Technical White Paper

aruba
a Hewlett Packard Enterprise company

ARUBA CX HARDENING GUIDE

AOS-CX 10.7

Contents

Revision History

Document Version	Reason for Change	Revision Date
1.0 (Initial Release)		Jun 2021

Overview

Security is a growing concern in today's all-digital enterprise infrastructure. Upper level managers and IT administrators alike are held to higher accountability for the integrity and availability of their critical data and infrastructure. While host clients and servers are often the focus of security discussions, the security of network devices such as switches, routers, and wireless access points should not be ignored. Critical enterprise data traverses these devices, and properly securing them is paramount to a stable and secure infrastructure.

The Aruba CX switching platform, powered by the AOS-CX network operating system, simplifies network operations by delivering automation, distributed analytics, security, and high availability to campus and data center networks. The microservices architecture around which AOS-CX is built delivers network-wide analytics and full programmability to enable complete network assurance.

The purpose of this document is to provide security guidelines and best practices for management features and protocols provided by the AOS-CX software, and to present sample configurations to illustrate these best practices in action. This document is not intended to be a comprehensive reference guide to the features and commands listed; for additional information on configuration syntax and advanced features referred to in this document, please obtain the latest software manual set from the Aruba Support Portal.

Hardening objectives

IETF BCP 61 points to a few definitions that help us define our goals, which we can summarize into three helpful points:

- Authentication: A security service that verifies an identity. This identity could be a user, a device, or a process.
- Data Confidentiality: A security service that protects data against unauthorized disclosure to unauthorized individuals or processes.
- Data Integrity: A security service that protects against unauthorized changes to data. Changes include intentional change and accidental change.

The applications and procedures we use in this document leverage these summarized definitions above and help to shape the following general guidelines:

- If there are methods we can use to ensure the identities of the users and devices with which we interact, we should prefer these over insecure alternatives.
- We should limit the exposure to the equipment from sources we cannot trust, whenever possible. We should also make attempts to utilize encryption methods so that our data is not easily read by anyone besides the trusted receiver of the data.
- Assume that eventually, an event will occur that causes a need for reliable information we know we can trust. We need to make sure this data is safe, available for us to access, and unavailable to anyone else.

Operational assumptions

- One or more authorized administrators are assigned who are competent to manage the device and the security of the information it contains, trained for the secure operation of the device, and who can be trusted not to deliberately abuse their privileges so as to undermine security.
- Authorized users are trusted to correctly install, configure and operate the device according to the instructions provided by the device documentation.
- There will be no untrusted users and no untrusted software on component servers.
- The switch must be installed in a physically secure area where only authorized administrators have access to the physical device.
- Users will protect their authentication data.

Syntax and conventions

This document provides examples for each configurable feature discussed. These examples follow a common format: commands and fixed options appear as fixed-width regular text, while *configurable* parameters appear in italics, as in the

following:

```
switch(config)# ssh server vrf default
```

For more details on command syntax, refer to the documentation referenced for each feature, or use the built-in command syntax help on the switch by typing a partial command, then typing ? (question mark) to see possible options and parameters for that command.

Software, documentation, and security advisories

Aruba CX switch software, release notes, and user documentation can be found at the Aruba Support Portal (ASP).

Security advisories are published on the Aruba Security Advisory archive, and notification services are provided by a Security Alerts mailing list, with subscriptions offered via the self-service portal.

Hardening Aruba CX switches

Factory defaults

Once the device boots, it is essential for an administrator to immediately connect to it and configure a password for the admin account. California signed into law bill SB-327 in 2018, requiring manufacturers of networking equipment to force users to create a password when they first connect to a device.

In a factory default state, AOS-CX devices are configured with the default user **admin** with no password. The user is prompted to create a password before access is given to the CLI:

```
Please configure the 'admin' user account password.
Enter new password: *****
Confirm new password: *****
```

The built-in management interface provides a way to access and manage the switch that is segregated from production traffic. Internal networks separated from production traffic are typically referred to as Out-Of-Band-Management networks (OOBM). By limiting the clients allowed to manage devices to only those who reside on the OOBM network, we sharply limit the universe of devices that can attempt to control the device.

In AOS-CX, the management interface is logically separated from the rest of the switch by means of a unique virtual routing and forwarding table (VRF), named the **mgmt** VRF. Please note that the **mgmt** VRF is unique in that it is permanently assigned to the physical management port and cannot be associated with any other switch interface; the management port itself cannot be associated with any other VRF.

Note: 6100 switch models do not have a dedicated management port or the associated **mgmt** VRF; when configuring supported features and settings referred to in this guide on a 6100 switch model, substitute any references to the **mgmt** VRF with the **default** VRF.

The management interface is enabled by default to learn an IP address via DHCP. To configure the management interface with a static IP address, gateway, and DNS:

```
switch(config)# interface mgmt
switch(config-if-mgmt)# ip static 10.1.1.5/24
switch(config-if-mgmt)# default-gateway 10.1.1.1
switch(config-if-mgmt)# nameserver 10.0.1.10 10.0.1.11
```

To show the status of the management interface:

```
switch# show interface mgmt
Address Mode : static
Admin State : up
Mac Address : d0:67:26:11:11:11
IPv4 address/subnet-mask : 10.1.1.5/24
Default gateway IPv4 : 10.1.1.1
```

```
IPv6 address/prefix :
IPv6 link local address/prefix : fe80::d267:2611:1111:1111/64
Default gateway IPv6 :
Primary Nameserver : 10.0.1.10
Secondary Nameserver : 10.0.1.11
```

The other VRF available on an AOS-CX device upon first boot is the **default** VRF. The **default** VRF is automatically associated with all non-management interfaces, including Layer 3 routed ports, non-routed ports, and switched VLAN interfaces (SVIs) created on the switch, unless the interface is explicitly attached to another VRF.

The following management services are enabled by default on an Aruba CX switch:

- SSH on TCP port 22
- HTTP/HTTPS and read/write REST API on TCP ports 80 and 443

Aruba CX 8320, 8325, 8360, and 8400 switches ship with these services enabled *only* on the **mgmt** VRF, while 6200, 6300, and 6400 switches ship with these services enabled on both the **default** and **mgmt** VRFs. As the 6100 series does not have a dedicated management port or the associated VRF, management services are enabled only on the **default** VRF.

For optimal security, Aruba strongly recommends managing switches from a dedicated management network, when possible, and disabling management services on all other VRFs.

Restoring switch to factory defaults

The recommended method to return an Aruba CX switch to factory default settings is to **zeroize** it. The following occurs when the zeroization process is initiated:

- The switch reboots to ServiceOS
- Primary and secondary software image files are backed up to memory from flash storage
- The entire flash storage device is overwritten with zeroes to securely erase all stored data
- The flash storage device is reformatted with a factory default filesystem
- Backed up software image files are written to flash in their original locations
- The switch reboots to the primary software image with a default configuration

There are four methods that may be used to zeroize a switch. First, an **admin** user may use the `erase all zeroize` command from the AOS-CX CLI:

```
switch# erase all zeroize
This will securely erase all customer data and reset the switch
to factory defaults. This will initiate a reboot and render the
switch unavailable until the zeroization is complete.
This should take several minutes to one hour to complete.
Continue (y/n)?
```

Second, an **admin** user may use the `erase zeroize` command from the ServiceOS CLI:

```
SVOS> erase zeroize
############################WARNING############################
This will securely erase all customer data and reset the switch
to factory defaults. This will initiate a reboot and render the
switch unavailable until the zeroization is complete.
This should take several minutes to one hour to complete.
############################WARNING############################

Continue (y/n)?
```

Third, a user with physical access to the switch front panel and a FAT32-formatted USB storage device may zeroize the switch from the ServiceOS login prompt by entering the username **zeroize** and following the provided instructions:

```
ServiceOS login: zeroize
```

```
This will securely erase all customer data, including passwords, and
reset the switch to factory defaults.
This action requires proof of physical access via a USB drive.
  * Create a FAT32 formatted USB drive
  * Create a file in the root directory of the USB drive named zeroize.txt
  * Type the following serial number into the zeroize.txt file: xxxxxxxxxx
  * Insert the USB drive into the target module
  * Confirm the following prompt to continue

Continue (y/n)?
```

Finally, changing the switch security mode results in the switch being zeroized; see the **Enhanced security mode** section for more information.

System settings and services

Enhanced security mode

AOS-CX provides two security modes that control access to certain system management features — *standard* and *enhanced*. All Aruba CX switches operate in standard mode by default, with no system-level restrictions in place for any functionality. The enhanced security mode disables access to the `start-shell` command in the AOS-CX CLI, as well as the ServiceOS commands `config-clear`, `password`, `sh`, and `update`.

Changing the switch security mode is performed from the ServiceOS shell, which requires a console connection to the switch. All changes to the switch security mode (standard to enhanced, or enhanced to standard) result in zeroization of the filesystem and a reset to factory defaults.

Reboot the switch to ServiceOS using the following command:

```
switch# boot system serviceos
One time boot to ServiceOS initiated.
Checking if the configuration needs to be saved...

This will reboot the system to ServiceOS and render
 the entire switch unavailable.
Access to ServiceOS is only available through the serial console.
Continue (y/n)?
```

Once the switch has rebooted and the ServiceOS login prompt is displayed, login as **admin** (no password is set by default). Use the following command to enable *enhanced* security mode:

```
SVOS> secure-mode enhanced
##############################WARNING##############################
This will set the switch into enhanced secure mode.  Before
enhanced secure mode is enabled, the switch must securely erase
all customer data and reset the switch to factory defaults.
This will initiate a reboot and render the switch unavailable
until the zeroization is complete.
##############################WARNING##############################

Continue (y/n)?
```

Entering **y** will cause the switch to reboot, zeroize the filesystem, then reboot an additional time.

To revert to the *standard* security mode, reboot to ServiceOS as above, login as **admin**, then use the following command:

```
SVOS> secure-mode standard
##############################WARNING##############################
This will set the switch into standard secure mode.  Before
standard secure mode is enabled, the switch must securely erase
```

```
all customer data and reset the switch to factory defaults.
This will initiate a reboot and render the switch unavailable
until the zeroization is complete.
############################WARNING#############################

Continue (y/n)?
```

ServiceOS password authentication

By default, the ServiceOS shell (accessible only from the local switch console port) requires no password to login as **admin**; to enable password authentication for ServiceOS, use the following command from the configuration context:

```
switch(config)# system serviceos password-prompt
```

When this setting is enabled, logging in to the ServiceOS shell with the **admin** user requires the same password used to authenticate the **admin** user in the AOS-CX CLI or Web UI.

If this setting is enabled, a forgotten **admin** user password cannot be reset using ServiceOS; if there are no other local or RADIUS/TACACS user accounts with administrator-level access, the switch must be zeroized by entering the username **zeroize** at the ServiceOS login prompt to restore administrator access. See **Restoring switch to factory defaults** for more information.

Time synchronization

Many secure protocols and auditing functions rely on system times being synchronized with a reliable time source, either within or (where security considerations permit) external to the managed network. One of the most commonly-used protocols to accomplish this is the Network Time Protocol (NTP), which can use both local and Internet-hosted servers to synchronize system time across a network. NTP should be configured and enabled on the device prior to enabling secure management protocols.

A common practice among organizations that span multiple time zones is to use NTP to synchronize time clocks and set the local time zone on all equipment to UTC. This practice aids in troubleshooting and security audits for devices that might be continents apart.

To configure a switch to use NTP authentication and connect to a local NTP server at 10.100.1.254 using the switch management port:

```
switch(config)# ntp authentication
switch(config)# ntp authentication-key 1 md5 ntpauthkey
switch(config)# ntp server 10.100.1.254 prefer
switch(config)# ntp vrf mgmt
```

To set the time zone to UTC:

```
switch(config)# clock timezone utc
```

Event and security logging

AOS-CX creates local event and security logs and can use the syslog protocol to forward these logs to a remote server for analysis and auditing purposes. The syslog client uses UDP for syslog communication as a default, though TCP and TLS are also supported.

When configuring AOS-CX to send logs to a remote server, it is common practice to set a facility value. This value acts as a label that the remote server can use to determine which file the syslog message should get appended.

Below is an example of how to configure AOS-CX to send event log messages via syslog to a remote server. This example uses the default facility of local7 and sends event messages marked informational and higher:

```
switch(config)# logging 10.100.1.250 vrf mgmt
```

To include security-related accounting logs in addition to the event logs, then add the *include-auditable-events* option to the

configuration:

```
switch(config)# logging 10.100.1.250 include-auditable-events vrf mgmt
```

Login banner

RFC 4252, which defines the SSH protocol, says, "In some jurisdictions, sending a warning message before authentication may be relevant for getting legal protection" This is prudent advice. Setting a banner that displays a message during the login process will notify clients that unauthorized use is prohibited and that access to and use of the system may be monitored and logged.

The following is an example of creating a "message of the day" (MOTD) banner that will get displayed when a user connects to the switch before logging in (using the ^ character to denote the end of the banner):

```
switch(config)# banner motd ^
Enter a new banner. Terminate the banner with the delimiter you have chosen.
switch(config-banner-motd)# This system is for authorized use only. Unauthorized or improper
switch(config-banner-motd)# use of this system may result in civil or criminal penalties. By
switch(config-banner-motd)# continuing to use this system you acknowledge your consent to
switch(config-banner-motd)# these conditions of use.
switch(config-banner-motd)# ^
```

Simple Network Management Protocol (SNMP)

SNMP is used to manage and monitor networked devices from a centralized platform. There are three versions of the SNMP protocol: v1, v2c, and v3. SNMPv1 and v2c use community names for read and write access, much like passwords are used for authentication; these community names are sent across the wire as clear text. If a malicious user were to capture these community names, they could potentially issue SNMP *set* commands to make unauthorized and potentially harmful configuration changes to a network device. SNMPv3, by comparison, utilizes a user-based security model with both authentication and privacy protocols to prevent unauthorized access or eavesdropping of management traffic.

SNMP is disabled by default on all AOS-CX devices. When enabled, SNMP on AOS-CX 10.7 provides limited write support in addition to read-only access and trap support for SNMP v1, v2c, and v3.

The default SNMP community string is *public*, a common setting for SNMP-capable devices. Replace the *public* community string with another value that is hard to guess:

```
switch(config)# snmp-server community meatball
```

The default access level for SNMP communities in AOS-CX 10.7 is read-only; if read-write support is required, set the access level for the community to *rw* from the community context accessed from the previous command:

```
switch(config-community)# access-level rw
```

IPv4 and/or IPv6 ACLs may be used to limit access to allowed management stations or subnets; only one ACL (IPv4 or IPv6) may be applied to a community at a time. Apply an IPv4 or IPv6 ACL from the SNMP community context as with the access-level command above:

```
switch(config-community)# access-list ipv4 snmpacl
```

Aruba strongly recommends using SNMPv3 over older versions of SNMP. Older versions of SNMP are unauthenticated and unencrypted, with the community string acting as a password, transmitted in plaintext. SNMPv3, meanwhile, offers support for users, authentication, and strong encryption.

Create an SNMPv3 user using SHA for authentication and DES for privacy:

```
switch(config)# snmpv3 user myUser auth sha auth-pass plaintext myAuthPswrd priv des priv-pass
plaintext myPrivPswrd
```

Next, create an SNMPv3 context with the community name created above and assigned to the **mgmt** VRF:

```
switch(config)# snmpv3 context snmpv3mgmt vrf mgmt community meatball
```

Disable support for SNMPv1 and SNMPv2c and only accept SNMPv3 messages using the following command:

```
switch(config)# snmp-server snmpv3-only
```

Finally, enable SNMP on the **mgmt** VRF:

```
switch(config)# snmp-server vrf mgmt
```

Control plane ACLs

Once an IP address is bound to an interface associated with a VRF, the AOS-CX device may become exposed to management access from untrusted users or devices. This potential point of vulnerability can be mitigated at the switch configuration level by binding an Access Control List (ACL) to the control plane for that VRF. The control plane handles the device's management and routing functionality.

Once a control plane ACL is applied to a VRF, it filters packets to all IP addresses bound to the device on that VRF. It is possible to create a control plane ACL for each existing VRF, including the **mgmt** VRF.

Below is an example of an ACL an administrator can apply that limits SSH and SNMP control plane access to source devices with IP addresses in the 10.10.0.0/24 subnet, with counters for denied SSH and SNMP packets:

```
switch(config)# access-list ip CONTROLPLANE
switch(config-acl-ip)# 05 comment ALLOW SSH AND SNMP ON ADMIN SUBNET, BLOCK ALL OTHERS
switch(config-acl-ip)# 10 permit tcp 10.10.0.0/24 any eq 22
switch(config-acl-ip)# 20 permit udp 10.10.0.0/24 any eq 161
switch(config-acl-ip)# 30 permit udp 10.10.0.0/24 any eq 162
switch(config-acl-ip)# 40 deny tcp any any eq 22 count
switch(config-acl-ip)# 50 deny udp any any eq 161 count
switch(config-acl-ip)# 60 deny udp any any eq 162 count
switch(config-acl-ip)# 990 comment ALLOW ANYTHING ELSE
switch(config-acl-ip)# 1000 permit any any any
```

Note: Logging of matched packets is not supported for control plane ACLs.

To apply this ACL to the **default** VRF:

```
switch(config)# apply access-list ip CONTROLPLANE control-plane vrf default
```

All ACLs in AOS-CX have an implicit "deny any" rule at the end of the rules list; this requires that allowed traffic be *explicitly* permitted to pass through an applied ACL. In the above example, SSH and SNMP traffic on ports 22 is allowed from 10.10.0.0/24. The same traffic is then blocked from any other subnets. Finally, all other traffic is permitted.

Secure file transfers

TFTP has been around for over thirty years and is still commonly used for device firmware management, backups, and the transferring of support-files between the device and a remote server. However, it offers no authentication or encryption and is an unreliable file transfer method as it is a UDP based protocol.

SFTP is preferred over TFTP because it is authenticated, encrypted, and more reliable since it uses TCP.

For example, to copy a software image from an SFTP server to the primary boot bank of an AOS-CX device, using the management interface:

```
switch# copy sftp://meatball@10.10.10.1/ArubaOS-CX_6400-6300_10.07.0005.swi primary vrf mgmt
```

The above command assumes the file “ArubaOS-CX_6400-6300_10.07.0005.swi” is in the user **meatball**’s home directory. If the file exists instead in a different directory outside of the users’ home directory, specify to use the absolute path to the file by providing two forward-slashes (//) before specifying the absolute path to the file:

```
switch# copy sftp://meatball@10.10.10.1//swimages/ArubaOS-CX_6400-6300_10.07.0005.swi primary
```

```
vrf mgmt
```

Authentication, Authorization, and Accounting (AAA)

AOS-CX allows the configuration of external servers based on RADIUS or TACACS+ to authenticate switch management users. The authentication method configured on the device can be configured to be universal to all access methods (console, SSH, WebUI).

If the primary authentication method fails, a secondary method can be used to authenticate users. For example, it is common practice to configure authentication to fall back to local user accounts if the device cannot connect to the TACACS+ server.

AOS-CX permits three methods of authenticating management users:

- Local: Uses usernames and passwords stored locally in the switch database.
- RADIUS: Uses one or more RADIUS servers to authenticate users.
- TACACS+: Uses one or more TACACS+ servers to authenticate users.

Local authentication

Local user accounts can be assigned to one of two user groups to provide different levels of access to the switch:

- Administrators—full access (privilege level 15)
    - Perform firmware upgrades
    - Make configuration changes
    - View all switch configuration information, including sensitive data such as ciphertext passwords
    - Add and remove local user accounts, and change user passwords
    - All REST interface methods (GET, PUT, POST, PATCH, DELETE) can be used
- Operators – limited access (privilege level 1)
    - Display-only CLI access
    - View non-sensitive configuration information
    - Only the REST interface GET method can be used

Local usernames and passwords are configured on a per-switch basis and provide the most basic form of authentication. Local authentication is often used as the fallback login method to provide a minimum security level should the primary method fail, without completely disabling management access to the switch.

To configure a local administrator-level user named *localadmin* with interactive password entry:

```
switch(config)# user localadmin group administrators password
Enter password: **********
Confirm password: **********
```

To create an operator-level user named *localoperator* with a plaintext password:

```
switch(config)# user localoperator group operators password plaintext I-h@ve-Read-0n1y-Acc3ss!
```

An administrator can also enter a password as a ciphertext string rather than being entered in plaintext. In AOS-CX, ciphertext passwords cannot be generated manually; they must be copied from another switch with the same export password configured.

To set the export password on the switch:

```
switch(config)# service export-password
Enter password: *************
Confirm password: *************
```

Once the export passwords on the source and destination switches are the same, copy the ciphertext password from the source switch and apply it to the destination:

```
switch(config)# user localadmin group administrators password ciphertext myCipherText
```

RADIUS

Authenticating users through RADIUS provides a centralized way to manage access to the switch. Aruba ClearPass Policy Manager supports RADIUS authentication.

In the following example, a RADIUS server at IP address 10.100.0.253, with the authentication key "R@d1us$erv3rkey", is configured to be used for authentication on the switch:

```
switch(config)# radius-server host 10.100.0.253 key plaintext R@d1us$erv3rkey
```

To enable RADIUS authentication for switch login as the primary authentication method, with local authentication as the secondary method, use the following configuration command:

```
switch(config)# aaa authentication login default group radius local
```

TACACS+

Authenticating users through TACACS+ provides a centralized way to manage access to the switch. TACACS+ authentication works along the same lines as RADIUS authentication, allowing the administrator to manage users from a central server. Aruba ClearPass Policy Manager also supports TACACS+ authentication.

Similar to the RADIUS example above, the following command designates a TACACS+ server at 10.100.0.252, with the authentication key "T@cac$serv3rkey", as an authentication server:

```
switch(config)# tacacs-server host 10.100.0.252 key plaintext T@cac$serv3rkey
```

To enable TACACS+ authentication as the primary method and local authentication as the secondary method for management access, use the following configuration command:

```
switch(config)# aaa authentication login default group tacacs local
```

TACACS+ also allows for commands authorization and accounting, permitting more granular control of command-line access and allowing managers to monitor user sessions and configuration activity on their devices.

The following commands enable authorization and accounting, handled by servers in the default tacacs group:

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

Minimum password length

Device administrators can specify a minimum password length to reduce the probability that management user passwords can be brute-forced to access devices.

To require a minimum password length of 12 characters:

```
switch(config)# aaa authentication minimum-password-length 12
```

CLI session settings

Administrators may set a maximum number of concurrent CLI management sessions per user, as well as a session timeout that will automatically close a CLI session after a period of inactivity.

By default, AOS-CX allows up to 5 concurrent CLI sessions for each user account. This value can be changed from the cli-session context with the `max-per-user` command; to set a maximum of 2 concurrent sessions per user:

```
switch(config)# cli-session
switch(config-cli-session)# max-per-user 2
```

Connected CLI session users can be automatically disconnected after a specified period of inactivity. By default the CLI session timeout is set to 30 minutes, and can be changed from the *cli-session* context using the `timeout` command. To set a timeout period of 5 minutes:

```
switch(config-cli-session)# timeout 5
```

The session timeout period can be set between 0 (no timeout) and 43,200 minutes (30 days). The timer only applies to CLI sessions, not the Web UI.

Limiting shell access

The AOS-CX operating system provides access to the underlying Linux system, allowing administrators to launch a **Bash** shell session from the switch command line. By default, all accounts that are part of the **administrators** group can access the shell using the `start-shell` command, and can then use the `sudo` command to execute shell commands with root permissions. Misuse of shell access could result in disclosure of sensitive network traffic to an unauthorized third party through packet mirroring to a remote device, or could cause a denial of service by modifying or removing system files and rendering the device unbootable, requiring software restoration through the ServiceOS console.

Access to the Bash shell can be completely disabled by changing the switch security mode to *enhanced* from ServiceOS; see the section **Enhanced security mode** for more information.

Another method to limit shell access would be to use an external TACACS+ authorization server and deny access to the `start-shell` command to all users *except* those who specifically require it. For further information on using TACACS+ to implement command authorization, refer to the documentation for your preferred TACACS+ software platform.

Attack mitigation

Control Plane Policing (CoPP)

Note: The CX 6100 Switch Series supports a smaller number of configurable CoPP traffic classes, and does not have configurable 'burst' values.

Control Plane Policing prevents flooding of certain types of packets from overloading the switch or module CPU by either rate-limiting or dropping packets. The switch software provides a number of configurable classes of packets that can be rate-limited, including (but not limited to) ARP broadcasts, multicast, routing protocols (BGP, OSPF), and spanning tree. CoPP is always active on AOS-CX, and cannot be disabled.

The following default CoPP policy applies the following traffic class definitions and rate limits (in packets per second) on a 6300 series switch:

```
switch# show copp-policy default
class                 drop priority rate pps burst pkts hardware rate pps
--------------------- ---- -------- -------- ---------- -----------------
acl-logging              0        25       25         25
arp-broadcast            2      1250     1250       1250
arp-protect              2      2075     2075       2075
arp-unicast              3       825      825        825
bfd-control              5       850      850        850
bgp                      5       750      750        750
captive-portal           2      2075      259       2075
client-onboard           5      1024     1024       1000
dhcp                     2       750      750        750
erps                     6       225      225        225
icmp-broadcast-ipv4      2       325      325        325
icmp-multicast-ipv6      2       475      475        475
icmp-security-ipv6       2       325      325        325
icmp-unicast-ipv4        3       225      225        225
icmp-unicast-ipv6        3       400      400        400
ieee-8021x               2      2075      259       2075
igmp                     4      1600      450       1600
ip-exceptions            0       100      100        100
ip-lockdown              0       100      100        100
ip-tracker               0       256      256        250
ipsec                    5      1025      128       1025
```

```
ipv4-options                1       100     100     100
lacp                        5       2050    2050    2050
lldp                        5       100     100     100
loop-protect                6       225     225     225
mac-lockout                 0       100     100     100
manageability               4       4218    4218    4200
mdns                        2       150     150     150
mirror-to-cpu               0       100     100     100
mld                         4       1600    450     1600
mvrp                        5       225     225     225
nae-packet-monitor          0       100     200     100
ntp                         4       100     100     100
ospf-multicast              5       1025    1025    1025
ospf-unicast                5       1025    1025    1025
pim                         5       1700    1700    1700
secure-learn                2       2075    259     2075
sflow                       1       1000    125     1000
stp                         6       1500    1500    1500
udld                        6       450     450     450
unknown-multicast           1       1025    128     1025
unresolved-ip-unicast       1       325     325     325
vrrp                        4       400     400     400
default                     2       4225    528     4225
```

The default CoPP policy can be modified, but cannot be deleted. To revert a modified default CoPP policy to factory default settings, use the following command:

```
switch(config)# copp-policy default revert
```

Administrators may create up to 32 custom CoPP policies, though only one can be active at any given time. The following commands demonstrate the creation of a simple custom CoPP policy, and how it is applied to the switch:

```
switch(config)# copp-policy copp_policy_01
switch(config-copp)# class arp-broadcast priority 2 rate 1000 burst 1000
switch(config-copp)# class unknown-multicast priority 2 rate 1000 burst 1000
switch(config-copp)# class unresolved-ip-unicast priority 2 rate 1000 burst 1000
switch(config-copp)# default-class priority 1 rate 3000 burst 3000
switch(config-copp)# exit
switch(config)# apply copp-policy copp_policy_01
```

To remove a custom CoPP policy from service and automatically apply the default policy:

```
switch(config)# no apply copp-policy copp_policy_01
```

To delete a custom CoPP policy:

```
switch(config)# no copp-policy copp_policy_01
```

An *active* custom CoPP policy cannot be deleted; it must first be removed from service using the above command.

DHCP snooping

Note: DHCP snooping is supported on the 6200, 6300, 6400, and 8400 platforms.

DHCP snooping protects the network from common DHCP attacks, including address spoofing resulting from a rogue DHCP server operating on the network, or exhaustion of addresses on a DHCP server caused by mass address requests by an attacker on the network. The feature works by designating trusted DHCP servers and ports on which DHCP requests and responses will be accepted.

Dynamic ARP Inspection

Note: Dynamic ARP Inspection is supported on the 6200, 6300, 6400, and 8400 platforms.

Address Resolution Protocol (ARP) allows hosts to communicate over the network by creating an IP to MAC address mapping used in the transmission of packets. Attackers can use ARP to generate bogus mappings, thereby allowing them to spoof other clients' MAC addresses and intercept traffic destined to them. Additionally, an attacker could generate an unlimited number of artificial ARP entries, filling up the caches of other clients on the network and causing a denial of service (DoS).

Border Gateway Protocol (BGP) routing

Securing BGP sessions

Note: BGP is supported on the 6300, 6400, 8320, 8325, 8360, and 8400 platforms.

The IETF Best Current Practices for BGP Security (BCP194) contains many useful suggestions. This guide will focus on the following three items:

- Utilizing the control-place ACL functionality to limit BGP communication to configured BGP peers
- Securing BGP sessions between peers with authentication.
- Use TTL Security Mechanisms to prevent spoofing attacks from third parties.

Control Plane ACL for BGP peering sessions

Devices running BGP listen for connections on TCP port 179. When establishing a BGP peering session, one device will actively establish a relationship with the other peer by sending the first TCP SYN packet. This device is referred to as the outgoing side of the connection. The other peer, hearing the TCP SYN, responds with a SYN/ACK, is referred to as the incoming connection. As either peer is capable of assuming either role, ACL entries need to be configured for BGP in both directions.

Building on the same Control Plane ACL example as before, the below entries will permit traffic from 10.20.0.10 so that it can establish a BGP peering session with the device. Because either side could play the outgoing or incoming role in the connection, the ACL requires two entries per peer:

```
switch(config)# access-list ip CONTROLPLANE
switch(config-acl-ip)# 800 comment LOCKDOWN BGP SESSIONS
switch(config-acl-ip)# 805 permit tcp 10.20.0.10 gt 1023 any eq 179
switch(config-acl-ip)# 810 permit tcp 10.20.0.10 eq 179 any gt 1023
```

After allowing traffic from all configured peers, block all other devices from establishing a BGP peering session by denying all other traffic to or from TCP port 179.

```
switch(config-acl-ip)# 890 deny tcp any gt 1023 any eq 179
switch(config-acl-ip)# 895 deny tcp any eq 179 any gt 1023
```

Authenticate BGP Peers Using MD5

The TCP sessions between the two peers can be secured by adding MD5 protection to the TCP session header. The MD5 digest acts like a password between peers. This configuration is done within the BGP configuration context, and both peers will need to configure the same password.

```
switch(config-bgp)# neighbor 10.20.0.10 password plaintext meatballs4me!
```

BGP TTL security

Assuming most routing neighbors are typically directly connected, a simple method to block remote spoofing from remote devices is to check the TTL of the peer's packets and drop packets whose TTL is less than the expected amount.

Here is an example using the BGP peer specified above. Assuming the maximum TTL value is 255, the peer's packets are compared against the hop-count, entered below as a value of 1.

```
switch(config-bgp)# neighbor 10.20.0.10 ttl-security-hops 1
```

With a maximum TTL value of 255 and a configured hop count value of 1, the packets with a TTL below 254 will be dropped.

Other BGP configuration

For additional BGP related items, such as configuring inbound and outbound route filtering or limiting the maximum number of routes to learn per BGP neighbor, please refer to the BGP and Route Policies and Route Maps portion of the **AOS-CX IP Routing Guide.** The IETF published Best Current Practice on BGP Operations and Security is also an excellent resource and highly recommended.

Open Shortest Path First (OSPF) routing

OSPF passive interfaces

Note: OSPF is supported on the 6200, 6300, 6400, 8320, 8325, 8360, and 8400 platforms.

Unlike BGP, most routing protocols tend to discover neighbors via the sending and receiving of Hello packets. Because the building of these neighbor relationships occurs dynamically, the administrator should take steps to control where neighbor relationships can form and that potential neighbors are known and trusted devices.

To limit where OSPF can learn neighbors, AOS-CX supports the concept of passive OSPF interfaces. A passive OSPF interface has its IP subnets announced, but it does not establish neighbor relationships with other OSPF devices on the interface.

The recommended method is to make all OSPF enabled interfaces passive. Setting the default to passive is done in the OSPF router instance context:

```
switch(config-ospf-10)# passive-interface default
```

The passive interface is then removed from each specific interface where OSPF neighbor relationships are allowed. Since this is an interface-level configuration change, it happens from the interface context:

```
switch(config-if)# no ip ospf passive
```

OSPF neighbor authentication

Note: Any SHA or MD5 key string entered in plaintext will automatically be hashed into ciphertext before being stored in the switch configuration.

Technically speaking, all OSPF exchanges are authenticated. However, the default authentication used by network vendors is "null," meaning empty or zero. OSPF also supports using a simple plaintext password and cryptographic authentication. AOS-CX supports several OSPF authentication methods, including SHA cryptographic hashes up to 512 bits, to authenticate messages between OSPF neighbors.

When configuring authentication between OSPF neighbors, the authentication method and key must be the same on the connected interfaces on both devices.

To configure SHA-512 authentication, change the default authentication method from **null** to **hmac-sha-512** from the interface context:

```
switch(config-if)# ip ospf authentication hmac-sha-512
```

Then configure a SHA key to be used for the connection; the key can be entered as plaintext or as a hashed ciphertext string:

```
switch(config-if)# ip ospf sha-key 1 plaintext ospfshakeystring
```

Alternatively, the AOS-CX keychain feature may be used to specify a system-level cryptographic authentication key that can be used by multiple OSPF interfaces:

```
switch(config)# keychain ospf-keychain
switch(config-keychain)# key 1
```

```
switch(config-keychain-key)# cryptographic-algorithm hmac-sha-512
switch(config-keychain-key)# key-string plaintext ospfshakeystring
switch(config-keychain-key)# interface 1/1/49
switch(config-if)# ip ospf authentication keychain
switch(config-if)# ip ospf keychain ospf-keychain
```

OSPFv3 area authentication and encryption with IPsec

Note: OSPFv3 area authentication or encryption settings will be overridden by interface-level authentication or encryption, where configured.

OSPFv3 neighbors may use interface-level authentication, as described in the previous section; however, an alternative method may be used to provide encryption and/or authentication for an entire OSPFv3 area using the IPsec protocol, which automatically applies the configured methods to all member interfaces. There are two IPsec encapsulation types supported on AOS-CX to secure OSPFv3 areas:

- IPv6 authentication header (AH), which adds an IPv6 authentication header to OSPFv3 packets
- Encrypted Security Payload (ESP), which provides both authentication and encryption for OSPFv3 packets

IPsec authentication and encryption are configured from the OSPFv3 router process context:

```
switch(config)# router ospfv3 1
switch(config-ospfv3-1)#
```

Both authentication and encryption require a specified Security Policy Index (SPI), which is an integer value between 256 and 4,294,967,295; this value is used on each OSPFv3 router in the secured area to match a configured IPsec authentication and/or encryption policy. Each OSPFv3 IPsec policy on a switch must use a different SPI value, and the SPI value (as well as authentication and/or encryption keys) must match across all OSPFv3 neighbor interfaces using that policy within the secured area.

To configure AH authentication for OSPFv3 area 1, specify the SPI, authentication method (**md5** or **sha1**), key type (**plaintext**, **hex-string**, or **ciphertext**) and the key string itself. If a key type and string are not specified, the user will be prompted to enter a plaintext key interactively:

```
switch(config-ospfv3-1)# area 1 authentication ipsec spi 1024 sha1
Enter the IPsec authentication key: *******
Re-Enter the IPsec authentication key: ********
```

To configure ESP encryption for area 1, specify the SPI, authentication method, authentication key type and string, encryption type (**3des**, **aes**, **des**, or **null**), key type, and encryption key string. If the encryption type and key string are not specified, the user will be prompted to enter a plaintext key interactively. If the authentication key type and string are not specified, the user will be prompted to enter both a plaintext authentication key as well as the desired encryption type and plaintext key.

```
switch(config-ospfv3-1)# area 1 encryption ipsec spi 1024 sha1
Enter the IPsec authentication key: *******
Re-Enter the IPsec authentication key: *******

Enter the IPsec encryption type (3des/aes/des/null)? aes

Enter the IPsec encryption key: ****************
Re-Enter the IPsec encryption key: ****************
```

Depending on the selected encryption type, a plaintext or hexadecimal encryption key must be set to a specific length:

- 3DES:
  - Hexadecimal: 48 digits
  - Plaintext: 24 characters
- DES:

  - Hexadecimal: 16 digits
  - Plaintext: 8 characters
- AES
  - Hexadecimal: 32, 48, or 64 digits
  - Plaintext: 16, 24, or 32 characters

For AES encryption, the specified key lengths correspond to AES128, AES192, or AES256, respectively; the type used will be automatically determined by the length of the entered encryption key.

Other OSPF configuration

For other OSPF related items that might be of interest, such as leveraging route-maps when redistributing routes from other protocols, please review the **AOS-CX IP Routing Guide**.

Applicable platforms

The content of the Aruba CX Hardening Guide is applicable to the following platforms:

- Aruba CX 6100 Switch Series
- Aruba CX 6200 Switch Series
- Aruba CX 6300 Switch Series
- Aruba CX 6400 Switch Series
- Aruba CX 8320 Switch Series
- Aruba CX 8325 Switch Series
- Aruba CX 8360 Switch Series
- Aruba CX 8400 Switch Series