Arista Networks 7280 Switches Running EOS 4.28

Common Criteria Guidance Supplement

Version: 2.0

07/18/2023

Prepared By:

Arista Networks, Inc.
5453 Great America Parkway
Santa Clara, CA 95054

Table of Contents

1. Introduction 5

1.1 TOE Type 5

1.2 TOE Usage 5

1.3 TOE Major Security Features Summary 5

1.4 Operational Environment 6

1.4 TOE Description 6

1.5 Console CLI access 8

2. Usage Assumptions 9

2.1 No General-Purpose Computing 9

2.2 Trusted Administrator 9

2.3 Physical Security 9

2.4 Secure Acceptance 9

3. Secure Configuration 9

3.1 Software Image 9

3.2 Hostname, DNS Server, Time Setting, Login Banner and Password Restrictions 11

3.3 Console Idle Timeout 12

3.4 Default Accounts Protection 13

3.5 Disallowed Access Methods (SNMP, Telnet, API) 13

3.5.1 SNMP 14

3.5.2 Telnet 14

3.5.3 API 14

3.6 IP Address 14

3.7 Entropy 14

3.8 SSH Configuration 15

3.9 SSH Tunnel 17

3.9.1 Tunnel Endpoint on the Syslog Server 17

3.9.1.1 Add switch’s public key for authentication to Syslog server 17

3.9.1.2 Generate/Copy public host key of Syslog server 17

3.9.1.3 Enable public key authentication for authuser 17

3.9.1.4 Software settings on Syslog server 17

3.9.2 Tunnel Endpoint on the Switch 17
3.10 Audit Logs Configuration 19
3.11 Named User Accounts 20
3.12 TLS Server 22
3.12.1 Create SSL profile 22
3.12.2 Enable FIPS mode 22
3.12.3 Generate CSR and obtain signed certificate 23
3.12.4 Define cipher suites 25
3.12.5 Configure certificate check restrictions 25
3.12.6 Load certificate chain and CRLs 25
3.12.7 Validate SSL profile configuration 27
3.12.8 Cryptographic key destruction 28
3.12.9 X509 certificate validation 28
3.13 eAPI Server Configuration 29
3.13.1 eAPI Fallback authentication 31
3.14 eAPI Client Operation 31
3.15 Timekeeping 33
4. Auditable Events 33
4.1 FAU_GEN.1.1 - Audit Data Generation 33
4.1.1 Start-up and Shut-Down of Audit Functions 33
4.1.2 Administrative Login and Logout 33
4.1.3 TSF Data Related Configuration Changes 33
4.1.4 Cryptographic Keys 34
4.1.5 Password Reset 37
4.2 FAU_GEN.1.2 - Audit Data Generation 37
4.2.1 FCS_SSHC_EXT.1 - SSH Client Protocol 37
4.2.2 FCS_SSHS_EXT.1- SSH Server Protocol 39
4.2.3 FCS_TLSS_EXT.2 - TLS Server Protocol with Mutual Authentication 40
4.2.4 FIA_AFL.1 - Authentication Failure Management 41
4.2.5 FIA_UIA_EXT.1 - User Identification and Authentication 41
4.2.6 FIA_UIA_EXT.2 - Password-based Authentication Mechanism 41
4.2.7 FIA_X509_EXT.1/Rev 42
4.2.8 FMT_MOF.1/ManualUpdate - Management of Security Functions Behavior 42
4.2.9 FMT_SMF.1 - Specification of Management Functions 43

4.2.10 FPT_TUD_EXT.1 - Trusted Update 44
4.2.11 FPT_STM_EXT.1 - Reliable Time Stamps 44
4.2.12 FPT_SSL_EXT.1 - TSF Initiated Session Locking 45
4.2.13 FPT_SSL.3 - TSF Initiated Session Termination 45
4.2.14 FPT_SSL.4 - User Initiated Termination 45
4.2.15 FPT_ITC.1 - TSF Initiated Session Locking 45
4.2.16 FPT_TRP.1/Admin - Trusted Path 46
4.2.17 Verifying FIPS mode is enabled and the FIPS POST was performed 46
4.2.18 Verifying Integrity of Software and FIPS self-tests on Startup 47

1. Introduction

This Common Criteria Guidance Supplement document pertains to Arista Networks 7280 series switches running EOS 4.28.0. It provides instructions for operation of these switches consistent with the Common Criteria evaluated configuration (CC mode) described in:

- [Security Target] Security Target - “Arista Networks 7280 Switches Running EOS 4.28”.
- [User Manual] User Manual - Arista EOS.
- [Quick Start] Hardware Installation Guides posted under Product Documentation - Hardware section on the Arista website.

1.1 TOE Type

The TOE is classified as a Network Device, that is, a device composed of both hardware and software that is connected to the network and has an infrastructure role within the network.

1.2 TOE Usage

The Arista Networks Data Center and Cloud Computing Switches are networking switches (Network Devices for CC purposes) that provide OSI model Layer 2, 3, and 4 Ethernet interconnectivity and network management services (Data Link, Network, and Transport Layers, respectively). Each model is manufactured with high performance electronics making it ideally suitable for demanding data center environments.

1.3 TOE Major Security Features Summary

- Security Audit
    - Generates audit records, storing them locally and transmitting them to a remote audit server.
    - Supports secure communication to remote syslog-compatible audit servers protected by the SSHv2 Trusted Channel.

- Cryptographic Support
    - Utilization of NIST-specified and CAVP validated cryptographic algorithms for asymmetric key generation, AES encryption/decryption, digital signature generation/verification, hashing, and keyed-hashing (Message Authentication Code).
    - Cryptographic-key Destruction using PP specified methods.
    - Deterministic Random Bit Generation (DRBG).
    - Assurance of seeding the DRBG with sufficient entropy (minimum of 256-bits of entropy).

- Identification and Authentication
    - Administrative password management.
    - Protected authentication data at the local and remote consoles.
    - Identification and authentication of the Security Administrative user.
    - X509 certificate-based authentication and validation.
    - X509 certificate request generation.

- Security Management
    - Trusted Update mechanism.
    - Restriction of TSF management to the Security Administrator.
    - Local and remote administration of the TOE by the Security Administrator.

- Protection of the TSF
    - Protection of stored passwords.
    - Prevention of disclosing passwords via normal management interfaces.
    - Prevention of disclosing private keys via normal management interfaces.
    - Automated self-testing upon boot-up.
    - Querying of the TOE firmware/software.
    - Reliable timestamps.

- TOE Access
    - Session termination (TSF initiated, and User initiated).
    - Display of a warning and consent banner on the local and remote management interfaces prior to authentication.

- Trusted Path/Channels
    - Cryptographically secure path between the TOE and the Security Administrative user for remote management.
    - Cryptographically secure channels between the TOE and authorized IT entities in the Operational Environment to support the TSF.

1.4 Operational Environment

The TOE's Operational Environment must provide the following services to support the secure operation of the TOE:

- Local Console Administrative Access
    - RS-232 Serial Console.
    - VT-100 terminal emulation program.
- Remote Management
    - SSH client for remote interactive session utilizing SSH.
    - eAPI JSON-RPC Client capable of establishing a mutually authenticated TLS session.
- Audit Server
    - Syslog server capable of accepting an SSHv2 tunnel utilizing SSH Protocol Version 2 (SSHv2).
- Certificate Revocation List (CRL) Server
    - Server from where CRLs can be downloaded on TOE to check validity of X509v3 certificates.

1.4 TOE Description

The Arista 7280 series switches are fixed form factor switches. The 7280 series switches range in size between 1 and 2 RU. Models vary in total throughput, port count, port speeds, route table scales etc.

Each switch model runs Arista's Linux-based network operating system called Extensible Operating System (EOS). The same EOS binary image runs on all TOE hardware models. All EOS code is compiled to the same i686 assembly, making it such that no processor runs anything different from any other processor. All processors implement the i686 assembly language. All SFRs in this Security Target are implemented by EOS. Hence, they behave identically on every switch model.

The table below provides the list of appliances across different series:

Table 1:Hardware Appliances

Series	Models	Interfaces	Host CPU
7280CR	● SKN-7280CR3-3C2	3x100GbE (CFP2) + 2x100GbE	Intel Broadwell-DE D1519
	● SKN-7280CR3-3C2-2	3x100GbE (CFP2) + 2x100GbE	Intel Broadwell-DE D1519
	● SKN-7280CR3-3C2-2-DEV	3x100GbE (CFP2) + 2x100GbE	Intel Broadwell-DE D1519
	● SKN-7280CR3-3C2-2G	3x100GbE (CFP2) + 2x100GbE	Intel Broadwell-DE D1519
	● SKN-7280CR3-3C2-3	3x100GbE (CFP2) + 2x100GbE	Intel Broadwell-DE D1519
	● SKN-7280CR3-3C2-3-DEV	3x100GbE (CFP2) + 2x100GbE	Intel Broadwell-DE D1519
	● SKN-7280CR3-3C2-3G	3x100GbE (CFP2) + 2x100GbE	Intel Broadwell-DE D1519
	● SKN-7280CR3-3C2-DEV	3x100GbE (CFP2) + 2x100GbE	Intel Broadwell-DE D1519
	● SKN-7280CR3-4C2	4x100GbE (CFP2) + 2x100GbE	Intel Broadwell-DE D1519
	● SKN-7280CR3-4C2-DEV	4x100GbE (CFP2) + 2x100GbE	Intel Broadwell-DE D1519
	● SKN-7280CR3-4C2G	4x100GbE (CFP2) + 2x100GbE	Intel Broadwell-DE D1519
	● SKN-7280CR3-4C6	3x100GbE (CFP2) + (9 or 10)x100GbE	Intel Broadwell-DE D1519
	● SKN-7280CR3-4C6-DEV	3x100GbE (CFP2) + (9 or 10)x100GbE	Intel Broadwell-DE D1519
	● SKN-7280CR3-4C6G	3x100GbE (CFP2) + (9 or 10)x100GbE	Intel Broadwell-DE D1519
	● SKN-7280CR3-5C2	5x100GbE (CFP2) + 2x100GbE	Intel Broadwell-DE D1519
	● SKN-7280CR3-5C2-DEV	5x100GbE (CFP2) + 2x100GbE	Intel Broadwell-DE D1519
	● SKN-7280CR3-5C2G	5x100GbE (CFP2) + 2x100GbE	Intel Broadwell-DE D1519
7280SR	● SKN-7280SR3-16YC8	4x CFP2 100G/200G + 4x 40/100G QSFP + 16x 25/10GbE SFP	Intel Broadwell-DE D1519

The TOE supports local administration via the local console port. Remote administration is performed over the Secure Shell v2 (SSHv2) protocol. Alternatively, the switch supports eAPI JSON-RPC interface over TLS for remote automation scripts to perform management functions on the switch. This interface supports only JSON request/response format. The eAPI interface supports both interactive and non-interactive modes of operation. In interactive mode, a user can issue commands and receive immediate feedback from the device. This allows for quick testing and debugging of eAPI scripts and applications, as well as ad hoc configuration and monitoring tasks.

In non-interactive mode, the user can use a third-party eAPI JSON-RPC client, such as Ansible, Python requests library, or cURL, to send eAPI requests and receive responses from the device. In the evaluated configuration, the switch only supports eAPI non-interactive mode over TLS for remote automation scripts to perform management functions on the switch.

The TOE also supports storage and forwarding of audit records, protected using SSHv2, to any syslog-compatible network entity.

1.5 Console CLI access

The console port is a serial port located on the front of the switch. Figure 1: Switch Ports, shows the console port on the TOE. The TOE's administrator should use a serial or RS-232 cable to connect to the console port. The accessory kit also includes an RJ-45 to DB-9 adapter cable for connecting to the switch.

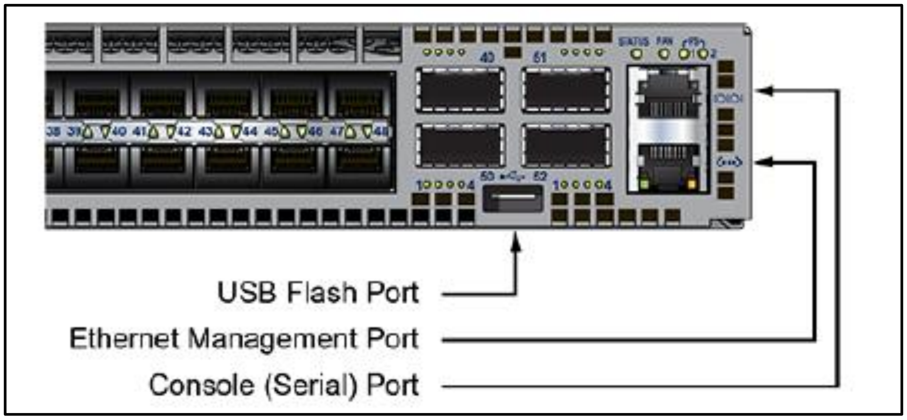

Figure 1: Switch Ports

Use these settings when connecting the console port:

- 9600 baud
- no flow control.
- 1 stop bit

- no parity bits
- 8 data bits

The boot console command configures terminal settings for serial devices connecting to the

console port. The no boot console and default boot console commands restore the factory default settings on the switch and remove the corresponding CONSOLESPEED command from the boot-config file.

Command Syntax

```
boot console speed baud_rate
```

Parameters

baud_rate console baud rate. Options include 1200, 2400, 4800, 9600, 19200, 38400, 57600, and 115200.

Example

- This command sets the console speed to 57600 baud.

```
switch(config)#boot console speed 57600
switch(config)#
```

- This command displays the result of the console-speed change.

```
switch(config)#show boot-config
Software image: flash:/EOS.swi
Console speed: 57600
Aboot password (encrypted): (not set)
switch(config)#
```

- The above boot console command adds the following line to boot-config.

```
CONSOLESPEED=57600
```

2. Usage Assumptions

At the outset, the CC mode requires the following to be followed during operation of the switch.

2.1 No General-Purpose Computing

The user must not install or run external applications, binaries or RPMs on the switch. The only allowed software to be run on the switch is the EOS image provided by Arista Networks.

2.2 Trusted Administrator

The switch administrator is trusted to follow and apply all instructions in this document.

2.3 Physical Security

The switch must be stored and installed in a physically secure location.

2.4 Secure Acceptance

- Obtain the product through authorized channels.
- Verify model identification label on the hardware.

3. Secure Configuration

This section describes the procedures to functionally enable the switch and operate in CC mode.

3.1 Software Image

It is likely that the factory default image running on the switch is not CC validated. Follow steps described below to run CC validated image on the switch.

1. Connect a terminal to the console port (9600/N/8/1) and login over the console port with username “admin”. This username does not have a password in factory default configuration. So just press enter on the password prompt.

2. Cancel zero touch provisioning for the current and subsequent boot sequences.

```
localhost>enable
localhost# zerotouch cancel
localhost# write
```

3. Download and validate the proper software image.

    a. Use a USB drive formatted with MS-DOS or FAT file system. Most USB drives are pre-formatted with a compatible file system.

    b. Download the CC validated image from the “Software Download” section of Arista website and save it to the USB drive. Keep the name of the image file as it is.

    c. Insert the USB flash drive into the USB flash port on the switch. The USB drive is auto-mounted to the following point /mnt/usb1.

    d. Before installing the CC validated version, do the following steps.

        ○ Check the current version of the EOS and note it down for the record.

        ```
        localhost#show version
        ```

        ○ Copy the running-config file to the flash drive in case you need to refer to it later.

        ```
        localhost#copy running-config flash:my_config
        ```

    e. Determine the size of the flash drive to make sure sufficient space is available for the new EOS image.

    ```
    localhost#dir flash:
    ```

    f. Copy the image file from USB to flash.

    ```
    localhost#copy usb1:<image_filename> flash:EOS.swi
    ```

    g. Verify SHA512 checksum of the image file.

    ```
    localhost#verify /sha512 flash:EOS.swi
    ```

    The output of the command will be in this format:

    ```
    verify /sha512 (flash:EOS.swi) =<hash>
    ```

    The output hash consists of 128 hex characters. Compare it with the published hash value on the Arista website. If the values match, the image is valid. However, if the values do not match the image is either corrupt or has been tampered with. Do not use it and contact Arista support to report the issue.

Note: During the update no services or functions of the TOE will cease working.

4. Install the software image.

    a. Modify boot-config file to point to the image on the flash.

    ```
    localhost#config
    localhost(config)#boot system flash:EOS.swi
    localhost(config)#show boot-config
    ```

Note: The activation of the new software image located in the flash: is activated after the reboot of the switch using the command “reload”. During this delayed activation of the new software image, the command “show boot-config” will display the loaded but inactive version of the

```
software image.
```

b. Set up an event handler to log the state of the self-tests at startup:

```
localhost#config
localhost(config)#event-handler startupLog
localhost(config-handler-startupLog)# trigger on-boot
localhost(config-handler-startupLog)# action bash
Enter Multi-line Action. Type 'EOF' on its own line
to end.
sudo echo "ARISTA_ENABLE_FIPS=1" >> /etc/environment
sudo echo "ARISTA_ENABLE_FIPS_LOGGING=1" >>
/etc/environment
logger -p local4.info `swi info /mnt/flash/.boot-
image.swi`
EOF
localhost(config-handler-startupLog)#
localhost(config-handler-startupLog)# write
Copy completed successfully.
hrm101(config-handler-startupLog)# copy running-
config startup-config
Copy completed successfully.
```

c. Save the configuration to the startup-config file and reboot the switch for the new image to take effect.

```
localhost(config)#write
localhost(config)#reload
```

Note: As part of the update procedure, a reboot of the switch will be required to facilitate the loading of the new image. Consequently, there will be a brief window of disruption affecting all functions and features until the switch is successfully reinitialized with the updated image. Subsequently, upon completion of the reboot process, all functions and features will be restored to their normal operational state.

d. Once the switch comes up, login back into the switch as “admin” and verify that the switch is running with the CC validated image.

```
localhost>enable
localhost#show version
```

3.2 Hostname, DNS Server, Time Setting, Login Banner and Password Restrictions

Set the hostname. For example, to set the hostname as “switch”, run the following command.

```
localhost#config
localhost(config)#hostname switch
```

Optionally, set the address of a DNS server to allow the use of resolvable hostnames in later instructions instead of IP addresses. For example, to set a DNS server whose IP address is “10.10.3.456” run the following command

localhost(config)#ip name-server 10.10.3.456

Set the current time.

```
switch(config)#clock timezone <zone>
switch(config)#clock set <hh:mm:ss> <mm/dd/yyyy>
switch(config)#show clock
```

Interactive sessions between the user and the switch must show a banner before connecting to warn the user on proper usage. Configure banner as follows. The exact banner text is entered as per organization policy and requirements.

```
switch(config)#banner login
      Enter TEXT message. Type 'EOF' on its own line to end.
      This is a secure switch for networking. Do not attempt to
      connect to this switch unless permitted to do so.
      EOF
switch(config)#write
```

Password length of 8 characters is the minimum requirement for CC mode, though this can be increased to 15 depending on the organizational policy. Configure it as follows.

```
switch(config)#management security
switch(config-mgmt-security)#password minimum length 8
```

Confirm the configuration.

```
switch(config-mgmt-security)#show run all | grep length
switch(config-mgmt-security)#write
switch(config-mgmt-security)#exit
switch(config)#
```

Make the default hash function used for storing passwords as SHA-512.

```
switch(config)#management defaults
switch(config-mgmt-defaults)#secret hash sha512
```

Confirm the configuration.

```
switch(config-mgmt-defaults)#show run all | section defaults
switch(config-mgmt-defaults)#write
switch(config-mgmt-defaults)#exit
switch(config)#
```

3.3 Console Idle Timeout

Configure idle timeout for console port. SSH idle timeout is configured later during SSH configuration. Following commands set the console idle timeout to 10 minutes.

```
switch(config)#management console
switch(config-mgmt-console)#idle-timeout 10
```

Confirm the configuration.

```
switch(config-mgmt-console)#show run all | section console
switch(config-mgmt-console)#write
switch(config-mgmt-console)#exit
switch(config)#
```

Note: Console session can be manually terminated by the user by entering “logout” command. The “exit” and “quit” commands also perform logout.

3.4 Default Accounts Protection

The “admin” account by default has no password. Assign a password to it.

```
switch(config)#username admin secret 0 <plaintext_password>
```

The switch has an integrated bootloader called Aboot that handles booting the main software image (EOS). This bootloader provides a shell and it must be password protected. This is accomplished with the following configuration:

```
switch(config)#boot secret 0 <plaintext_password>
```

The following characters can be used in the password: uppercase and lowercase letters, numbers and special characters “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”. The length of the password needs to be as least as much as configured in Password Restrictions, else password entry will be rejected.

User account “root” is disabled by default. To reaffirm, you can run the following command.

```
switch(config)#no aaa root
```

Arista recommends following these guidelines for creating a strong password:

1. Length: Create a password that is at least 12 characters long. Longer passwords provide better security.
2. Complexity: Include a combination of uppercase and lowercase letters, numbers, and special characters. Using a variety of character types makes the password more difficult to guess.
3. Avoid Common Patterns: Avoid using easily guessable patterns such as sequential numbers or letters (e.g., 12345678 or abcdefgh), repeated characters (e.g., 11111111), or common words and phrases.
4. Unique and Random: Generate a password that is unique and not easily associated with personal information, such as your name, birthdate, or address. Randomly generated passwords are typically more secure.
5. Avoid Dictionary Words: Avoid using common dictionary words or easily recognizable terms, as they can be easily guessed using brute-force or dictionary-based attacks.
6. Regular Updates: Change your password regularly. It is good practice to update your passwords every 90 days or according to your organization's security policies.

Remember to keep your password confidential and avoid sharing it with others.

3.5 Disallowed Access Methods (SNMP, Telnet, API)

SNMP, Telnet and some APIs must not be used in CC mode.

3.5.1 SNMP

By default SNMP is disabled. Reaffirm by running following command:

```
switch(config)#show snmp
```

There should be “SNMP agent disabled: no communities or users configured” in the output.

3.5.2 Telnet

By default Telnet is disabled. Reaffirm by running following command:

```
switch#show run all | section telnet
```

Look inside the “management telnet” section of the output and ensure that it says a “shutdown”.

3.5.3 API

By default all API management interfaces are disabled. Reaffirm by running following commands.

```
switch#show management api ?
```

This will list all available management API methods. Reaffirm that each one is disabled (except http-commands which is required for eAPI access). For example,

```
switch#show management api gnmi
```

In case, you see any of these enabled, you can disable it as follows.

```
switch#config
switch(config)#no management api gnmi
switch(config)#write
```

3.6 IP Address

The switch provides an Ethernet management port for configuring the switch and managing the network out of band. Only one port is required to manage the switch. Assign IP address to this management port. Use the management interface number that is displayed when “interface management ?” is run. In the example below, IP address 192.168.0.105 on /24 subnet is assigned to the management interface number 1/1 and default route is configured to the gateway located at 192.168.0.1.

```
switch#config
switch(config)#interface management 1/1
switch(config-if-Ma1)#IP address 192.168.0.105/24
switch(config-if-Ma1)#exit
switch(config)#ip route 0.0.0.0/0 192.168.0.1
switch(config)#write
```

3.7 Entropy

In CC mode, the switch uses two sources of entropy: i) network interrupts, ii) hardware-based entropy (from TPM). Network interrupts entropy source is always running when the switch is running. Hardware-based entropy needs to be enabled as follows.

```
switch(config)#management security
switch(config-mgmt-sec)#entropy source hardware
```

3.8 SSH Configuration

```
switch(config)#management ssh
```

Configure FIPS mode for SSH by running the following command.

```
switch(config-mgmt-ssh)#fips restrictions
```

Confirm that FIPS mode is enabled.

```
switch(config-mgmt-ssh)#show management ssh
```

When this configuration is done, it forces FIPS power-on self-tests every time a new SSH connection instance is created. Note that this happens during run time, not configuration time. After self-tests pass, then the SSH connection instance is started. When a new SSH connection instance is successfully started, a log such as follows is generated showing service id [PID] of SSH connection instance.

```
2022 Jun  1 11:22:23 switch sshd[32499]: Connection from 10.95.66.234 port 63766 on 172.30.167.171 port 22
```

If power-on self-tests fail, the SSH connection instance is not started. The failure is indicated by log pair such as follows.

```
2022 Jun  4 16:46:21 switch kernel: [  721.533308] potentially unexpected fatal signal 6.
2022 Jun  4 16:46:21 switch kernel: [  721.533315] CPU: 0 PID: 4200 Comm: ssh Tainted: P         O    4.9.122.Ar-11549262.eostrunk.1 #1
```

If you see repeated failures to start SSH connection instance during run time, contact Arista Networks.

Continuing with the configuration, specify cryptographic options used by both SSH server and client: Symmetric encryption cipher, key exchange, message authentication and server host key algorithms.

```
switch(config-mgmt-ssh)#cipher aes256-cbc aes128-cbc
switch(config-mgmt-ssh)#key-exchange diffie-hellman-group14-sha1
switch(config-mgmt-ssh)#mac hmac-sha2-256 hmac-sha2-512
switch(config-mgmt-ssh)#hostkey server rsa-sha2-256 ecdsa-nistp384
```

When the switch connects to a remote host (e.g., Syslog server) over SSH, it acts as SSH client. The switch must ensure that the public key presented by the remote host matches up with the public key imported in the switch for that host. To enforce such checking, run the following command.

```
Switch(config-mgmt-ssh)#hostkey client strict-checking
```

Set SSH logging level to verbose and enable logging to “show target system”.

```
Switch(config-mgmt-ssh)#log-level verbose
switch(config-mgmt-ssh)#logging target system
```

Configure idle timeout for SSH. For example, set idle timeout of 10 minutes for SSH sessions as follows.

```
Switch(config-mgmt-ssh)#idle-timeout 10
```

Note: SSH session can be manually terminated by user by entering “logout” command. The “exit” and “quit” commands also perform logout.

Warning: To guarantee adherence to the common criteria requirements, it is important that the SSH rekeying function is executed with unwavering compliance. This necessitates a specific timeframe of no more than 60 minutes from either the last SSH rekeying process or the initiation of the connection. Additionally, an alternative trigger for the rekeying function is upon reaching an absolute data transfer threshold of no more than a 1GB of data, whichever circumstance takes precedence.

Complete remaining SSH configurations as follows.

```
Switch(config-mgmt-ssh)#rekey frequency 1 gbytes
switch(config-mgmt-ssh)#rekey interval 1 hours
switch(config-mgmt-ssh)#exit
```

Allow the configuration to be processed by the switch, and once processed, verify it. Note that “wait-for-warmup” is a handy command to run any time to wait until the most recent configuration takes effect.

```
Switch(config)#wait-for-warmup
switch(config)#show run all | section management ssh
switch(config)#write
switch(config)#exit
```

Generate RSA host key pair for SSH. The size of the key is 2048 bits and is not configurable. Note that the following command deletes the previous key and replaces it with the new key. There can be only one rsa key for SSH at any one time. FIPS conditional self-test is performed at this time on the generated key pair and only if it passes the command completes successfully. If the command does not complete successfully repeatedly, please contact Arista Networks.

```
Switch#reset ssh hostkey rsa
```

To view the new public key, use the following command:

```
switch#show management ssh hostkey rsa public
switch#write
```

If the above changes succeed, the old key files have been destroyed per FCS_CKM.4. If these changes do not take effect, please contact Arista Support before continuing the operation of the switch.

Copy the above public key (in entirety including leading `ssh-rsa’ and trailing `chassisAddr=xx:xx:xx:xx:xx:xx' strings) to USB as it will be required to set up SSH Tunnel to the remote Syslog server.

3.9 SSH Tunnel

Audit log forwarding from the switch to the remote Syslog server is done inside an SSH Tunnel. To configure SSH Tunnel to the Syslog server, configure SSH Tunnel endpoints on the switch and the Syslog server.

3.9.1 Tunnel Endpoint on the Syslog Server

Perform following steps to configure log tunnel endpoint on the Syslog server. The user account "authuser" on the Syslog server is used in below command samples.

3.9.1.1 Add switch's public key for authentication to Syslog server

Log into the Syslog server with username "authuser". Copy the switch's entire public key from the USB to the *~/.ssh/authorized_keys* file in the home directory of "authuser" and give following permissions to the file.

```
authuser@syslog: sudo chmod 700 ~/.ssh
authuser@syslog: sudo chmod 600 ~/.ssh/authorized_keys
suthuser@syslog: sudo service sshd restart
```

3.9.1.2 Generate/Copy public host key of Syslog server

Syslog server should contain its public host key in /etc/ssh/ssh_host_rsa_key.pub file automatically created at the time the SSH service was started. You can also regenerate the host key pair if desired as follows.

```
authuser@syslog: sudo ssh-keygen -f /etc/ssh/ssh_host_rsa_key -t
rsa -b 2048
```

Copy the public host key file of Syslog server located at /etc/ssh/ssh_host_rsa_key.pub to USB. This will be required to set up the tunnel endpoint on the switch to include Syslog server into the known hosts list.

3.9.1.3 Enable public key authentication for authuser

Enable PublicKeyAuthentication for the "authuser" in the Syslog server. If Syslog server uses OpenSSH, it is enabled by default. This can be reaffirmed by uncommenting the following line in */etc/ssh/sshd_config* file.

```
PubkeyAuthentication yes
```

3.9.1.4 Software settings on Syslog server

Configuration for the logging software running on the Syslog server to collect logs forwarded by the switch to port 514 on the Syslog server is outside the scope of EOS user guidance. Please follow the corresponding software guidance (e.g., syslon-ng, rsylog, vendor proprietary syslog etc.) for these instructions.

3.9.2 Tunnel Endpoint on the Switch

Perform following steps to configure log tunnel endpoint on the switch.

First import public host key of the Syslog server from USB into known-hosts list of the switch. The key should be used in the command as Base 64 string, without its leading `ssh-rsa' and trailing `root@syslog' strings.

```
Switch#config
switch(config)#management ssh
```

If using the non-default VRF, now enter the "vrf <vrfName>" submode, otherwise the commands will be configured in the default VRF. Then continue configuring the commands.

```
Switch(config-mgmt-ssh)#known-hosts <syslog_ip> rsa <key>
```

If using ECDSA host keys, there is a known issue where the above command will not configure the known-hosts list correctly. As a workaround, configure the following event-handler to adjust the known-hosts list of the switch:

```
switch(config)# event-handler knownHosts

switch(config-handler-knownHosts)# action bash echo <syslog_ip>
ecdsa-sha2-nistp384 <key> >> /etc/ssh/ssh_known_hosts-<vrfName>
switch(config-handler-knownHosts)# trigger on-logging
switch(config-handler-knownHosts-logging)# poll interval 1
switch(config-handler-knownHosts-logging)# regex
SSH_TUNNEL_HOSTKEY_VERIFY_FAILED
```

With this workaround, the switch will automatically add the host key to the known-hosts list directly if it detects a failure in host key verification. If using the default VRF, use /etc/ssh/ssh_known_hosts instead of /etc/ssh/ssh_known_hosts-<vrfName>. Then, go back to the management SSH mode (and vrf submode if applicable) to continue configuring the commands.

Now, to create a tunnel with name "LogTunnel".

```
Switch(config-mgmt-ssh)#tunnel LogTunnel
```

Configure the tunnel so that the messages sent to the switch port 514 in the switch will be forwarded to the remote Syslog server on port 22 using the user account "authuser" and then forwarded to port 514 on the remote Syslog server. If the remote syslog server is listening on a different port than 514, the port number under "remote host" will need to be changed to that listening port.

```
Switch(config-mgmt-ssh-tunnel-LogTunnel)#local port 514

switch(config-mgmt-ssh-tunnel-LogTunnel)#ssh-server <syslog_ip>
user authuser port 22

switch(config-mgmt-ssh-tunnel-LogTunnel)#remote host localhost
port 514
```

Set the rate that SSH keep-alive packets will be sent and how many can be lost before the connection is declared dead. Note that these packets are sent inside the SSH tunnel. Run the following command to send 6 keep-alive packets in 60 seconds.

```
Switch(config-mgmt-ssh-tunnel-LogTunnel)#server-alive count-max 6

switch(config-mgmt-ssh-tunnel-LogTunnel)#server-alive interval 10
switch(config-mgmt-ssh-tunnel-LogTunnel)#no shutdown
switch(config-mgmt-ssh-tunnel-LogTunnel)#exit
switch(config-mgmt-ssh)#exit
```

Allow the configuration to be processed by the switch, and once processed, verify it.

```
Switch(config)#wait-for-warmup
switch(config)#show run all | section management ssh
switch(config)#write
```

3.10 Audit Logs Configuration

Configure logging to specific port number (514 is default) on the switch. By virtue of configuration of SSH Tunnel on this port as described before, log messages are securely tunneled inside SSH to remote Syslog server. Log is sent into the tunnel as soon as it is generated.

```
Switch(config)#logging host localhost protocol tcp
```

The above command is if logging is running in the default VRF. If a different VRF is being used, use the following command instead:

```
switch(config)#logging vrf <vrfName> host localhost protocol tcp
```

Configure local persistent audit log storage. Log is written to local persistent storage as soon as it is generated.

```
Switch(config)#logging persistent 4096
```

Here the number 1024 indicates the size limit of the persistent log file in bytes. When the file exceeds its size limit, it is trimmed to remove the oldest audit logs until the size drops below 1024 bytes.

Run following commands to ensure that logs for necessary security relevant events are generated.

```
Switch(config)#logging format timestamp traditional year
switch(config)#logging trap informational

switch(config)#logging level all 6

switch(config)#logging trap system tag sshd

switch(config)#logging trap system tag ssh

switch(config)#logging trap system tag nginx

switch(config)#logging trap system tag rsyslogd

switch(config)#aaa accounting exec default start-stop logging

switch(config)#aaa accounting system default start-stop logging

switch(config)#aaa accounting commands all default start-stop
```

```
logging

switch(config)#wait-for-warmup

switch(config)#show logging system

switch(config)#write
```

Run following command to see the logs on CLI. You can also do | grep to this command to search specific keywords.

```
Switch(config)#show logging system
```

The number of logs returned can be limited by passing a number that indicates the number of most recent logs to view. For example, the below command will return the 10 most recent logs.

```
Switch(config)#show logging system 10
```

3.11 Named User Accounts

Consecutive authentication failures result into temporary account lockout for remote administrative user. The threshold number of failures between 1 and 255 and subsequent lockout period are configured by Security Administrator when initializing the TOE. The RS-232/VT-100 local administrative interface is never locked out.

The “admin” account should not be used during operation of the switch in CC mode. Once the switch is operation in CC mode, the “admin” account can be used to take the switch out of CC mode to provide system updates, maintenance and user management. During operation of the switch in CC mode, the named user accounts created in “cc-admin” role as described below must be used.

Configure user authentication:

```
switch(config)#aaa authentication login default local
switch(config)#aaa authentication enable default local
```

Configure authentication policy to log successful and failed login attempts:

```
switch(config)#aaa authentication policy on-success log
switch(config)#aaa authentication policy on-failure log
```

Configure authentication policy to lockout invalid login attempts. For example, to configure 15 minutes lockout after 3 consecutive failures in 5 minutes window, run the following command.

```
Switch(config)#aaa authentication policy lockout failure 3 window
300 duration 900
```

Configure command authorization:

```
switch(config)#aaa authorization exec default local
switch(config)#aaa authorization commands all default local
switch(config)#aaa authorization serial-console
```

Set up default authorization role as “cc-admin”. This way, if user is created without specifying a role, the user gets this default role.

```
Switch(config)#aaa authorization policy local default-role cc-
admin
```

Confirm the configuration:

```
switch(config)#show run all | section aaa

switch(config)#wait-for-warmup

switch(config)#write
```

Configure authorizations for “cc-admin” role. If following rules are configured for “cc-admin” role, users in this role can do most configurations except AAA. They cannot execute general purpose commands. Also any secrets such as password hashes are suppressed in their output. Note that you have to enter <ctrl v> as escape signal before entering ?.

```
switch(config)#role cc-admin

switch(config-role-cc-admin)#permit command show running-config
all sanitized

switch(config-role-cc-admin)#permit command copy .* certificate:

switch(config-role-cc-admin)#permit command copy .* sslkey:

switch(config-role-cc-admin)#permit command copy .* flash:EOS.swi

switch(config-role-cc-admin)#permit command copy running-config
startup-config

switch(config-role-cc-admin)#permit command ssh-server

switch(config-role-cc-admin)#permit command delete certificate:.*

switch(config-role-cc-admin)#permit command delete sslkey:.*

switch(config-role-cc-admin)#deny command
>|>>|extension|\||session|do|delete|copy|rmdir|mkdir|python-
shell|bash|platform|scp|append|redirect|tee|more|diag|less|ssh
|who|show run.*|show start.*

switch(config-role-cc-admin)#deny mode config command
(no|default)?(role|aaa|tcpdump|schedule|event.*)

switch(config-role-cc-admin)#permit command .*

switch(config-role-cc-admin)#exit
```

View the role authorization definition (note that network-admin and network-operator are roles that are natively defined in EOS).

```
Switch(config)#wait-for-warmup

switch(config)#show users roles

switch(config)#write
```

Note: You can remove any rule from the role by entering “no <rule number>” command when inside that role submode and then exiting the role submode.

If organization policy so requires, custom roles can be created using guidance provided in “Section 4.4 Role-Based Authorization” of [User Manual]. You can use online tools such as regexper.com to visualize the rule implemented by regular expression.

Create named users in “cc-admin” role as follows.

```
Switch(config)#username <name> secret 0 <plaintext_password> role
```

```
        cc-admin
```

Valid usernames begin with A-Z, a-z, or 0-9 and may also contain any of these characters: @ # $ % ^ & * - _ =+ ; < > , . ~ |.

The following characters can be used in the password: uppercase and lowercase letters, numbers and special characters "!", "@", "#", "$", "%", "^", "&", "*", "(", ")". The length of the password needs to be as least as much as configured before in Password Restrictions, else password entry will be rejected.

In order to configure key based SSH authentication for the user, use following command.

```
        Switch(config)# username <name> sshkey <SSH-KEY>
```

Note that the SSH-KEY will have the key type mentioned at the beginning (ssh-rsa) followed by the space and the key value. The key size for user authentication must be at least 2048 bits. When key based authentication is configured, that is performed by the switch first. If that fails, the switch performs password based authentication. If password based authentication is not desired for the key based user, disable password based authentication for that user by running the following command.

```
        Switch(config)# username <name> secret *
```

To view the user accounts, run the following command.

```
        Switch(config)#show users accounts
```

Additionally, a 'default' role must be configured to prevent users that are not assigned a role from running any commands. Configure the following to set this:

```
        switch(config)#role do-nothing
        switch(config-role-do-nothing)#10 deny command .*
        switch(config-role-do-nothing)#exit
        switch(config)#aaa authorization policy local default-role do-
nothing
```

Now, any username which does not have a role assigned will be assigned the 'do-nothing' role upon logging in. They will be able to log in but will be unable to run any commands other than 'exit' to disconnect from the system. This behavior sees the most usage in eAPI, which is documented in a different section.

3.12 TLS Server

Perform following steps to securely configure TLS Server. It will be used by eAPI.

3.12.1 Create SSL profile

Run the following commands to create SSL profile called "TLSserv".

```
        Switch#config
        switch(config)#management security
        switch(config-mgmt-security)#ssl profile TLSserv
```

3.12.2 Enable FIPS mode

Run the following command to enable FIPS mode for TLS server.

```
Switch(config-mgmt-sec-ssl-profile-TLSserv)#fips restrictions
```

When the above command is run, FIPS power-up self-tests are performed. If they succeed, FIPS mode is enabled. Confirm that FIPS mode is enabled.

```
Switch(config-mgmt-sec-ssl-profile-TLSserv)#show management http-
server
```

If power-up self-tests fail, the TLS server (nginx service) does not start. The failure s is indicated by a pair of audit logs as follows:

```
2022 Jun  4 16:48:40 switch kernel: [  860.681208] potentially unexpected fatal signal 6.

2022 Jun  4 16:48:40 switch kernel: [  860.681216] CPU: 0 PID: 4970 Comm: nginx
Tainted: P           O    4.9.122.Ar-11549262.eostrunk.1 #1
```

If you see repeated failures to start nginx service, please contact Arista Networks.

When FIPS mode is enabled for the TLS server issues with client certificate authentication result in a rejection at the TLS layer. This is different from the normal behavior where the rejection occurs at the HTTP (application) layer and contains a detailed log message on the switch. Rejections at the TLS layer will not contain detailed log messages. If detailed failure information for client certificates is required, for example, when testing the creation of valid client certificates, FIPS mode for the TLS server can be temporarily turned off. The following sample error messages show the detailed failure reasons that can be seen when FIPS mode is off:

Error log when the client certificate signing CA is unknown:

```
2021 Oct 14 14:59:33 do392 nginx: 2021/10/14 14:59:33 [info] 11974#0:
*29 SSL_do_handshake() failed (SSL: error:14094418:SSL
routines:ssl3_read_bytes:tlsv1 alert unknown ca:SSL alert number 48)
while SSL handshaking, client: ::ffff:10.242.233.93, server: [::]:443
```

Error log for when the client certificate has expired:

```
2021/10/15 10:38:20 [info] 19710#0: *1 client SSL certificate verify
error: (10:certificate has expired) while reading client request
headers, client: ::ffff:10.242.233.93, server: , request: “POST
/command-api HTTP/1.1”, host: “172.30.167.157”
```

Error log when the client certificate has been revoked by a CRL:

```
2021 Oct 14 15:07:48 do392 nginx: 2021/10/14 15:07:48 [info] 12434#0:
*32 client SSL certificate verify error: (23:certificate revoked) while
reading client request headers, client: ::ffff:10.242.233.93, server: ,
request: “POST /command-api HTTP/1.1”, host: “172.30.167.194”
```

Turning off FIPS mode is only allowed for testing purposes. When testing is completed FIPS mode must be turned back on for the TLS server.

3.12.3 Generate CSR and obtain signed certificate

Generate a TLS server RSA key pair of 2048 bits, let us call this key TLSserv_key. Note that if this key already exists, it will be modified after running the following command. If it does not exist, it will be newly created. Also, FIPS conditional self-test is performed at this time on the generated key pair and only if it passes the command completes successfully. If the command does not complete successfully repeatedly, please contact Arista Networks.

```
      Switch#security pki key generate rsa 2048 TLSserv_key
```

The list of keys can be viewed by running the following command.

```
      Switch#dir sslkey:
```

If any key is not needed, it can be deleted by running the following command. You can confirm that key is generated by doing the following:

```
      switch#deletedir sslkey:<key name>
```

Using the above key, generate CSR by running the following command. When the command is run, the user is prompted to answer a number of questions. Provides answers to some of those questions as stated below. Providing answers to other questions is optional.

```
      Switch#security pki certificate generate signing-request

      PKI Key to use for CSR: TLSserv_key

      Common Name for use in subject: <switch identifier>

      Two-Letter Country Code for use in subject: <country code>

      Organization Name for use in subject: <organization name>

      Organization Unit Name for use in subject: <sub-organization
      name>
```

Copy and paste the output to a file on USB. Submit this file to a trusted CA to obtain an X.509v3 certificate signed by it. Suppose the signed certificate file is TLSserv.pem. Import this file into the switch’s certificate folder. Note that if this certificate already existed, it will be modified after running the following command. If it did not exist, it would be newly created.

```
      Switch#copy usb1:/TLSserv_certificate.pem
certificate:TLSserv_certificate.pem
```

Import all intermediate certificates and root certificates for the above certificate in the switch’s certificate folder. For example, suppose TLSserv_certificate.pem is signed by IntCA_certificate.pem, which in turn is signed by RootCA_certificate.pem.

```
      switch#copy usb1:/IntCA_certificate.pem
      certificate:IntCA_certificate.pem

      switch#copy usb1:/RootCA_certificate.pem
      certificate:RootCA_certificate.pem
```

The list of certificates can be viewed by running the following command.

```
      Switch#dir certificate:
```

If any certificate there is not needed, it can be deleted by running the following command.

```
      Switch# delete certificate:<certificate name>
```

If the above changes succeed, any old certificates or key files have been destroyed per FCS_CKM.4. If these changes do not take effect, please contact Arista Support before continuing the operation of the switch.

3.12.4 Define cipher suites

Configure FIPS compliant ciphers and Diffie-Hellman parameters. EOS supports the Diffie-Hellman parameters ffdhe2048, ffdhe3072, and ffdhe4096.

```
Switch(config-mgmt-sec-ssl-profile-TLSserv)#tls versions 1.2
```

```
switch(config-mgmt-sec-ssl-profile-TLSserv)#cipher-list AES128-
SHA256:AES256-SHA256:DHE-RSA-AES128-SHA256:DHE-RSA-AES256-
SHA256:AES256-GCM-SHA384:DHE-RSA-AES256-GCM-SHA384
```

```
switch(config-mgmt-sec-ssl-profile-TLSserv)#diffie-hellman
parameters ffdhe2048
```

The ffdhe3072 and ffdhe4096 parameters can be configured similarly. Only one parameter can be enabled at a time.

```
switch(config-mgmt-sec-ssl-profile-TLSserv)#write
```

Note: The TLS server implements session resumption using session IDs according to RFC 5246 (TLS 1.2). Session resumption is enabled by default and cannot be disabled. There is no configuration on the TOE to modify the session resumption behavior.

3.12.5 Configure certificate check restrictions

Extended key usage in server certificate must have Server Authentication purpose and in client presented certificate must have Client Authentication purpose.

```
switch(config)#management security
```

```
switch(config-mgmt-sec)#ssl profile TLSserv
```

```
switch(config-mgmt-sec-ssl-profile-TLSserv)#certificate
requirement extended-key-usage
```

Basic constraints must be set True for all CA certificates.

```
switch(config-mgmt-sec-ssl-profile-TLSserv)#trust certificate
requirement basic-constraint ca true
```

```
switch(config-mgmt-sec-ssl-profile-TLSserv)#chain certificate
requirement basic-constraint ca true
```

Certificate chain imported in the switch must end in root.

```
switch(config-mgmt-sec-ssl-profile-TLSserv)#chain certificate
requirement include root-ca
```

3.12.6 Load certificate chain and CRLs

Load server leaf certificate.

```
switch(config-mgmt-sec-ssl-profile-TLSserv)#certificate
TLSserv_certificate.pem key TLSserv_key
```

Designate intermediate and root certificates in the chain and mark them as trusted. Note that when the certificate chain is presented by an eAPI client, it will be verified by TLS Server up to the first CA encountered in the presented chain that is among the above designated trusted CAs.

```
switch(config-mgmt-sec-ssl-profile-TLSserv)#chain certificate
```

```
IntCA_certificate.pem

switch(config-mgmt-sec-ssl-profile-TLSserv)#trust certificate
IntCA_certificate.pem

switch(config-mgmt-sec-ssl-profile-TLSserv)#chain certificate
RootCA_certificate.pem

switch(config-mgmt-sec-ssl-profile-TLSserv)#trust certificate
RootCA_certificate.pem
```

Obtain current CRLs of IntCA and RootCA and import them into switch as follows.

```
switch(config-mgmt-sec-ssl-profile-TLSserv)#copy
usb1:IntCA_CRL.pem certificate:IntCA_CRL.pem

switch(config-mgmt-sec-ssl-profile-TLSserv)#crl IntCA_CRL.pem

switch(config-mgmt-sec-ssl-profile-TLSserv)#copy
usb1:RootCA_CRL.pem certificate:RootCA_CRL.pem

switch(config-mgmt-sec-ssl-profile-TLSserv)#crl RootCA_CRL.pem
```

Now make configuration for daily update of above CRLs. To do this you have to make a bash script as follows. Please note that “bash” access is only allowed for the “admin” user during setup. Once the “role” is loaded for the cc-admin, bash access is disallowed.

```
switch(config)#bash

[admin@switch ~]$cd /mnt/flash
```

Make a bash script file called IntCA_CRL.sh.

```
[admin@switch flash]$vi IntCA_CRL.sh
```

Add the following lines in the file, then save the file and exit the editor. You have to enter “i” to enter the edit mode of the vi editor and ESC :wq ENTER to save and exit the vi editor.

```
#!/bin/bash

FastCli -p15 -e -c $'conf \ndelete certificate: AcumenInterCA.crl
\nend \n'

sudo ip netns exec ns-mgmt curl
http://10.1.4.175/AcumenInterCA.crl -o /tmp/AcumenInterCA.crl

FastCli -p15 -e -c $'conf \ncopy file:/tmp/AcumenInterCA.crl
certificate:IntCA_CRL.pem \nend \n'
```

Now schedule the periodic execution of the above script as follows.

```
switch(config)#schedule updateIntCRL <start time> interval 1440
timeout 10 max-log-files 3 command  bash /mnt/flash/IntCA_CRL.sh
```

Confirm the schedule.

```
switch(config)#show schedule summary

switch(config)#write
```

Do the same steps as above for Root CA CRL and other Int CA CRL updates.

Note: If you want to remove any schedule, you can do it with the “no schedule <schedule name>” command.

Every time the above bash script runs, it first deletes an existing CRL. This causes the SSL profile to become invalid. Thereafter, when a new valid CRL download and copy succeeds, the SSL profile becomes valid again. Thus, successful outcome of the script in each run generates a sequence of audit logs as shown in below sample.

A. 2022 May 31 12:03:03 switch Aaa: %ACCOUNTING-6-CMD: unknown,uid=0 unknown unknown stop task_id=27351 start_time=1559329383.66 timezone=PST service=shell priv-lvl=15 cmd=delete certificate:IntCA_CRL.pem <cr>

B. 2022 May 31 12:04:35 switch ConfigAgent: %SECURITY-6-SSL_KEY_CERTIFICATE_DELETED: SSL certificate IntCA_CRL.pem has been deleted with the SHA-256 hash of c336aad7bf58403b9235c460de6c01a9de576c965bb86838be412fe570cd7145

C. 2022 May 31 12:05:03 switch SuperServer: %SECURITY-3-SSL_PROFILE_INVALID: SSL profile 'TLSserv' is invalid. Check 'show management security ssl profile TLSserv' for details

D. 2022 May 31 12:05:54 switch Aaa: %ACCOUNTING-6-CMD: unknown,uid=0 unknown unknown stop task_id=27354 start_time=1559329384.11 timezone=PST service=shell priv-lvl=15 cmd=copy <http URL for CRL> certificate:IntCA_CRL.pem <cr>

E. 2022 May 31 12:08:29 switch ConfigAgent: %SECURITY-6-SSL_KEY_CERTIFICATE_IMPORTED: SSL certificate IntCA_CRL.pem has been imported with the SHA-256 hash of c336aad7bf58403b9235c460de6c01a9de576c965bb86838be412fe570cd7145

F. 2022 May 31 12:10:23 switch SuperServer: %SECURITY-6-SSL_PROFILE_VALID: SSL profile 'TLSserv' is valid

If the new valid CRL download and copy does not succeed, logs E and F above will not be generated, and the SSL profile will remain invalid until the valid CRL is again downloaded and copied. No new authentication attempts succeed when the SSL profile is invalid.

In the unlikely situation that the script fails to execute, logs A to F above will not generate.

In the unlikely situation that the delete command fails, logs B and C will not generate.

Administrators must monitor the above logs to ensure that they all generate around the time the script is scheduled to run. If they don’t, it indicates a problem with CRL update and must be investigated.

3.12.7 Validate SSL profile configuration

At this time, make sure that the SSL profile is valid. If not, the state of the profile will be shown as invalid in the output of the command along with the reason for invalidity. Make sure to remediate the reason for invalidity and then exit the profile sub-mode.

```
switch(config-mgmt-sec-ssl-profile-TLSserv)#wait-for-warmup

switch(config-mgmt-sec-ssl-profile-TLSserv)#show management
security ssl profile

switch(config-mgmt-sec-ssl-profile-TLSserv)#write

switch(config-mgmt-sec-ssl-profile-TLSserv)#exit

switch(config-mgmt-sec)#exit
```

Note: During the configuration, if it is required to delete a CRL or an SSL Profile, use commands "no crl <CRL name>" and "no ssl profile <profile name>". Trust certificates can be removed via "no trust certificate <certificate name>". Chain certificates can be removed via "no chain certificate <certificate name>".

3.12.8 Cryptographic key destruction

There are limited circumstances under which the destruction of cryptographic keys could be delayed, but not prevented. When cryptographic keys are generated, they are initially saved to disk on a RAM backed folder named /persist/secure. Any changes made to /persist/secure are backed up as an archive to flash memory. Before saving a new archive to flash memory, the old archive is removed using the UNIX "scrub" command. The new archive is then saved to flash memory in a single move operation.

The combination of the "scrub" command followed by the move operation ensures that old cryptographic keys are first scrubbed before they are deleted, thereby delaying their destruction. This approach provides a defense in depth for clearing the keys, as it is not possible to prevent the destruction of the keys. However, if other operations are being performed, the flash memory may take a brief period of time to process, causing a delay in the destruction of the keys.

3.12.9 X509 certificate validation

During the establishment of a TLS connection, the eAPI client presents its x.509v3 certificate to the TLS server within the TOE (Target of Evaluation). This certificate is utilized by the TOE to authenticate the eAPI client. The TOE performs the following checks on the certificate provided by the eAPI client:

- The TOE verifies that the certificate chain presented by the eAPI client can be traced back to the lowest CA certificate within the hierarchy of certificates imported into the TOE. This lowest CA certificate acts as the trust anchor. It is important to note that the trust anchor can either be an intermediate CA certificate or the root CA certificate.
- The TOE validates that the current date and time fall within the "Valid from" and "Valid to" time frame for each certificate in the chain presented by the eAPI client, starting from the leaf certificate up to and including the trust anchor.
- The TOE checks that the basicConstraints extension is included in all CA certificates within the chain, with the CA flag set to TRUE.
- The TOE ensures that the leaf certificate in the chain presented by the eAPI client contains the extendedKeyUsage field with the purpose of Client Authentication (id-kp 2 with OID 1.3.6.1.5.5.7.3.2).
- The TOE verifies the correctness of the digital signatures in all certificates.
- The TOE confirms that none of the certificates within the chain, from the leaf up to the trust anchor, are revoked. It is not necessary to check the revocation status of the trust anchor and other CA certificates upstream of the trust anchor.

If any of the mentioned checks fail, the TLS connection is rejected. The TOE does not support a X509 certificate fallback authentication function, if any of these x509 certificate validity checks fail, the session is terminated.

The EOS does not provide support for code signing and OCSP signing attributes in the extendedKeyUsage field of the leaf certificate when presented by the e-API client or when included in an x509 certificate imported into its trusted store.

To enable revocation checking on the eAPI client certificate chain as described above, the Security Administrator is responsible for specifying Certificate Distribution Points (CDPs) during the initial configuration of the TOE (Target of Evaluation). It is mandatory for the Security Administrator to designate CDPs for the Certificate Revocation Lists (CRLs) published by the trust anchor and each CA certificate between the trust anchor and the leaf certificate of the eAPI client.

The TOE automatically downloads CRLs from the designated CDPs at regular intervals, refer to Section 3.14 for the instructions, and stores the most recently retrieved CRLs locally. During the download process, the digital signatures on the downloaded CRLs are validated. If the signature verification fails, the CRL is not added to the local copy.

The locally stored copies of the CRLs are utilized for performing certificate revocation checking. During revocation checking, the TOE ensures that the current time falls within the validity period of the CRL, which is specified between the effective date and the next update date mentioned in the CRL.

Revocation checking can result in failure for two reasons. Firstly, the certificate may be revoked according to the CRL. Secondly, the recent CRL for the CA that issued the certificate may not be present in the local copy, thus preventing the revocation checking from being performed.

3.13 eAPI Server Configuration

Now that the SSL profile is configured, eAPI can be set to use that profile as follows. First ensure that the http-commands API is appropriately initialized.

```
switch(config)#no management api http-commands
switch(config)#management api http-commands
switch(config-mgmt-api-http-cmds)#vrf mgmt
switch(config-mgmt-api-http-cmds)#no shutdown
switch(config-mgmt-api-http-cmds)#write
switch(config-mgmt-api-http-cmds)#exit
switch(config)#write
switch#show management api http-commands
```

Configure HTTP server settings.

```
switch(config)#management http-server
switch(config-mgmt-http-server)#no protocol http
switch(config-mgmt-http-server)#no default-services
switch(config-mgmt-http-server)#protocol https ssl profile
TLSserv
```

```
switch(config-mgmt-http-server)#log-level info

switch(config-mgmt-http-server)#exit
```

Check HTTPS server settings.

```
switch(config)#wait-for-warmup

switch(config)#show management http-server

switch#show management api http-commands

switch(config)#write
```

In the event of X509 authentication failure for any reason, the Target of Evaluation (TOE) terminates the TLS communication with the eAPI client. It's important to note that the TOE does not incorporate any mechanism for overriding x509 certificates. Clients are required to authenticate using x509 certificates. This is achieved by creating a user account whose password is disabled. The common name provided in the certificate must exist on the switch. Perform the following steps to create user account for eAPI client. This assigns network-operator role to eAPI client. So, it can remotely run show commands, but cannot run any config commands. Also, eAPI client cannot login with password credential.

```
switch(config)#username eAPI secret * role network-operator
```

One last step is to restrict untrusted connection attempts to the eAPI TLS server. To that end, first, set up an access control list (ACL) to only allow trusted IP addresses to access the TLS server:

```
switch(config)# ip access-list block-untrusted
```

This is how to add an ACL to permit a specific subnet:

```
switch(config-acl-block-untrusted)#permit tcp <IP subnet to
permit> any eq https
```

As an example, permitting a specific subnet would look like:

```
switch(config-acl-block-untrusted)#permit tcp 172.16.0.0/16 any eq
https
```

This is how to add an ACL to permit a specific IP address:

```
switch(config-acl-block-untrusted)#permit tcp host <specific IP
address> any eq https
```

As an example, permitting a specific ip address would look like:

```
switch(config-acl-block-untrusted)#permit tcp host 10.0.0.3 any
eq https
```

Multiple rules can be configured depending on which IP addresses are allowed. When all rules have been entered, exit the mode and save the configuration.

```
switch(config-acl-block-untrusted)#exit

switch(config)#write
```

Then, configure the ACL on the server as follows:

```
switch(config)#management http-server

switch(config-mgmt-http-server)#vrf <fill in with VRF name or use
"default" if no VRF configured>
```

```
switch(config-mgmt-http-server-vrf-<vrfName>)#ip access-group
block-untrusted in

switch(config)#write
```

3.13.1 eAPI Fallback authentication

The eAPI provides support for password-based authentication as a fallback option. The availability of fallback authentication depends on the configuration of the "secret" parameter in the "username" command. To enable or disable fallback authentication, the following command should be used as a replacement:

```
switch(config)#username eAPI secret <user_Password> role network-
operator
```

If the "secret" parameter is configured with a password, such as in the example "`username eAPIAdmin1 secret myPassw0rd role cc-admin`", fallback authentication is enabled. In this case, the client can authenticate by providing their password (myPassw0rd) instead of a client certificate.

However, if the "secret" parameter is configured with the "*" (asterisk) character, as shown in the "`username eAPIAdmin1 secret * role cc-admin`" command, fallback authentication is disabled. In this scenario, the client will not be able to authenticate using their password.

It is highly recommended to prioritize the use of client certificates over passwords whenever possible. This can be achieved by disabling fallback authentication through the configuration "`secret *`"

3.14 eAPI Client Operation

The eAPI client can issue a subset of CLI commands and receive response using any program that can retrieve content from the server via HTTP methods. For example, a popular Linux utility called ‘wget’ provides such facility. Run the following command on client machine to issue eAPI request to the switch.

```
eapi-host#wget --ca-certificate=<file1> --certificate=<file2> --
private-key=<file3> -q -O - https://<server>/command-api --post-
data="$(cat <file-with-json-request>)”
```

Here <file1> stores bundle of certificate authorities in PEM format to verify server’s certificate, <file2> stores client certificate in PEM format and <file3> stores client’s private key.

Request details are stored in <file-with-json-request> in following format:

```
{
        "jsonrpc": "2.0",
        "method": "runCmds",
        "params": {
                "version": 1,
                "cmds": [ "<command to run>" ],
                "format": "text"
                },
        "id": "1"
}
```

The response from the switch will contain the output of the command if the command succeeded or error indication if there was a problem with running the command.

Note that the certificate used to authenticate, <file2> in the above example, must have the username as the CN attribute in the Subject: of the x509 certificate. This is to tie the username to the allowed role. An example output obtained by running “openssl x509 -text -in <file2>” shows a certificate tied to the user account eAPIAdmin1.

```
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number: 8 (0x8)
  Signature Algorithm: sha256WithRSAEncryption
    Issuer: CN=IntCA1_Good_EKU
    Validity
      Not Before: Aug 22 17:59:19 2021 GMT
      Not After : Aug 21 17:59:19 2022 GMT
    Subject:  CN=eAPIAdmin1
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
        Public-Key: (2048 bit)
```

For the login to complete there should be a corresponding username configured on the switch, with the correct role, via a command like:

```
switch(config)#username eAPIAdmin1 secret * role cc-admin
```

The account used for eAPI operations must be configured with the "secret *" command, as demonstrated in the provided configuration. This command serves two purposes: disallowing password-based logins and enforcing the use of client certificate authentication. It is important to note that client certificate authentication is always available for eAPI, but it is not mandatory. The purpose of configuring "secret *" is to prevent the use of passwords entirely and maximize security by requiring the use of certificates during the authentication process.

It is worth mentioning that even if a client certificate is not provided, application data flow will still occur. This is because, as a fallback measure, the application will request a password. However, due to the "secret *" setting, no password will ever be accepted. The reason for requesting a password despite its inability to succeed is to avoid leaking information about the device configuration. If the EOS switch did not request a password and simply closed the connection, it would signal to a malicious client that the connected account does exist and is configured with "secret *". To prevent information leakage, a password is always requested but will never result in a successful authentication.

3.15 Timekeeping

After each power cycle it is important to confirm, and potentially reset, the time of the clock. To confirm the time of the clock use “show clock” which will return an output along the lines of:

```
switch >show clock
Fri Apr 14 13:32:19 2023
Timezone: UTC (using )
Clock source: local
```

If the time is off by more than 30 seconds the time must be reset to the current time using the command “clock set hh:mm:ss yyyy-mm-dd”. An example of setting the time to Apr 14, 2023, at 1:32 PM is:

```
(config)#clock set 13:32:00 2023-04-12
```

After setting the time the “show clock” command can be used to confirm the value being set.

4. Auditable Events

This section describes the format of audit logs. The format identifies what and where the key information is present in the audit logs. Actual audit logs may have additional system specific information in them. Several log samples are provided below for actual audit logs.

4.1 FAU_GEN.1.1 - Audit Data Generation

4.1.1 Start-up and Shut-Down of Audit Functions

The audit function is automatically started when the EOS is loaded. The audit function runs as long as the system is running. By preventing the user in the cc-admin role to run any config mode commands as described in the Named User Accounts section, it is ensured that no changes are made to the audit configuration by the user.

4.1.2 Administrative Login and Logout

See description of audit logs for FIA_UIA_EXT.1 and FIA_UIA_EXT.2.

4.1.3 TSF Data Related Configuration Changes

See description of audit logs for FMT_SMF.1.

4.1.4 Cryptographic Keys

SSH key

The system permits only one SSH host key of a given algorithm type to be present. In CC mode, since we have restricted SSH to use only RSA for the host key, the only host key for SSH that is relevant here is the one with key name as “rsa”. This key is created or updated by using the same command. Execution of this commands is logged as follows.

- Command Log

  Format: <time> <switch> <username> <user IP address> service=shell priv-lvl=15 cmd=reset ssh hostkey <key name>.

  Log Sample:

  ```
  2022 May 24 00:36:53 switch Aaa: %ACCOUNTING-6-CMD: CCUser vty3 10.95.66.234
  stop task_id=297 start_time=1558676213.45 timezone=EST service=shell priv-lvl=15
  cmd=reset ssh hostkey rsa <cr>
  ```

The above log will be followed by the following log which provides additional information.

- SSH_HOST_KEY_UPDATED

  Format: <time> SSH host key <key name> has been updated. The SHA-256 hash of public key <key name> fingerprint is <hash value>.

  Log Sample:

  ```
  2022 May 31 09:17:29 switch SuperServer : SYS-6-SSH_HOST_KEY_UPDATED: SSH host
  key rsa has been updated. The SHA-256 hash of public key <keyName> fingerprint is
  aec070645fe53ee323763059376134f0f8cc337247c978add178b6ccdfb0012a
  ```

TLS Key

When a key pair is generated for TLS for the first time, a name has to be given to the key which uniquely identifies it. Later when that key needs to be updated, the same command is run again. The key can also be deleted. Execution of these commands is logged as follows.

- Command Logs

  Format: <time> <switch> <username> <user IP address> service=shell priv-lvl=15 cmd=security pki key generate rsa 2048 <key name>.

  Format: <time> <switch> <username> <user IP address> service=shell priv-lvl=15 cmd=delete sslkey: <key name>.

  Log Samples:

  ```
  2022 May 24 01:49:20 switch Aaa: %ACCOUNTING-6-CMD: admin vty3 10.95.66.234
  stop task _id=356 start_time=1558680560.52 timezone=EST service=shell priv-lvl=15
  ```

```
cmd=security pki key generate rsa 2048 TLSserv_key <cr>

2022 May 24 01:51:06 switch Aaa: %ACCOUNTING-6-CMD: admin vty3 10.95.66.234
stop task _id=361 start_time=1558680667.0 timezone=EST service=shell priv-lvl=15
cmd=delete TLSserv_key <cr>
```

The above log will be followed by the following log to provide additional information.

- SSL_KEY_CERTIFICATE_IMORTED

  Format: <time><switch> SSL private key <key name> has been created with SHA-256 hash of <hash value>.

  Log Sample:

  ```
  2022 May 29 09:17:29 switch ConfigAgent: %SECURITY-6-
  SSL_KEY_CERTIFICATE_IMPORTED: SSL private key TLSserv_key has been imported with
  the SHA-256 hash of
  aec070645fe53ee3b3763059376134f058cc337247c978add178b6ccdfb0019f
  ```

- SSL_KEY_CERTIFICATE_DELETED

  Format: <time><switch> SSL private key <key name> has been deleted with the previous SHA-256 hash of <hash value>.

  Log Sample: 2022 May 29 09:17:29 switch ConfigAgent: %SECURITY-6-

  ```
  SSL_KEY_CERTIFICATE_IMPORTED: SSL private key TLSserv_key has been deleted with
  the previous SHA-256 hash of
  aec070645fe53ee3b3763059376134f058cc337247c978add178b6ccdfb0019f
  ```

TLS Certificate

When importing CA certificates into EOS's CA certificate store, a verification process ensures that the certificates are valid x509 PEM files. However, certain attributes, such as the attribute in the Basic Constraints extension, are not checked during this step since their relevance to the certificate's specific usage is unknown.

EOS introduces a feature called "SSL Profiles" which governs the behavior of services, including eAPI, regarding the presentation of leaf certificates, the selection of trusted root certificates, the required attributes for certificates, and other TLS-related settings. When a certificate is designated as a trust certificate with the basic constraints CA restriction activated, the SSL Profile thoroughly validates all associated settings.

SSL Profiles offer an event log and an interactive CLI command for users to monitor the current status of profiles. The CLI command not only indicates whether a profile is valid or invalid but also provides detailed reasons for any invalid status. For instance, if a certificate is configured as a root of trust for validating client certificates but lacks the Basic Constraints CA set to "true," the command will highlight this discrepancy. Audit events are generated for both invalid and valid profiles.

Invalid commands are not outright rejected to prevent services from utilizing an invalid configuration. Instead, the SSL Profile ensures that such configurations are not used by the end service. To validate the correct configuration of a profile, you can use the command "show management security ssl profile <profile name>".

Signed certificate is imported into the switch by copying it to the certificate: folder. The certificate is identified by certificate file name. The certificate can be later updated by copying a new certificate file to it. Certificates can also be deleted. Execution of these commands is logged as follows.

- Command Logs

  Format: <time> <switch> <username> <user IP address> service=shell priv-lvl=15 cmd=copy <new certificate> <certificate file name>.

  Format: <time> <switch> <username> <user IP address> service=shell priv-lvl=15 cmd=delete <certificate file name>.

  Log Sample:

  ```
  2022 May 24 01:48:08 switch Aaa: %ACCOUNTING-6-CMD: admin vty3 10.95.66.234
  stop task _id=355 start_time=1558680488.84 timezone=EST service=shell priv-lvl=15
  cmd=copy usb1:TLSserv_certificate.pem certificate:TLSserv_certificate.pem <cr>
  ```

  ```
  2022 May 24 00:49:32 switch Aaa: %ACCOUNTING-6-CMD: admin vty3 10.95.66.234
  stop task_id=318 start_time=1558676972.75 timezone=EST service=shell priv-lvl=15
  cmd=delete certificate:TLSserv_certificate.pem <cr>
  ```

The above log will be followed by one of the following logs as appropriate to provide additional information.

- SSL_KEY_CERTIFICATE_IMPORTED

  Format: <time><switch> SSL [certificate <certificate file name>] has been imported with the SHA-256 hash of <hash value>.

  Log Sample:

  ```
  2022 May 29 09:17:29 switch ConfigAgent: %SECURITY-6-
  SSL_KEY_CERTIFICATE_IMPORTED: SSL certificate TLSserv_certificate.pem has been
  imported with the SHA-256 hash of
  c336aad7bf58403b9235c460de6c01a9de576c965bb86838be412fe570cd7145
  ```

- SSL_KEY_CERTIFICATE_DELETED

  Format: <time><switch> SSL [certificate <certificate file name>] has been deleted with the previous SHA-256 hash of <hash value>.

  Log Sample:

```
2022 May 29 09:18:35 switch ConfigAgent: %SECURITY-6-SSL_KEY_CERTIFICATE_DELETED: SSL certificate TLSserv_certificate.pem has been deleted with the SHA-256 hash of c336aad7bf58403b9235c460de6c01a9de576c965bb86838be412fe570cd7145
```

4.1.5 Password Reset

Resetting the password is done by changing the password with "username <username> secret 0 <new password>" command. The running of this command is logged (after removing the password string from it).

- Command Log

  Format: <time> <switch> <actor username> <IP address> service=shell priv-lvl=15 cmd=username <subject username> secret 0 *.

  Log Sample:

  ```
  2022 May 23 22:48:19 switch Aaa: %ACCOUNTING-6-CMD: admin vty3 10.95.66.234 stop task_id=283 start_time=1558669699.17 timezone=EST service=shell priv-lvl=15 cmd=username CCUser secret 0 * <cr>
  ```

4.2 FAU_GEN.1.2 - Audit Data Generation

Additional audit logs are generated as described below.

4.2.1 FCS_SSHC_EXT.1 - SSH Client Protocol

Switch acts as an SSH client to establish an SSH tunnel to the remote audit server. The following audit logs indicate failure to establish the SSH tunnel.

- SECURITY_SSH_TUNNEL_HOSTNAME

  Format: SSH tunnel <tunnel name> was unable to resolve hostname: <host to connect to>.

- SECURITY_SSH_TUNNEL_REMOTE_PORT_ERROR

  Format: SSH tunnel <tunnel name> is unable to open the port on the remote host.

- SECURITY_SSH_TUNNEL_LOCAL_PORT_ERROR

  Format: SSH tunnel <tunnel name> was unable to use the configured local port.

- SECURITY_SSH_TUNNEL_INITIAL_TIMEOUT

  Format: SSH tunnel <tunnel name> was unable to reach the remote host and the connection timed out.

- SECURITY_SSH_TUNNEL_CONNECTION_REFUSED

  Format: SSH tunnel <tunnel name> had its initial connection refused by the remote host.

- SECURITY_SSH_TUNNEL_HOSTKEY_VERIFY_FAILED

Format: SSH tunnel <tunnel name> was unable to connect to the configured host because it could not verify the hostkey of the remote host.

- SECURITY_SSH_TUNNEL_ALGORITHM_MISMATCH

  Format: SSH tunnel <tunnel name> was unable to connect to the configured remote server due to not finding a matching <algorithm>.

- SECURITY_SSH_TUNNEL_SWITCH_HOSTKEY_DENIED

  Format: SSH tunnel <tunnel name> was unable to log into its configured host via public-key authentication.

- SECURITY_SSH_TUNNEL_ESTABLISHED

  Format: SSH tunnel <tunnel name> from local TCP port <TCP port on switch> to <remote host>:<remote port> via <SSH server username>@<SSH server hostname> is established.

- SECURITY_SSH_TUNNEL_TIMEOUT

  Format: SSH tunnel <tunnel name> had the remote host timeout while connected.

- SECURITY_SSH_TUNNEL_CLOSED_REMOTELY

  Format: SSH tunnel <tunnel name> had it's connection closed by the remote host.

- SECURITY_SSH_TUNNEL_CONFIGURED

  Format: SSH tunnel <tunnel name> from local TCP port <TCP port on switch> to <Remote host>:<Remote Port> via <SSH server username>@<SSH server> is fully configured.

  (Note: This log generates when the tunnel is enabled via “no shutdown” command).

- SECURITY_SSH_TUNNEL_UNCONFIGURED

  Format: SSH tunnel <tunnel name> from local TCP port <TCP port on switch> to <Remote host>:<Remote Port> via <SSH server username>@<SSH server> has been unconfigured.

  (Note: This log generates when the tunnel is shut down via “shutdown” command).

Log Samples:

```
2022 May 23 21:51:11 switch SuperServer: %SECURITY-3-SSH_TUNNEL_INITIAL_TIMEOUT: SSH tunnel LogTunnel was unable to reach the remote host  and the connection timed out

2022 May 24 02:00:12 switch ConfigAgent: %SECURITY-6-SSH_TUNNEL_CONFIGURED: SSH tunnel LogTunnel from local TCP port 514 to localhost:514 via authuser@1.1.1.1 is fully configured

2022 May 24 02:00:09 switch ConfigAgent: %SECURITY-6-SSH_TUNNEL_UNCONFIGURED: SSH tunnel LogTunnel from local TCP port 514 to localhost:514 via authuser@1.1.1.1 has been unconfigured
```

4.2.2 FCS_SSHS_EXT.1- SSH Server Protocol

Switch acts as SSH server for the remote administrative session. The following audit logs are generated by the sshd process. These logs can be viewed by issuing the following command.

```
switch# show log system | grep sshd
```

Log generated when connection attempt starts:

- Format: <time> <switch> sshd[<PID>]: Connection from <remote IP address> port <port> on <switch IP address> port <port>.

From there, if algorithm negotiations fail, the following logs are generated:

- Format: <time> <switch> sshd[<PID>]: fatal: Unable to negotiate a key exchange method.
- Format: <time> <switch> sshd[<PID>]: fatal: no matching cipher found.
- Format: <time> <switch> sshd[<PID>]: fatal: no matching mac found.

Thereafter, depending on the type of authentication and its success or failure, the following logs are generated:

- Format: <time> <switch> sshd[<PID>]: <Accepted or Failed> publickey for <username> from <remote IP address> port <port>.
- Format: <time> <switch> sshd[<PID>]: <Accepted or Failed> keyboard-interactive/pam for <username> from <remote IP address> port <port>.

Note that the public key authentication is always tried first, so it is normal to see a public key authentication failed message before a password failure. If the authentication is successful, the following log is generated.

- Format: <time> <switch> sshd[<PID>]: pam_unix(sshd:session): session opened for user <username>.

Thereafter, when the session disconnects, the following audit log is generated.

- Format: <time> <switch> sshd[<PID>]: pam_unix(sshd:session): session closed for user <username>.

If the disconnection happens due to packet errors, there will be an additional audit log as follows indicating so before the session closure log. There may be a second line for this additional log that explains the reason for the failure, such as corrupted MAC on input, padding error, bad packet length etc.

- Format: <time> <switch> sshd[<PID>]: Disconnecting: Packet corrupt

Log Samples:

2022 May 31 23:36:42 switch sshd[7179]: Connection from 10.95.66.234 port 52921 on 172.30.167.171 port 22

2022 May 31 23:36:47 switch sshd[7179]: Accepted keyboard-interactive/pam for CCUser from 10.95.66.234 port 52921 ssh2

2022 Jun 1 00:05:15 switch sshd[7179]: pam_unix(sshd:session): session closed for user CCUser

4.2.3 FCS_TLSS_EXT.2 - TLS Server Protocol with Mutual Authentication

The following audit logs are created by nginx process when an eAPI client initiates a connection to TLS server in the switch. These logs can be viewed by issuing following command.

```
switch# show log system | grep nginx
```

When TLS connection attempt starts:

- Format: <time> <switch> nginx Starting TLS connection with remote client while SSL handshaking, client: <remote client IP>, server: [::]:<TLS server port>.

  Log Sample:

  2022 Jun 1 00:16:18 switch nginx: 2019/06/01 00:16:18 [info] 8888#0: *8 Starting TLS connection with remote client while SSL handshaking, client: ::ffff:10.95.66.234, server: [::]:443

If the connection is successful:

- Format: <time> <switch> nginx Successful client certificate authentication: x509 Subj:'CN = <subject name>' from source IP:<IP address>.

When the TLS connection is closed after reaching the point that valid certificate was presented:

- Format: <time> <switch> nginx[<PID>] TLS connection closed: x509 Subj:'<subject name>'.

If the TLS connection fails, the following audit log will be generated. The <TLS error> field will contain the information about the failure. The failure codes are as defined in TLS Alert Protocol (see RFC 8446 Appendix B.2).

- Format: <time> <switch> nginx[<PID>] SSL_do_handshake() failed (<TLS error>) while SSL handshaking, client:<IP address>.

  Log Sample:

  2022 Jun 1 00:16:18 switch nginx: 2019/06/01 00:16:18 [info] 8888#0: *6 SSL_do_handshake() failed (SSL: error:14094416:SSL routines:ssl3_read_bytes:sslv3 alert certificate expired:SSL alert number 45) while SSL handshaking, client: ::ffff:10.95.66.234, server: [::]:443

If authentication fails because the username provided in the certificate is not locally configured, the following log will be seen.

- Format: <time> <switch> nginx[<PID>] Unknown user: <username> attempted to authenticate via x509 client certificate.

4.2.4 FIA_AFL.1 - Authentication Failure Management

Upon multiple failed login attempts, the following log is generated.

- LOGIN_FAILED

  Format: user <username> failed to login [from: <source IP address>] [service: sshd] [reason: Account temporarily locked from remote access due to too many consecutive failed login attempts.]

  Log Sample:

  2022 May 23 21:49:06 switch Aaa: %AAA-4-LOGIN_FAILED: user CCUser failed to login [from: 10.95.66.234] [service: sshd] [reason: Account temporarily locked from remote access due to too many consecutive failed login attempts.]

4.2.5 FIA_UIA_EXT.1 - User Identification and Authentication

The following audit logs are generated for remote administrator login over SSH and for remote eAPI login over TLS (called command-api).

- AAA LOGIN

  Format: <time> <switch> user <username> logged in [from: <IP address>] [service: < sshd, command-api>]

  Log Sample:

  2022 May 23 21:17:53 switch Aaa: %AAA-5-LOGIN: user CCUser logged in [from: 10.95.66.234] [service: sshd]

- AAA LOGOUT

  Format: <time> <switch> user <username> logged out from [from: <IP address>] [service: <sshd, command-api>]

  Log Sample:

  2022 May 23 21:17:41 switch Aaa: %AAA-5-LOGOUT: user CCUser logged out [from: 10.95.66.234] [service: sshd]

- AAA FAILED

  Format: <time> <switch> user <username> logged in [from: <IP address>] [service: <sshd, command-api>] [reason: <reason why the login failed>]

  Log Sample:

  2022 May 23 21:26:06 switch Aaa: %AAA-4-LOGIN_FAILED: user CCUser failed to login [from: 10.95.66.234] [service: sshd] [reason: Authentication failed - Bad secret]

4.2.6 FIA_UIA_EXT.2 - Password-based Authentication Mechanism

The following audit logs are generated for local logins over the console interface. By definition, the console interface is local to the switch. So origin is not meaningful for the console login. As a result, “from” field is left blank.

- AAA LOGIN

  Format: <time> <switch> user <username> logged in [from:] [service: login]

- AAA LOGOUT

  Format: <time> <switch> user <username> logged out from [from:] [service: login]

- AAA FAILED

  Format: <time> <switch> user <username> logged in [from:] [service: login] [reason: <reason why the login failed>]

4.2.7 FIA_X509_EXT.1/Rev

See description of audit logs for FCS_TLSS_EXT.2.

4.2.8 FMT_MOF.1/ManualUpdate - Management of Security Functions Behavior

Updates are performed by either copying a new image file to the current .swi file on Flash or by pointing boot config to a different .swi filename on the Flash, and then rebooting the device.

When a new image file is copied to the current .swi file, the following log is generated.

- Command Log

  Format: <time> <switch> <username> <user IP address> service=shell priv-lvl=15 cmd=copy <source file name> <destination file name>.

  Log Sample:

  ```
  2022 May 24 23:26:38 switch Aaa: %ACCOUNTING-6-CMD: admin vty3 10.95.66.234
  stop task_id=2658 start_time=1558765598.5 timezone=EST service=shell priv-lvl=15
  cmd=copy flash:EOS.swi.orig flash:EOS.swi <cr>
  ```

If the copy succeeds, the following log is generated after the above copy operation log.

- NEW_SWI

  Format: <time> <switch> Boot image has been updated and has a SHA-512 hash of: <has value>.

  Log Sample:

  ```
  2022 May 24 23:28:21 switch ConfigAgent: %SYS-6-BOOT_NEW_SWI: Boot image has
  been updated and has a SHA-512 hash
  of:28395ca3c5c785654d2a02876426b1f987f1b7a796b8adabb52d7636b0a866c156f41f9
  08147b480bd265d62e29a221c5d26e69032d35cd3c982f0e493345ee9
  ```

If the copy fails, the following log is generated after the above copy operation log.

- COPY_ERROR

  Format: <time> <switch> An error occurred when copying from <source file name> to <destination file name> (<reason for failure>).

Log Sample:

```
2022 May 24 23:38:46 switch ConfigAgent: %SYS-3-FILE_COPY_ERROR: An error
occurred when copying from flash:EOS.swi.orig to flash:EOS.swi (No space left on
device).
```

When boot config is pointed to the new .swi file, the following audit log is generated.

- Command Log

  Format: <time> <switch> <username> <user IP address> service=shell priv-lvl=15 cmd=boot system flash:<filename>.

  Log Sample:

  ```
  2022 May 24 23:10:08 switch Aaa: %ACCOUNTING-6-CMD: admin vty3 10.95.66.234
  stop task_id=2604 start_time=1558764608.68 timezone=EST service=shell priv-lvl=15
  cmd=boot system flash:EOS.swi <cr>
  ```

  If the pointing did not succeed, error log is generated:

  ```
  2022 May 24 23:46:11 switch ConfigAgent: %SYS-3-
  BOOT_FAILED_UPDATE_BOOT_IMAGE: There was an issue with updating the boot
  image.
  ```

Thereafter actual update in performed by rebooting the switch. Reboot generates the following log.

- Command Log

  Format: <time> <switch> <username> <user IP address> service=shell priv-lvl=15 cmd=reload.

  Log Sample:

  ```
  2022 May 24 23:10:16 switch Aaa: %ACCOUNTING-6-CMD: admin vty3 10.95.66.234
  stop task_id=2605 start_time=1558764616.48 timezone=EST service=shell priv-lvl=15
  cmd=reload <cr>
  ```

4.2.9 FMT_SMF.1 - Specification of Management Functions

All CLI commands run by users are logged in the following format. As a result, all management activities performed on TSF data are logged.

- Command Logs

  Format: <time><switch> <username> <user IP address> service=shell priv-lvl=15 cmd=<command run>.

  Log Samples:

  ```
  2022 May 24 02:32:54 switch Aaa: %ACCOUNTING-6-CMD: CCUser vty3 10.95.66.234
  stop task_id=415 start_time=1558683174.38 timezone=EST service=shell priv-lvl=15
  cmd=banner login <cr>
  ```

  ```
  2022 May 24 02:36:18 switch Aaa: %ACCOUNTING-6-CMD: CCUser vty3 10.95.66.234
  ```

```
stop task_id=424 start_time=1558683378.22 timezone=EST service=shell priv-lvl=15
cmd=idle-timeout 10 <cr>
```

```
2022 May 24 02:59:52 switch Aaa: %ACCOUNTING-6-CMD: admin vty3 10.95.66.234
stop task_id=482 start_time=1558684792.37 timezone=EST service=shell priv-lvl=15
cmd=verify /sha512 flash:eos.swi <cr>
```

(For TOE update, see description of audit logs in FMT_MOF.1/ManualUpdate.)

```
2022 May 24 02:56:18 switch Aaa: %ACCOUNTING-6-CMD: admin vty3 10.95.66.234
stop task_id=479 start_time=1558684578.8 timezone=EST service=shell priv-lvl=15
cmd=aaa authentication policy lockout failure 3 window 300 duration 900 <cr>
```

(For cryptographic keys, see description of audit logs in FAU_GEN.1.1, Cryptographic Keys.)

```
2022 May 24 14:26:05 switch Aaa: %ACCOUNTING-6-CMD: CCUser vty3 10.95.66.234
stop task_id=970 start_time=1558733165.91 timezone=EST service=shell priv-lvl=15
cmd=cipher aes256-cbc aes128-cbc <cr>
```

```
2022 May 24 14:29:25 switch Aaa: %ACCOUNTING-6-CMD: CCUser vty3 10.95.66.234
stop task_id=987 start_time=1558733365.75 timezone=EST service=shell priv-lvl=15
cmd=cipher-list AES128-SHA256:AES256:SHA256:DHE-RSA-AES128-SHA256:DHE-RSA-
AES-256-SHA256 <cr>
```

(For time stamps, see description of audit logs in FPT_STM_EXT.1.)

(For certificate import, see description of audit logs in FAU_GEN.1.1, Cryptographic Keys.)

```
2022 May 24 16:04:50 switch Aaa: %ACCOUNTING-6-CMD: CCUser vty3 10.95.66.234
stop task_id=1311 start_time=1558739090.85 timezone=EST service=shell priv-lvl=15
cmd=trust certificate RootCA_certificate.pem <cr>
```

4.2.10 FPT_TUD_EXT.1 - Trusted Update

See description of audit logs in FMT_MOF.1/ManualUpdate.

4.2.11 FPT_STM_EXT.1 - Reliable Time Stamps

When a new time value is set by the administrator, the following logs are generated.

- Command Log

  Format: <previous time> <switch> <username> <user IP address> service=shell priv-lvl=15 cmd=clock set <new time> <new date>

  Log Sample:

  ```
  2022 May 23 21:56:17 switch Aaa: %ACCOUNTING-6-CMD: CCUser vty3 10.95.66.234
  stop task_id=264 start_time=1558666577.21 timezone=EST service=shell priv-lvl=15
  cmd=clock set 21:56:01 05/23/2019 <cr>
  ```

- Command Log

  Format: <time> <switch> <username> <user IP address> service=shell priv-lvl=15 cmd=clock timezone <new zone>

Log Sample:

```
2022 May 23 21:49:52 switch Aaa: %ACCOUNTING-6-CMD: CCUser vty3 10.95.66.234
stop task_id=259 start_time=1558666192.41 timezone=EST service=shell priv-lvl=15
cmd=clock timezone EST <cr>
```

4.2.12 FPT_SSL_EXT.1 - TSF Initiated Session Locking

The following audit log is generated upon termination of the remote SSH session due to inactivity.

- SESSION_IDLE_TIMEOUT

  Format: <time> <switch> Session for user <username> on service <login> terminated due to idle timeout.

  Log Sample:

  ```
  2022 May 23 21:40:29 switch SuperServer: %SECURITY-6-SESSION_IDLE_TIMEOUT:
  Session for user CCUser on service ssh terminated due to idle timeout.
  ```

4.2.13 FPT_SSL.3 - TSF Initiated Session Termination

The following audit log is generated upon termination of the remote SSH session due to inactivity.

- SESSION_IDLE_TIMEOUT

  Format: <time> <switch> Session for user <username> on service <ssh> terminated due to idle timeout.

  Log Sample:

  ```
  2022 May 23 21:40:29 switch SuperServer: %SECURITY-6-SESSION_IDLE_TIMEOUT:
  Session for user CCUser on service ssh terminated due to idle timeout.
  ```

4.2.14 FPT_SSL.4 - User Initiated Termination

When a user logs out of an interactive session, the following log is generated.

- Command Log

  Format: <time> <switch> user <user> logged out [from: <user IP address>] [service: <sshd, login>].

  Log Sample:

  ```
  2022 May 23 21:17:41 switch Aaa: %AAA-5-LOGOUT: user CCUser logged out [from:
  10.95.66.234] [service: sshd]
  ```

4.2.15 FPT_ITC.1 - TSF Initiated Session Locking

For a trusted channel over SSH Tunnel to the audit server, see description of audit logs for FCS_SSHC_EXT.1.

For trusted channels over TLS from an eAPI client, see description of audit logs for FCS_TLSS_EXT.2 and FIA_X509_EXT.1/Rev.

4.2.16 FPT_TRP.1/Admin - Trusted Path

For trusted path over SSH from human interactive user to TOE, see description of audit logs for FCS_SSHS_EXT.1.

4.2.17 Verifying FIPS mode is enabled and the FIPS POST was performed

To verify that FIPS mode has been enabled for SSH and TLS connections there is a procedure to perform for each connection type.

For SSH connections there will be a log message generated before the information on the source of the remote connection. The log message indicates the FIPS POST was performed. This message will appear before each new connection:

```
Jun  1 11:22:23 switch sshd[32499]: FIPS mode initialized
Jun  1 11:22:23 switch sshd[32499]: Connection from 10.95.66.234 port 63766 on
172.30.167.171 port 22
```

For TLS connections, verifying the FIPS POST requires bash shell access. This means that the "admin" account must be used to enter "debug mode" and validate the presence of the log indicating FIPS mode. The following procedure is carried out:

- Enter bash mode
- Restart the nginx server via the "service" command. This is because the FIPS POST is run when the nginx server starts and begins to serve TLS commands.
- Validate the "FIPS" log appears in /var/log/error.log
- Validate the PID for the log matches that of the current nginx instance.

An example of doing so is shown below. The pid of the most recent log and the matching pid of the nginx instance are shown highlighted in yellow. After carrying out this procedure the "admin" account should be logged out of.

```
switch(config)#bash
Arista Networks EOS shell

[admin@switch ~]$ sudo service nginx restart
Stopping nginx:                                    [  OK  ]
Starting nginx:                                    [  OK  ]
[admin@do401 ~]$ sudo cat /var/log/error.log
2019/09/25 09:01:26 [alert] 4354#0: FIPS mode enabled for openssl
2019/09/25 10:11:10 [alert] 6718#0: FIPS mode enabled for openssl
2019/09/25 10:14:50 [alert] 6921#0: FIPS mode enabled for openssl
[admin@switch ~]$ ps aux | grep nginx
root      6921  0.0  0.0  11904   816 ?        Ss   10:14   0:00 nginx: master process
/usr/sbin/nginx -c /etc/nginx/nginx.conf -g pid /var/run/nginx.pid;
```

```
nobody    6922  0.0  0.2  19576  8956 ?       S    10:14   0:00 nginx: worker process
admin     6963  0.0  0.0   4992  1632 pts/4   S+   10:15   0:00 grep --color=auto nginx
```

4.2.18 Verifying Integrity of Software and FIPS self-tests on Startup

Verifying the integrity of the software and FIPS self-tests during startup requires bash shell access. This means that the "admin" account must be used to enter "debug mode" and validate the presence of the logs indicating FIPS mode and successful software verification at startup. The following procedure is carried out:

- Enter bash mode
- Validate "SWI verification successful" shows up in /var/log/messages
- Validate FIPS-related logs show up in /var/log/messages

Examples of the expected log messages are shown below. After carrying out this procedure, log out of the admin account.

Logs indicating verification of the integrity of the software on the TOE as well as FIPS tests on startup can be seen as follows in /var/log/messages:

```
Jan 25 13:38:24 al207 root: Image type: swim version: BLESSED=1
BUILD_DATE=20220413T053119Z BUILD_HOST=dhcp-242-170-
100.sjc.aristanetworks.com IMAGE_FORMAT_VERSION=3.0 SERIALNUM=cc8af7a4-3c5b-
4f00-bf78-68394d62ceff SWI_ARCH=i686 SWI_FLAVOR=DEFAULT
SWI_MAX_HWEPOCH=2 SWI_OPTIMIZATION=Unoptimized
SWI_RELEASE=26924507.4280F SWI_VARIANT=US SWI_VERSION=4.28.0F SWI
verification successful.
```

Here the "SWI verification successful" message indicates that the integrity check was done over the software, and it passed.

If this step was unsuccessful, a different message may be printed depending on the reason for SWI verification failure. "SWI verification failed" will be printed if the hash of the SWI does not match the hash in the signature file. If such failure messages are seen, immediately remove the switch from the network and contact the Arista TAC team.

To verify the FIPS cryptographic self-tests were performed at startup, see the audit log in /var/log/messages:

```
Dec  7 17:46:56 hrm101 sshd[4155]: FIPS mode initialized
```

This indicates FIPS has been initialized, so all the self-tests have been run and passed.

There are also additional audit logs also in /var/log/messages that provide more information about the cryptographic self-tests that were run, for example:

```
Jan 25 13:39:14 al207 arista-python: OpenSSL: Key derivation service for TLS performed
```

The above audit log indicates that the TLS KDF cryptographic self-test was performed.

If these messages were not observed, or if there are any occurrences of OpenSSL FIPS-related errors in the log messages, such as:

```
Failed to enable FIPS mode

140737353054080:error:2D06B06F:FIPS
routines:FIPS_check_incore_fingerprint:fingerprint does not match:fips.c:271:
```

The error message suggests that there has been a failure in the self-tests of the FIPS, mode during startup. Depending on the specific error message, the failure could be related to either the integrity test or the Known Answer Tests (KAT).

For instance, if the error message states: **"fingerprint does not match,"** it indicates an integrity test failure. On the other hand, if the error message displays **"self-test failed,"** it signifies a failure in the KAT self-test.

To address the issue and recover the device, start by attempting a reboot. This may resolve the problem and allow FIPS mode to initialize properly. However, if the issue persists and FIPS mode fails to initialize, it is advisable to disconnect the device from the network and contact the Arista Technical Assistance Center (TAC) team for further assistance and guidance in resolving the problem. They will be able to provide specific troubleshooting steps or recommend appropriate actions to rectify the FIPS mode self-test failure.

Pairwise consistency tests are conducted during the generation of RSA key pairs. If the pairwise consistency test yields a failure, the attempt to create an RSA key pair will be unsuccessful, resulting in an error message stating "**pairwise test failed.**"

To resolve this situation, it is necessary to discard the inputs used in the previous RSA key pair generation attempt. Subsequently, new inputs should be generated, and another attempt can be made to create the RSA key pair.

If the failure persists despite multiple attempts, it is recommended to contact the Arista Technical Assistance Center (TAC) team.

Finally, the TOE performs continuous testing of random number generation during attempts to gather additional entropy. If the entropy source appears to be malfunctioning and fails to provide sufficiently random bits, an error message labeled "**entropy source stuck**" will be generated.

If this error message persists repeatedly, appearing multiple times in succession, it is advisable to reach out to the Arista Technical Assistance Center team for further support and resolution.