check of Main Upgrade Bundle(mub) file failed. Tampering with downloaded bundle is not supported.
FPT_STM_EXT.1	Discontinuous changes to time - either Administrator actuated or changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1)	For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).	<182>1 2020-12-07T17:25:00.026Z UA NSX 30172 - [nsx@6876 comp="nsx-manager" subcomp="node-mgmt" username="admin" level="INFO" audit="true"] admin 'POST /api/v1/node?action=set_system_time --- New value: {"system_datetime": "2020-12-07T17:25:00Z"}' 200 203 "" "curl/7.65.3" - 10670.726170
FTA_SSL_EXT.1 (if "terminate the session is selected)	The termination of a local interactive session by the session locking mechanism.	None.	2022-04-21T19:10:32.115Z UA2 NSX 17436 - [nsx@6876 comp="nsx-manager" subcomp="cli" username="admin" level="INFO"] NSX CLI stopped due to inactivity timeout for user: admin
FTA_SSL.3	The termination of	None	

	a remote session by the session locking mechanism.		<182>1 2020-12-11T22:04:24.208Z UA NSX 9380 SYSTEM [nsx@6876 audit="true" comp="nsx-manager" level="INFO" subcomp="http"] UserName="admin@192.168.254.181", ModuleName="ACCESS_CONTROL", Operation="LOGOUT", Operation status="success"
FTA_SSL.4	The termination of an interactive session.	None	182>1 2020-12-11T22:04:24.208Z UA NSX 9380 SYSTEM [nsx@6876 audit="true" comp="nsx-manager" level="INFO" subcomp="http"] UserName="admin@192.168.254.181", ModuleName="ACCESS_CONTROL", Operation="LOGOUT", Operation status="success" 2022-04-04T13:13:45.611Z UA2.acumensec.local NSX 7947 - [nsx@6876 comp="nsx-manager" subcomp="cli" username="admin" level="INFO"] NSX CLI stopped for user: admin
FTA_TAB.1	None	None	
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.	The logs below show the Initiation of a trusted channel. <182>1 2020-12-04T21:18:14.589Z UA NSX 13602 SYSTEM [nsx@6876 audit="true" comp="nsx-manager" level="INFO" subcomp="http"] UserName="admin@192.168.254.181", ModuleName="ACCESS_CONTROL", Operation="LOGIN", Operation status="success" <182>1 2020-12-04T21:18:14.594Z UA NSX 32303 - [nsx@6876 audit="true" comp="nsx-manager" level="INFO" reqId="26a50050-1822-4df9-875d-a83659558505" subcomp="manager" update="true" username="admin"] UserName="admin", ModuleName="AAA", Operation="GetUserFeaturePermissions", Operation status="success" The logs below show the termination of a trusted channel. <182>1 2020-12-04T21:18:27.631Z UA NSX 13602 SYSTEM [nsx@6876 audit="true" comp="nsx-manager" level="INFO" subcomp="http"] UserName="admin@192.168.254.181", ModuleName="ACCESS_CONTROL", Operation="LOGOUT", Operation status="success" <182>1 2020-12-07T16:27:35.138Z UA NSX 28968 - [nsx@6876 comp="nsx-manager" subcomp="cli" username="admin" level="INFO" audit="true"] CMD: set logging-server 10.1.1.205 proto tls level debug serverca RootCA.crt clientca RootCA2.crt certificate client.crt key client.pem (duration: 13.885s), Operation status: CMD_EXECUTED
FTP_TRP.1/Admin	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions.	None.	The logs below show the initiation of a trusted path. <182>1 2020-12-11T22:41:15.764Z UA NSX 9380 SYSTEM [nsx@6876 audit="true" comp="nsx-manager" level="INFO" subcomp="http"] UserName="admin@10.1.1.205", ModuleName="ACCESS_CONTROL", Operation="LOGIN", Operation status="success" The logs below show the termination of a trusted path. <182>1 2020-12-11T22:43:59.245Z UA NSX 9380 SYSTEM [nsx@6876 audit="true" comp="nsx-manager" level="INFO" subcomp="http"] UserName="admin@10.1.1.205", ModuleName="ACCESS_CONTROL", Operation="LOGOUT", Operation status="success"

FIA_X509_EXT.1/Rev	Unsuccessful attempt to validate a certificate Any addition, replacement or removal of trust anchors in the TOE's trust store	Reason for failure of certificate validation Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store	The logs below show a verification failure. 2020-12-04T18:46:04.949Z UA NSX 32303 SYSTEM [nsx@6876 comp="nsx-manager" level="WARNING" reqId="a3bac851-f4f7-4e02-96cd-0519316c94c9" subcomp="manager" username="admin"] Couldn't read CRL from http://10.1.1.205/ica2crl.pem 2020-12-04T18:46:04.950Z UA NSX 30172 - [nsx@6876 comp="nsx-manager" subcomp="node-mgmt" username="root" level="ERROR" errorCode="NOD110"] Certificate trust check failed. status: 200, result: {'status': 'REJECTED', 'error_message': "Certificate was rejected: CRL check failed: Couldn't read CRL from http://10.1.1.205/ica2crl.pem"} 2020-12-04T18:46:04.951Z UA NSX 30172 - [nsx@6876 comp="nsx-manager" subcomp="node-mgmt" username="admin" level="ERROR" errorCode="NOD110"] Failed to create certificate PEM file /config/vmware/nsx-node-api/syslog/e007ac5a-f9ed-455d-b410-8855c2662997_cert.pem for logging server 10.1.1.205:6514 The logs below show an unsuccessful connection because the ICA was removed 2020-12-04T15:20:14.792579+00:00 UA rsyslogd - - - cannot connect to 10.1.1.205:6514: Connection refused [v8.2002.0 try https://www.rsyslog.com/e/2027] 2020-12-04T15:21:20.207095+00:00 UA rsyslogd - - - Certificate error at depth: 1 issuer = /C=US/ST=CA/L=PA/O=VMW/OU=NSBU/CN=Len-Root subject = /C=US/ST=CA/L=PA/O=VMW/OU=NSBU/CN=Len-ica1 err 2:unable to get issuer certificate [v8.2002.0] 2020-12-04T15:21:20.207221+00:00 UA rsyslogd - - - OpenSSL Error Stack: error:1416F086:SSL routines:tls_process_server_certificate:certificate verify failed [v8.2002.0] The logs below show that connection was not successful because the TOE failed to validate the certificate which was expired. 2020-12-04T00:58:59.086Z UA NSX 32303 SYSTEM [nsx@6876 comp="nsx-manager" level="WARNING" reqId="db74ac0b-5c89-4472-a743-f88997655dc4" subcomp="manager" username="admin"] Certificate expired for CN=10.1.1.205,OU=CC,O=Acumensec,L=Rockville,ST=MD,C=US 2020-12-04T00:58:59.089416+00:00 UA rsyslogd - - - Error: peer certificate did not pass validation check [v8.2002.0 try https://www.rsyslog.com/e/2090] 2020-12-04T00:58:27.628130+00:00 UA rsyslogd - - - cannot connect to 10.1.1.205:6514: Connection refused [v8.2002.0 try https://www.rsyslog.com/e/2027] Addition of trust anchors to the trust store: 2020-12-03T14:58:15.530Z UA NSX 3184 - [nsx@6876 comp="nsx-manager" subcomp="cli" username="admin" level="INFO" audit="true"] CMD: set logging-server 10.1.1.205 proto tls level debug serverca RootCA.crt clientca RootCA2.crt certificate client.crt key client.pem (duration: 13.924s), Operation status: CMD_EXECUTED
FIA_X509_EXT.2	None	None	

FIA_X509_EXT.3	None	None	

Table 1 – Examples of Audit Log Records

7.3 FCS_COP.1/TLS Cryptographic Operation (TLS encryption/decryption)

All internal communications between NSX-T nodes use TLS 1.2. The following cipher suite will be used for internal communication as well as external REST API communication:

- *TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289*
- *TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289*
- *TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492*
- *TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492*
- *TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289*
- *TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289*
- *TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288*
- *TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246*
- *TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268*
- *TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288*
- *TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246*
- *TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268*

By default the NSX-T Manager supports both TLS 1.1 and TLS 1.2. The TOE REST API implements TLS 1.1, conformant to RFC 4346 and TLS 1.2, conformant to RFC 5246. The TOE supports X509v3 certificate-based authentication.

TLS1.1 can be disabled with the following API:

```
curl -u admin:${PASSWORD} -i -k -X PUT https://$IP/api/v1/cluster/api-
service -H "Content-Type:application/json" -d '{
"global_api_concurrency_limit" : 199, "basic_authentication_enabled": true,
"cipher_suites": [ { "enabled": true, "name":
"TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384" }, { "enabled": true, "name":
"TLS_RSA_WITH_AES_256_GCM_SHA384" }, { "enabled": true, "name":
"TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256" }, { "enabled": true, "name":
"TLS_RSA_WITH_AES_128_GCM_SHA256" }, { "enabled": true, "name":
"TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384" }, { "enabled": true, "name":
"TLS_RSA_WITH_AES_256_CBC_SHA256" }, { "enabled": true, "name":
"TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA" }, { "enabled": true, "name":
"TLS_RSA_WITH_AES_256_CBC_SHA" }, { "enabled": true, "name":
"TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256" }, { "enabled": true, "name":
"TLS_RSA_WITH_AES_128_CBC_SHA256" }, { "enabled": true, "name":
"TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA" }, { "enabled": true, "name":
"TLS_RSA_WITH_AES_128_CBC_SHA" } ], "protocol_versions": [ { "enabled":
true, "name": "TLSv1.2" } ], "lockout_immune_addresses": [ ] }'
```

Users can confirm that TLS1.1 is disabled:

```
curl -u admin:${PASSWORD} -i -k https://$IP/api/v1/cluster/api-service
```

```
"protocol_versions": [
  {
    "enabled": false,
    "name": "TLSv1.1"
  },
```

The NSX Proxy on the NSX-T Edge and ESX host uses TLS 1.2 and is not configurable.

7.4 FCS_COP.1/TLS.HMAC Cryptographic Operation (Hashing)

The NSX-T Manager and NSX-T Edge will use SHA-1, SHA-256, or SHA-384 algorithms for computing hashes and message digest sizes of 160, 256, and 384 bits in cryptographic operations.

7.5 FMT_MTD.1/CryptoKeys (ability to manage keys)

The user can import CA-signed public certs/private keys into the NSX-T Manager. The public certs can then optionally be bound to services (ie. the reverse-proxy). These public certs/private keys can be deleted if they are not bound to a service. No other keys are managed by the administrators. All communication, both internal and external, uses TLS to ensure that only authorized users can gain access to the NSX-T system. Hence, keys associated with CA-signed X509 certs are protected from unauthorized access.

The CA-signed X509 v3 certificates can be imported into the TOE and designated as trust anchor.

8 Self-Tests

8.1 Cryptographic POST (Power-On Self-Test)

The various error messages related to a failed self-test are described in the following sections. In case of error which persists after attempting to reboot to clear the error, the administrator is advised to contact VMware for additional troubleshooting.

8.1.1 OpenSSL POST

Evidence of OpenSSL POST can be found in /var/log/syslog of the UA and Edge node. The sample logs in the UA are:
<182>1 2019-08-22T18:18:22.442Z nsx-controller-leng6 NSX 2656 - [nsx@6876 comp="nsx-manager" subcomp="python" username="root" level="INFO"] FipsPost::OpenSSL FIPS library found, /etc/vmware/system_fips is present.
<182>1 2019-08-22T18:18:22.491Z nsx-controller-leng6 NSX 2656 - [nsx@6876 comp="nsx-manager" subcomp="python" username="root" level="INFO"] FipsPost::FIPS SELFTEST PASS:: SYSTEM is running in FIPS mode.

And the sample logs in the Edge node are:
<182>1 2019-07-12T17:45:49.590Z nsx-edge NSX 1554 - [nsx@6876 comp="nsx-autonomous-edge" subcomp="python" username="root" level="INFO"] FipsPost::OpenSSL FIPS library found, /etc/vmware/system_fips is present.
<182>1 2019-07-12T17:45:49.657Z nsx-edge NSX 1554 - [nsx@6876 comp="nsx-autonomous-edge" subcomp="python" username="root" level="INFO"] FipsPost::FIPS SELFTEST PASS:: SYSTEM is running in FIPS mode.

The above logs can only be observed when the FIPS modes in the appliances are enabled (default setting). The FIPS mode can also be verified by the existence of the indicator file /etc/vmware/system_fips

8.1.2 Bouncy Castle POST

Evidence of Bouncy Castle POST can be found in both UA's /var/log/syslog and /var/log/proton/nsxapi.log (Edge node does not use Bouncy Castle). For example in /var/log/syslog:
<182>1 2019-08-22T18:24:09.828Z nsx-controller-leng6 NSX 9481 SYSTEM [nsx@6876 comp="nsx-manager" level="INFO" subcomp="manager"] FIPS provider (bouncycastle) status is READY.
And in /var/log/proton/nsxapi.log
2019-08-22T18:24:09.828Z INFO coordinationEventsProcessor-1 EntropyReporter - SYSTEM [nsx@6876 comp="nsx-manager" level="INFO" subcomp="manager"] FIPS provider (bouncycastle) status is READY.

8.1.3 Software/Firmware Integrity POST

Evidence of software/firmware integrity check can be seen in /var/log/syslog.

When the integrity check fails in UA or edge, their /var/log/syslog should contain the following entry:
2020-08-03T20:24:36.841351+00:00 nsx-controller-leng6 sw-integrity-checker - - - Software Integrity Check Failed. NSX services will not come up.

When the integrity check is successful, the log entry would be:
2020-08-03T20:01:30.10352+00:00 nsx-controller-leng6 sw-integrity-checker - - - Software Integrity Check Successful

9 References

[1] VMware NSX-T Data Center Installation Guide
[2] VMware NSX-T Data Center Upgrade Guide
[3] VMware NSX-T Data Center Administration Guide

10 Appendix A

Sample Logs for TLS failures:

FCS_TLSC_EXT.1.1/FCS_TLSC_ EXT.2.1 Test #2:
<179>1 2019-08-22T20:43:26.909Z nsx-controller-leng4 NSX 9526 SYSTEM [nsx@6876 comp="nsx-manager" errorCode="MP2002" level="ERROR" reqId="c30aa0b6-f31d-43bd-82bc-f44213a5d4cc" subcomp="manager" username="admin"] Extended key usage field not present in the certificate.

FCS_TLSC_EXT.1.1/FCS_TLSC_ EXT.2.1 Test #3:
<43>1 2019-08-23T17:54:19.359470+00:00 nsx-controller-leng5 rsyslogd - - - SSL_ERROR_SSL in 'osslHandshakeCheck Client' [v8.1901.0]
<43>1 2019-08-23T17:54:19.359812+00:00 nsx-controller-leng5 rsyslogd - - - OpenSSL Error Stack: error:1409017F:SSL routines:ssl3_get_server_certificate:wrong certificate type [v8.1901.0]

FCS_TLSC_EXT.1.1/FCS_TLSC_ EXT.2.1 Test #4:
<43>1 2019-08-23T18:04:11.735175+00:00 nsx-controller-leng5 rsyslogd - - - SSL_ERROR_SSL in 'osslHandshakeCheck Client' [v8.1901.0]
<43>1 2019-08-23T18:04:11.735523+00:00 nsx-controller-leng5 rsyslogd - - - OpenSSL Error Stack: error:140920F8:SSL routines:ssl3_get_server_hello:unknown cipher returned [v8.1901.0]

FCS_TLSC_EXT.1.1/FCS_TLSC_ EXT.2.1 Test #5(1):
<43>1 2019-08-23T18:08:50.907920+00:00 nsx-controller-leng5 rsyslogd - - - SSL_ERROR_SSL in 'osslHandshakeCheck Client' [v8.1901.0]
<43>1 2019-08-23T18:08:50.908383+00:00 nsx-controller-leng5 rsyslogd - - - OpenSSL Error Stack: error:14077102:SSL routines:SSL23_GET_SERVER_HELLO:unsupported protocol [v8.1901.0]

FCS_TLSC_EXT.1.1/FCS_TLSC_ EXT.2.1 Test #5(2):
<43>1 2019-08-23T18:10:58.286475+00:00 nsx-controller-leng5 rsyslogd - - - SSL_ERROR_SSL in 'osslHandshakeCheck Client' [v8.1901.0]
<43>1 2019-08-23T18:10:58.286715+00:00 nsx-controller-leng5 rsyslogd - - - OpenSSL Error Stack: error:1408D07B:SSL routines:ssl3_get_key_exchange:bad signature [v8.1901.0]

FCS_TLSC_EXT.1.1/FCS_TLSC_ EXT.2.1 Test #5(3):
<43>1 2019-08-23T18:15:35.514709+00:00 nsx-controller-leng5 rsyslogd - - - SSL_ERROR_SSL in 'osslHandshakeCheck Client' [v8.1901.0]

<43>1 2019-08-23T18:15:35.515138+00:00 nsx-controller-leng5 rsyslogd - - - OpenSSL Error Stack: error:14092105:SSL routines:ssl3_get_server_hello:wrong cipher returned [v8.1901.0]

FCS_TLSC_EXT.1.1/FCS_TLSC_ EXT.2.1 Test #5(4):
<43>1 2019-08-23T18:18:26.845828+00:00 nsx-controller-leng5 rsyslogd - - - SSL_ERROR_SSL in 'osslHandshakeCheck Client' [v8.1901.0]
<43>1 2019-08-23T18:18:26.846169+00:00 nsx-controller-leng5 rsyslogd - - - OpenSSL Error Stack: error:1408D07B:SSL routines:ssl3_get_key_exchange:bad signature [v8.1901.0]

FCS_TLSC_EXT.1.1/FCS_TLSC_ EXT.2.1 Test #5(5):
<43>1 2019-08-23T18:20:12.108335+00:00 nsx-controller-leng5 rsyslogd - - - SSL_ERROR_SSL in 'osslHandshakeCheck Client' [v8.1901.0]
<43>1 2019-08-23T18:20:12.108661+00:00 nsx-controller-leng5 rsyslogd - - - OpenSSL Error Stack: error:1408C095:SSL routines:ssl3_get_finished:digest check failed [v8.1901.0]

FCS_TLSC_EXT.2.2 Test #1:
<43>1 2019-08-22T20:47:34.014723+00:00 nsx-controller-leng4 rsyslogd - - - error: peer name not authorized - not permitted to talk to it. Names: name: /C=US/ST=CA/L=Palo Alto/O=VMW/OU=NSBU/CN=xyz.eng.vmware.com; [v8.1901.0 try https://www.rsyslog.com/e/2088]

FCS_TLSC_EXT.2.2 Test #1:
<43>1 2019-08-22T20:54:13.038550+00:00 nsx-controller-leng4 rsyslogd - - - error: peer name not authorized - not permitted to talk to it. Names: name: /C=US/ST=CA/L=Palo Alto/O=VMW/OU=NSBU/CN= w1-mvpcloud-219.eng.vmware.com; [v8.1901.0 try https://www.rsyslog.com/e/2088]

FCS_TLSC_EXT.2.2 Test #5(1):
<43>1 2019-08-22T21:17:11.288343+00:00 nsx-controller-leng4 rsyslogd - - - error: peer name not authorized - not permitted to talk to it. Names: name: /C=US/ST=CA/L=Palo Alto/O=VMW/OU=NSBU/CN=w1-mvpcloud-219.*.eng.vmware.com; [v8.1901.0 try https://www.rsyslog.com/e/2088]

FCS_TLSC_EXT.2.2 Test #5(3):
<43>1 2019-08-22T21:23:11.265049+00:00 nsx-controller-leng4 rsyslogd - - - error: peer name not authorized - not permitted to talk to it. Names: name: /C=US/ST=CA/L=Palo Alto/O=VMW/OU=NSBU/CN=eng.vmware.com; [v8.1901.0 try https://www.rsyslog.com/e/2088]

FCS_TLSC_EXT.2.2 Test #5(4):
<43>1 2019-08-22T21:25:36.545337+00:00 nsx-controller-leng4 rsyslogd - - - error: peer name not authorized - not permitted to talk to it. Names: name: /C=US/ST=CA/L=Palo Alto/O=VMW/OU=NSBU/CN=foo.w1-mvpcloud-219.eng.vmware.com; [v8.1901.0 try https://www.rsyslog.com/e/2088]

FCS_TLSC_EXT.2.2 Test #5(5):
<43>1 2019-08-22T21:31:23.893790+00:00 nsx-controller-leng4 rsyslogd - - - error: peer name not authorized - not permitted to talk to it. Names: name: /C=US/ST=CA/L=Palo Alto/O=VMW/OU=NSBU/CN=dummy; [v8.1901.0 try https://www.rsyslog.com/e/2088]

FCS_TLSC_EXT.2.2 Test #5(7):

<43>1 2019-08-22T21:36:24.568743+00:00 nsx-controller-leng4 rsyslogd - - - error: peer name not authorized - not permitted to talk to it. Names: name: /C=US/ST=CA/L=Palo Alto/O=VMW/OU=NSBU/CN=dummy; [v8.1901.0 try https://www.rsyslog.com/e/2088]

FCS_TLSC_EXT.2.2 Test #5(8):
<43>1 2019-08-22T21:40:23.394350+00:00 nsx-controller-leng4 rsyslogd - - - error: peer name not authorized - not permitted to talk to it. Names: name: /C=US/ST=CA/L=Palo Alto/O=VMW/OU=NSBU/CN=dummy; [v8.1901.0 try https://www.rsyslog.com/e/2088]

FCS_TLSC_EXT.2.2 Test #7:
<43>1 2019-08-22T21:57:53.616894+00:00 nsx-controller-leng4 rsyslogd - - - not permitted to talk to peer: certificate validation failed: unable to verify the first certificate [v8.1901.0 try https://www.rsyslog.com/e/2090]

FCS_TLSC_ESX.2.3 Test #2(1):
<43>1 2019-08-22T20:47:34.014723+00:00 nsx-controller-leng4 rsyslogd - - - error: peer name not authorized - not permitted to talk to it. Names: name: /C=US/ST=CA/L=Palo Alto/O=VMW/OU=NSBU/CN=xyz.eng.vmware.com; [v8.1901.0 try https://www.rsyslog.com/e/2088]

FCS_TLSC_ESX.2.3 Test #2(2):
<43>1 2019-08-22T22:07:19.361317+00:00 nsx-controller-leng4 rsyslogd - - - not permitted to talk to peer: certificate validation failed: unable to verify the first certificate [v8.1901.0 try https://www.rsyslog.com/e/2090]

FCS_TLSC_ESX.2.3 Test #2(3):
<180>1 2019-08-22T22:11:15.718Z nsx-controller-leng4 NSX 9526 SYSTEM [nsx@6876 comp="nsx-manager" level="WARN" reqId="c04dae6f-a33d-47b5-99a9-26c482d256a6" subcomp="manager" username="admin"] Certificate expired for CN=w1-mvpcloud-219.eng.vmware.com,OU=NSBU,O=VMW,L=Palo Alto,ST=CA,C=US
<43>1 2019-08-22T22:11:15.723840+00:00 nsx-controller-leng4 rsyslogd - - - Error: peer certificate did not pass validation check [v8.1901.0 try https://www.rsyslog.com/e/2090]

FCS_TLSC_ESX.2.3 Test #2(4):
<180>1 2019-08-22T22:17:40.864Z nsx-controller-leng4 NSX 9526 SYSTEM [nsx@6876 comp="nsx-manager" level="WARN" reqId="8078cab9-23b3-48db-96f0-b3ec541c8df8" subcomp="manager" username="admin"] No CDP specified in the certificate for CN=w1-mvpcloud-219.eng.vmware.com,OU=NSBU,O=VMW,L=Palo Alto,ST=CA,C=US
<43>1 2019-08-22T22:17:40.867938+00:00 nsx-controller-leng4 rsyslogd - - - Error: peer certificate did not pass validation check [v8.1901.0 try https://www.rsyslog.com/e/2090]

FCS_TLSC_ESX.2.4 Test #1
<43>1 2019-08-22T22:21:40.166658+00:00 nsx-controller-leng4 rsyslogd - - - SSL_ERROR_SSL in 'osslHandshakeCheck Client' [v8.1901.0]
<43>1 2019-08-22T22:21:40.166904+00:00 nsx-controller-leng4 rsyslogd - - - OpenSSL Error Stack: error:14077410:SSL routines:SSL23_GET_SERVER_HELLO:sslv3 alert handshake failure [v8.1901.0]

FCS_TLSS_EXT.1.1 Test #2:
INFO | jvm 1 | 2019/08/22 19:18:20 | ***
INFO | jvm 1 | 2019/08/22 19:18:20 | %% Initialized: [Session-1, SSL_NULL_WITH_NULL_NULL]
INFO | jvm 1 | 2019/08/22 19:18:20 | https-jsse-nio-10.143.1.48-443-exec-1, fatal error: 40: no cipher suites in common

INFO | jvm 1 | 2019/08/22 19:18:20 | javax.net.ssl.SSLHandshakeException: no cipher suites in common
INFO | jvm 1 | 2019/08/22 19:18:20 | %% Invalidated: [Session-1, SSL_NULL_WITH_NULL_NULL]
INFO | jvm 1 | 2019/08/22 19:18:20 | https-jsse-nio-10.143.1.48-443-exec-1, SEND TLSv1.2 ALERT: fatal, description = handshake_failure

FCS_TLSS_ EXT.1.1 Test #3:
INFO | jvm 1 | 2019/08/22 19:41:05 | ***
INFO | jvm 1 | 2019/08/22 19:41:05 | %% Initialized: [Session-2, SSL_NULL_WITH_NULL_NULL]
INFO | jvm 1 | 2019/08/22 19:41:05 | https-jsse-nio-10.143.1.48-443-exec-2, fatal error: 40: no cipher suites in common
INFO | jvm 1 | 2019/08/22 19:41:05 | javax.net.ssl.SSLHandshakeException: no cipher suites in common
INFO | jvm 1 | 2019/08/22 19:41:05 | %% Invalidated: [Session-2, SSL_NULL_WITH_NULL_NULL]
INFO | jvm 1 | 2019/08/22 19:41:05 | https-jsse-nio-10.143.1.48-443-exec-2, SEND TLSv1.2 ALERT: fatal, description = handshake_failure

FCS_TLSS_ EXT.1.1 Test #4 (1):
INFO | jvm 1 | 2019/08/22 19:55:46 | ***
INFO | jvm 1 | 2019/08/22 19:55:46 | https-jsse-nio-10.143.1.48-443-exec-1, fatal error: 40: client 'finished' message doesn't verify
INFO | jvm 1 | 2019/08/22 19:55:46 | javax.net.ssl.SSLHandshakeException: client 'finished' message doesn't verify
INFO | jvm 1 | 2019/08/22 19:55:46 | %% Invalidated: [Session-3, TLS_RSA_WITH_AES_128_CBC_SHA]
INFO | jvm 1 | 2019/08/22 19:55:46 | https-jsse-nio-10.143.1.48-443-exec-1, SEND TLSv1.2 ALERT: fatal, description = handshake_failure
INFO | jvm 1 | 2019/08/22 19:55:46 | https-jsse-nio-10.143.1.48-443-exec-1, WRITE: TLSv1.2 Alert, length = 2
INFO | jvm 1 | 2019/08/22 19:55:46 | https-jsse-nio-10.143.1.48-443-exec-1, fatal: engine already closed. Rethrowing javax.net.ssl.SSLHandshakeException: client 'finished' message doesn't verify

FCS_TLSS_ EXT.1.1 Test #4 (2):
INFO | jvm 1 | 2019/08/22 20:01:46 | https-jsse-nio-10.143.1.48-443-exec-5, fatal error: 80: problem unwrapping net record
INFO | jvm 1 | 2019/08/22 20:01:46 | javax.net.ssl.SSLException: Unsupported record version Unknown-0.0
INFO | jvm 1 | 2019/08/22 20:01:46 | %% Invalidated: [Session-6, TLS_RSA_WITH_AES_128_CBC_SHA]
INFO | jvm 1 | 2019/08/22 20:01:46 | https-jsse-nio-10.143.1.48-443-exec-5, SEND TLSv1.2 ALERT: fatal, description = internal_error

FCS_TLSS_ EXT.1.2 Test #2:
INFO | jvm 1 | 2019/08/22 20:14:59 | ***
INFO | jvm 1 | 2019/08/22 20:14:59 | Warning: No renegotiation indication in ClientHello, allow legacy ClientHello
INFO | jvm 1 | 2019/08/22 20:14:59 | https-jsse-nio-10.143.1.48-443-exec-3, fatal error: 40: Client requested protocol SSLv3 not enabled or not supported
INFO | jvm 1 | 2019/08/22 20:14:59 | javax.net.ssl.SSLHandshakeException: Client requested protocol SSLv3 not enabled or not supported
INFO | jvm 1 | 2019/08/22 20:14:59 | https-jsse-nio-10.143.1.48-443-exec-3, SEND TLSv1.2 ALERT: fatal, description = handshake_failure
INFO | jvm 1 | 2019/08/22 20:14:59 | https-jsse-nio-10.143.1.48-443-exec-3, WRITE: TLSv1.2 Alert, length = 2

INFO | jvm 1 | 2019/08/22 20:14:59 | https-jsse-nio-10.143.1.48-443-exec-3, fatal: engine already closed. Rethrowing javax.net.ssl.SSLHandshakeException: Client requested protocol SSLv3 not enabled or not supported

INFO | jvm 1 | 2019/08/22 20:14:59 | javax.net.ssl.SSLHandshakeException: SSLv2Hello is disabled
INFO | jvm 1 | 2019/08/22 20:14:59 | https-jsse-nio-10.143.1.48-443-exec-4, SEND TLSv1.2 ALERT: fatal, description = unexpected_message
INFO | jvm 1 | 2019/08/22 20:14:59 | https-jsse-nio-10.143.1.48-443-exec-4, WRITE: TLSv1.2 Alert, length = 2
INFO | jvm 1 | 2019/08/22 20:14:59 | https-jsse-nio-10.143.1.48-443-exec-4, fatal: engine already closed. Rethrowing javax.net.ssl.SSLHandshakeException: SSLv2Hello is disabled

INFO | jvm 1 | 2019/08/22 20:15:03 | ***
INFO | jvm 1 | 2019/08/22 20:15:03 | Warning: No renegotiation indication in ClientHello, allow legacy ClientHello
INFO | jvm 1 | 2019/08/22 20:15:03 | https-jsse-nio-10.143.1.48-443-exec-3, fatal error: 40: Client requested protocol TLSv1 not enabled or not supported
INFO | jvm 1 | 2019/08/22 20:15:03 | javax.net.ssl.SSLHandshakeException: Client requested protocol TLSv1 not enabled or not supported
INFO | jvm 1 | 2019/08/22 20:15:03 | https-jsse-nio-10.143.1.48-443-exec-3, SEND TLSv1.2 ALERT: fatal, description = handshake_failure
INFO | jvm 1 | 2019/08/22 20:15:03 | https-jsse-nio-10.143.1.48-443-exec-3, WRITE: TLSv1.2 Alert, length = 2
INFO | jvm 1 | 2019/08/22 20:15:03 | https-jsse-nio-10.143.1.48-443-exec-3, fatal: engine already closed. Rethrowing javax.net.ssl.SSLHandshakeException: Client requested protocol TLSv1 not enabled or not supported