Technical white paper

HP Networking guide to hardening Comware-based devices

Table of contents

Introduction

This document contains information to help you secure your HP Comware OS-based devices, which will help increase the overall security of your network. This document, which is structured around the three planes into which network device functions can be categorized, provides an overview of each feature and references related documentation.

The three functional planes of a network—the management plane, control plane, and data plane—each provide different functionality that must be protected.

Management plane

The management plane consists of functions that achieve the management goals of the network. It includes interactive management sessions using secure shell (SSH), as well as statistics gathering with SNMP, NetStream, or sFlow. When you consider the security of a network device, it is critical that the management plane be protected. If a security incident is able to undermine the functions of the management plane, it can be impossible for you to recover or stabilize your network.

The sections of this document detail the security features and configurations available in Comware-based HP software that help fortify the management plane.

General management plane hardening

The management plane is used to access, configure, and manage a device, as well as monitor its operations and the network on which it is deployed. The management plane is the plane that receives and sends traffic for operations of these functions. You must secure both the management plane and control plane of a device, as operations of the control plane directly affect operations of the management plane. The following protocols are used by the management plane:

- SNMP
- Telnet
- Secure shell (SSH)
- FTP
- TFTP
- Secure FTP
- HWTACACS
- RADIUS
- NetStream
- sFlow
- NTP
- Syslog

Steps must be taken to help ensure the survival of the management and control planes during security incidents. If one of these planes is successfully exploited, all network planes can be compromised.

Password control

Password control refers to a set of functions provided by the local authentication server to control user login passwords, super passwords, and user login status based on predefined policies. If passwords are cracked by attackers, the whole network is compromised.

Generally, an administrator sets a password for each network user. A password is displayed in either plain text or cipher text. A plain-text password is visible to all users logged in through the console port. In addition, a user can log in to a device and obtain its configuration file, from which the user can view user names and plain-text passwords. Cipher-text passwords are therefore recommended, especially when password control is no enabled.

Cipher text prevents logged-in users from viewing passwords, but the passwords can still be cracked by some software. If a user obtains the configuration file, the user can then easily use crack software to obtain the passwords.

After password control is configured, a password is displayed as ***, and is saved in a special format in the configuration file.

Users will often choose their user names or simple digits such as 123456 as their passwords. These passwords can easily be cracked. Increasing password complexity can make it more difficult to crack passwords.

With password control, the administrator can configure the minimum password length, password composition check, password complexity check, password update interval, password aging, early notice on pending password expiration, login with an expired password, password history, login attempt limit, password display, authentication timeout management, maximum account idle time, and logging. (The system logs all successful password change events and user blacklisting events due to login failures.)

The following gives a typical configuration example of password control:

Enable password control globally.

```
[Sysname] password-control enable
```

Prohibit a user from logging in forever after two consecutive login failures.

```
[Sysname] password-control login-attempt 2 exceed lock
```

Set an age time of 30 days for all passwords.

```
[Sysname] password-control aging 30
```

Set the minimum password update interval to 36 hours.

```
[Sysname] password-control password update interval 36
```

Specify that a user can log in five times within 60 days after the password expires.

```
[Sysname] password-control expired-user-login delay 60 times 5
```

Set the maximum account idle time to 30 days.

```
[Sysname] password-control login idle-time 30
```

Refuse any password that contains the user name or the reverse of the user name.

```
[Sysname] password-control complexity user-name check
```

Specify that no character of the password can be repeated three or more times consecutively.

```
[Sysname] password-control complexity same-character check
```

Set the minimum number of composition types for super passwords to 3 and the minimum number of characters of each composition type to 5.

```
[Sysname] password-control super composition type-number 3 type-length 5
```

Configure a super password.

```
[Sysname] super password level 3 simple 12345ABGFTweuix
```

Create a local user named test.

```
[Sysname] local-user test
```

Set the service type of the user to Telnet.

```
[Sysname-luser-test] service-type telnet
```

Set the minimum password length to 12 for the local user.

```
[Sysname-luser-test] password-control length 12
```

Set the minimum number of password composition types to 2 and the minimum number of characters of each password composition type to 5 for the local user.

```
[Sysname-luser-test] password-control composition type-number 2 type-length 5
```

Set the password age time to 20 days for the local user.

```
[Sysname-luser-test] password-control aging 20
```

Configure the password of the local user in interactive mode.

```
[Sysname-luser-test] password
Password:***********
Confirm :***********
Updating user(s) information, please wait........
[Sysname-luser-test] quit
```

Disable unused services

As a security best practice, any unnecessary service must be disabled. These unneeded services, especially those that use User Datagram Protocol (UDP), are infrequently used for legitimate purposes, but can be used to launch DoS and other attacks that can otherwise be prevented by packet filtering.

Following is a list of additional services that must be disabled if not in use:

- Issue the **undo dhcp enable** command in system view to disable DHCP.
- Issue the **undo dns resolve** command in system view to disable DNS.
- Issue the **undo x25 switching** command in system view to disable X.25 switching function.
- Issue the **undo ip http enable** command in system view to disable HTTP server.
- Issue the **undo ip https enable** command in system view to disable HTTPS server.

Neighbor Discovery Protocol (NDP) is used to discover other NDP-enabled devices for neighbor adjacency and network topology. NDP can be used by HGMP to manage a cluster. NDP must be disabled on all interfaces that are connected to untrusted networks. This is accomplished by issuing the **undo ndp enable** command in interface view. Alternatively, NDP can be disabled globally with the **undo ndp enable** command in system view or on interfaces by specifying an interface list in system view. Note that NDP can be used by a malicious user for reconnaissance and network mapping.

Link Layer Discovery Protocol (LLDP) is an IEEE protocol that is defined in 802.1AB. LLDP is similar to NDP; however, this protocol allows interoperability between other devices that do not support NDP. LLDP must be treated in the same manner as NDP and disabled on all interfaces that connect to untrusted networks. To accomplish this, issue the **undo lldp enable** command in interface view. To disable LLDP globally, issue the **undo lldp enable** command in system view. LLDP can also be used by a malicious user for reconnaissance and network mapping.

EXEC timeout

To set the interval so that the command interpreter waits for user input before it terminates a session, issue the **idle-timeout** command in interface view. The **idle-timeout** command must be used to log out sessions on a virtual type terminal (VTY) or true type terminal (TTY) interface that is left idle. By default, sessions are disconnected after 10 minutes of inactivity.

```
#
user-interface con 0
 idle-timeout 2 0
user-interface aux 0
 idle-timeout 2 0
user-interface vty 0 4
idle-timeout 2 0
#
```

Using management interfaces

A device's management plane is accessed in band or out of band on a physical or logical management interface. Ideally, both in-band and out-of-band management access exist for each network device so that the management plane can be accessed during network outages.

One of the most common interfaces that are used for in-band device access is the logical loopback interface. Loopback interfaces are always up, whereas physical interfaces can change state and potentially be inaccessible. It is recommended that you add a loopback interface to each device as a management interface and that it be used

exclusively for the management plane. This allows the administrator to apply policies throughout the network for the management plane. Once the loopback interface is configured on a device, it can be used by management plane protocols such as SSH, SNMP, and syslog to send and receive traffic.

Memory Threshold Notification

The Memory Threshold Notification feature allows you to mitigate low-memory conditions on a device. This feature uses two methods to accomplish this: Memory Threshold Notification and Memory Reservation.

Memory Threshold Notification generates a log message to indicate that free memory on a device has fallen below the configured threshold.

Memory Reservation is used so that sufficient memory is available for critical notifications.

Comware does not support the manual modification of thresholds for Memory Threshold Notification and Memory Reservation, which are determined by the system during startup.

CPU Threshold Notification

The CPU Threshold Notification feature allows you to detect and be notified when the CPU load on a device crosses a configured threshold. When the threshold is crossed, the device generates and sends an SNMP trap message.

Comware does not support the manual modification of threshold for CPU Threshold Notification, which is determined by the system during startup.

Limiting access to the network with infrastructure ACLs

Devised to prevent unauthorized direct communication to network devices, infrastructure access control lists (ACLs) are one of the most critical security controls that can be implemented in networks. Infrastructure ACLs leverage the idea that nearly all network traffic traverses the network and is not destined to the network itself.

An infrastructure ACL is constructed and applied to specify connections from hosts or networks that need to be allowed access to network devices. Common examples of these types of connections are eBGP, SSH, and SNMP. After the required connections have been permitted, all other traffic to the infrastructure is explicitly denied. All transit traffic that crosses the network and is not destined to infrastructure devices is then explicitly permitted.

The protections provided by infrastructure ACLs are relevant to both the management and control planes. The implementation of infrastructure ACLs can be made easier through the use of distinct addressing for network infrastructure devices.

The following example ACL configuration illustrates the structure that must be used as a starting point when you begin the ACL implementation process:

#

```
acl number 3000 name ACL-INFRASTRUCTURE-IN
```

#

Permit required connections for routing protocols and network management

#

```
rule permit tcp source <trusted-ebgp-peer> 0 destination <local-ebgp-address> 0
destination-port eq 179

rule permit tcp source <trusted-ebgp-peer> 0 source-port eq 179 destination <local-ebgp¬
address> 0

rule permit tcp source <trusted-management-stations> 0 destination-port eq 22

rule permit udp source <trusted-netmgmt-servers> 0 destination-port eq 161
```

#

Deny all other IP traffic to any network device

#

```
rule deny ip destination <infrastructure-address-space> <wildcard>
```

#

Permit transit traffic

#

```
rule permit ip
```

#

Once created, the ACL must be applied to all interfaces that face non-infrastructure devices. This includes interfaces that connect to other organizations, remote access segments, user segments, and data center segments.

ICMP packet filtering

Internet Control Message Protocol (ICMP) is designed as an IP control protocol. As a result, the messages it conveys can have far-reaching ramifications to TCP and IP protocols in general. While the network troubleshooting tools **ping** and **traceroute** use ICMP, external ICMP connectivity is rarely needed for the proper operation of a network.

HP Comware OS software provides functionality to specifically filter ICMP messages by name or type and code. The following example ACL, which must be used with the access control entries (ACEs) from previous examples, allows pings from trusted management stations and NMS servers, and blocks all other ICMP packets:

#

```
acl number 3000 name ACL-INFRASTRUCTURE-IN
```

#

Permit ICMP Echo (ping) from trusted management stations and servers

```
rule permit icmp source <trusted-management-stations> 0 icmp-type echo
rule permit icmp source <trusted-netmgmt-servers> 0 icmp-type echo
```

#

Deny all other IP traffic to any network device

```
rule deny ip destination <infrastructure-address-space> <wildcard>
```

#

Permit transit traffic

```
rule permit ip
```

#

Filtering IP fragments

Filtering fragmented IP packets can pose a challenge to security devices. This is because the Layer 4 information that is used to filter TCP and UDP packets is only present in the initial fragment. HP Comware software uses a specific method to check non-initial fragments against configured access lists. HP Comware software evaluates these non-initial fragments against the ACL and ignores any Layer 4 filtering information. This causes non-initial fragments to be evaluated solely on the Layer 3 portion of any configured ACE.

In the example configuration that follows, if a TCP packet destined to 192.168.1.1 on port 22 is fragmented in transit, the initial fragment is dropped as expected by the second ACE based on the Layer 4 information within the packet. However, all remaining (non-initial) fragments are allowed by the first ACE based completely on the Layer 3 information in the packet and ACE. This scenario is shown in this configuration:

#

```
acl number 3000 name ACL-FRAGMENT-EXAMPLE
rule permit tcp destination 192.168.1.1 0 destination-port eq 80
rule deny tcp destination 192.168.1.1 0 destination-port eq 22
```

#

Due to the nonintuitive nature of fragment handling, IP fragments are often inadvertently permitted by infrastructure ACLs. Fragmentation is also often used in attempts to evade detection by intrusion detection systems. It is for these reasons that IP fragments are often used in attacks, and why they must be explicitly filtered at the top of any configured

infrastructure ACLs. The example ACL that follows includes comprehensive filtering of IP fragments. The functionality from this example must be used in conjunction with the functionality of the previous examples.

```
#
acl number 3001 name ACL-INFRASTRUCTURE-IN
#
# Deny IP fragments using protocol-specific ACEs to aid in classification of attack traffic
#
rule deny tcp fragment
rule deny udp fragment
rule deny icmp fragment
rule deny ip fragment
#
# Deny all other IP traffic to any network device #
rule deny ip destination <infrastructure-address-space> <wildcard>
#
# Permit transit traffic #
rule permit ip
#
```

For more information regarding ACL handling of fragmented IP packets, see “Filtering IP fragments.”

Securing interactive management sessions

Management sessions to devices allow you the ability to view and collect information about a device and its operations. If this information is disclosed to a malicious user, the device can become the target of an attack, become compromised, and be used to perform additional attacks. Anyone with privileged access to a device has the capability for full administrative control of that device. Securing management sessions is imperative to prevent information disclosure and unauthorized access.

Console and AUX ports

In HP Comware devices, and console and auxiliary (AUX) ports are asynchronous lines that can be used for local and remote access to a device. Console ports on HP Comware devices have special privileges. By default, an administrator can access a device through its console port without password authentication.

You can configure authentication, authorization, and accounting (AAA) to authenticate users accessing the console port. Following is an example configuration:

```
#
user-interface con 0
  authentication-mode scheme
  idle-timeout 1 0
user privilege level 3
#
```

To adopt username/password authentication, configure the following:

```
#
user-interface con 0
  authentication-mode password
  set authentication password cipher password
```

```
idle-timeout 1 0
user privilege level 3
#
```

To access the AUX port remotely, the user must first pass local password authentication by default. You can configure AAA to authenticate users accessing the AUX port as follows:

```
#
user-interface aux 0
  authentication-mode scheme
  idle-timeout 1 0
user privilege level 3
#
```

You can disable authentication so that users can access the device through the AUX port directly as follows:

```
#
user-interface aux 0
  authentication-mode none
  user privilege level 3
  idle-timeout 1 0
#
```

Control VTY and TTY lines

Interactive management sessions in HP Comware software use a TTY or virtual TTY (VTY). A TTY is used by a terminal for local access to the device or to a modem for dialup access to a device. Note that TTYs can be used for connections to the console ports of other devices. This function allows for reverse Telnet to the device. The TTY lines for these reverse connections must also be controlled.

A VTY line is used for all other remote network connections supported by the device. To ensure that a device can be accessed via a local or remote management session, proper controls must be enforced on both VTY lines. HP Comware devices have a limited number of VTY lines. When all VTY lines are in use, new management sessions cannot be established, creating a DoS condition for access to the device.

Authentication can be enforced through the use of AAA, which is the recommended method for authenticated access to a device. The following gives an example configuration:

```
#
user-interface tty 33
  authentication-mode scheme
  user privilege level 3
  idle-timeout 1 0
user-interface vty 0 4
  authentication-mode scheme
  user privilege level 3
  idle-timeout 1 0
#
```

Note: Set a short time value with the **idle-timeout** command to ensure that users who no longer use the TTYs or VTYs are logged out in time. The default time value is 10 minutes.

Warning banners
In some legal jurisdictions, it can be impossible to prosecute and illegal to monitor malicious users unless they have been notified that they are not permitted to use the system. One method to provide this notification is to place this information into a banner message that is configured with the HP Comware software header legal command.

Legal notification requirements are complex, vary by jurisdiction and situation, and should be discussed with legal counsel. Even within jurisdictions, legal opinions can differ. In cooperation with counsel, a banner can provide some or all of the necessary information.

The notice should indicate that the system is to be logged into or used only by specifically authorized personnel; it can also contain information about who can authorize use:

- Notice that any unauthorized use of the system is unlawful and can be subject to civil and criminal penalties.
- Notice that any use of the system can be logged or monitored without further notice and that the resulting logs can be used as evidence in court.
- Specific notices required by local laws.
- From a security point of view, rather than a legal one, a login banner should not contain any specific information about the router name, model, software, or ownership. This information can be abused by malicious users.

Note: You can use the **undo copyright-info enable** command to disable displaying copyright information upon login.

Using authentication, authorization, and accounting

The authentication, authorization, and accounting (AAA) framework is critical to securing interactive access to network devices. The AAA framework provides a highly configurable environment that can be tailored depending on the needs of the network.

Authentication, authorization, and accounting with RADIUS
Remote Authentication Dial-In User Service (RADIUS) is a distributed information interaction protocol that uses a client/server model. RADIUS can protect networks against unauthorized access and is often used in network environments where both high security and remote user access are required.

RADIUS uses UDP as the transport protocol. It uses UDP port 1812 for authentication and port 1813 for accounting.

RADIUS was originally designed for dial-in user access. With the diversification of access methods, RADIUS has been extended to support more access methods, for example, Ethernet access and ADSL access. It provides access authentication and authorization services.

Information exchanged between a RADIUS client and the RADIUS server is authenticated with a shared key, which is never transmitted over the network to enhance security. In addition, to prevent user passwords from being intercepted in non-secure networks, RADIUS encrypts passwords before transmitting them.

The following gives an example RADIUS configuration:

```
#
radius scheme radius
  primary authentication 192.168.0.1
  primary accounting 192.168.0.1
  secondary accounting 192.168.0.2
  key authentication HP
  key accounting HP
  user-name-format without-domain
#
```

Authentication, authorization, and accounting with HWTACACS

HWTACACS and RADIUS both provide authentication, authorization, and accounting services. They have many common features in implementing AAA, such as using the client/server model, using shared keys for user information security, and having good flexibility and extensibility. They also have differences, which are listed below.

HWTACACS	RADIUS
Uses TCP, providing more reliable networking transmission.	Uses UDP, providing higher transport efficiency.
Encrypts the entire packet except for the HWTACACS header.	Encrypts only the user password field in an authentication packet.
Protocol packets are complicated, and authorization is independent of authentication. Authentication and authorization can be deployed on different HWTACACS servers.	Protocol packets are simple, and authorization is combined with authentication.
Supports authorization of configuration commands. Which commands a user can use depends on both the user level and AAA authorization. A user can use only commands that are not only of, or lower than, their user level but also authorized by the HWTACACS server.	Does not support authorization of configuration commands. Which commands a user can use depends on the user's level. A user can use all the commands of, or lower than, their user level.

The following gives an example HWTACACS AAA configuration:

```
#
hwtacacs scheme tacacs
  primary authentication 192.168.0.1
  secondary authentication 192.168.0.2
  primary authorization 192.168.0.1
  secondary authorization 192.168.0.2
  primary accounting 192.168.0.1
  secondary accounting 192.168.0.2
  key authentication HP
  key authorization HP
  key accounting HP
  user-name-format without-domain
#
```

Authentication and authorization with LDAP

Based on TCP/IP, Lightweight Directory Access Protocol (LDAP) is used to provide standard multi-platform directory service. It is developed on the basis of the X.500 protocol, and improves the read/write interactive access, and browse and search functions of X.500. It is suitable for storing data that is not often changed.

LDAP is typically used to store user information in a system. For example, the Active Directory Server is used in Microsoft® Windows® operating systems to store user information and user group information for authentication and authorization at login.

The following gives an example LDAP configuration:

```
#
ldap scheme ldap
  authentication-server 192.168.0.244
  authorization-server 192.168.0.244
  login-dn cn=administrator,cn=users,dc=server
  login-password simple sys508
  user-parameters search-base-dn dc=server
#
```

Authentication fallback
If all authentication servers are unavailable, local authentication can be used.

Local authentication can use the password control function to secure user passwords.

Redundant AAA servers
You can specify multiple RADIUS or HWTACACS authentication/authorization servers to achieve redundancy.

When the primary authentication/authorization server is unreachable, the access device contacts the secondary server to perform authentication/authorization. You can specify one primary server, and up to 16 secondary servers. You can also specify a server as the primary authentication/authorization server in a scheme, and at the same time specify it as the secondary authentication/authorization server in another scheme.

Fortifying Simple Network Management Protocol

This section highlights several methods that can be used to secure the deployment of SNMP within HP Comware devices. It is critical that SNMP be properly secured to protect the confidentiality, integrity, and availability of both the network data and the network devices through which this data transits. SNMP provides you with a wealth of information on the health of network devices. This information should be protected from malicious users who want to leverage this data to perform attacks against the network.

SNMP community strings

Community strings are passwords that are applied to a Comware device to restrict access (both read-only and read-write access) to the SNMP data on the device. These community strings, as with all passwords, should be carefully chosen to ensure they are not trivial. Community strings should be changed at regular intervals and in accordance with network security policies. For example, the strings should be changed when a network administrator changes roles or leaves the company.

These configuration lines configure a read-only community string of *READONLY* and a read/write community string of *READWRITE*:

```
#
snmp-agent community read READONLY
snmp-agent community write READWRITE
#
```

Note that the preceding community string examples have been chosen to clearly explain the use of these strings. For production environments, community strings should be chosen with caution and should consist of a series of alphabetical, numerical, and nonalphanumeric symbols.

For more information about this feature, see “SNMP” in the *Network Management and Monitoring Command Reference Guide*.

SNMP community strings with ACLs

In addition to the community string, an ACL should be applied that further restricts SNMP access to a select group of source IP addresses. The following configuration restricts SNMP read-only access to end host devices that reside in the 192.168.100.0/24 address space and restricts SNMP read/write access to only the end host device at 192.168.100.1.

Note that the devices that are permitted by these ACLs require the proper community string to access the requested SNMP information:

```
#
acl number 2001
  rule 1 permit source 192.168.100.0 0.0.0.255
acl number 2002
  rule 1 permit source 192.168.100.1 0
#
snmp-agent community read READONLY acl 2001
```

```
snmp-agent community write READWRITE acl 2002
```

#

For more information, see the **snmp-server community** command in “SNMP” in the *Network Management and Monitoring Command Reference Guide*.

SNMP Views

SNMP Views are a security feature that can permit or deny access to certain SNMP MIBs. Once a view is created and applied to a community string with the **snmp-agent community** command, if you access MIB data, you are restricted to the permissions that are defined by the view. When appropriate, you are advised to use views to limit SNMP users to the data that they require.

The configuration example that follows restricts SNMP access with the community string *LIMITED* to the MIB data that is located in the *system* group:

#

```
snmp-agent mib-view included VIEW-SYSTEM-ONLY system
```

#

```
snmp-agent community read LIMITED mib-view VIEW-SYSTEM-ONLY
```

#

For more information, see “SNMP” in the *Network Management and Monitoring Command Reference Guide*.

SNMP Version 3

SNMP Version 3 (SNMPv3) is defined by RFC3410, RFC3411, RFC3412, RFC3413, RFC3414, and RFC3415, and is an interoperable standards-based protocol for network management. SNMPv3 provides secure access to devices by authenticating and optionally encrypting packets over the network. Where supported, SNMPv3 can be used to add another layer of security when deploying SNMP. SNMPv3 consists of three primary configuration options:

- **no authentication**
  This mode does not require any authentication or any encryption of SNMP packets.
- **authentication**
  This mode requires authentication of the SNMP packet without encryption.
- **privacy**
  This mode requires both authentication and encryption (privacy) of each SNMP packet.

An authoritative engine ID must exist before the SNMPv3 security mechanisms authentication or authentication and encryption can be used for handling SNMP packets. By default, the engine ID is generated locally. The engine ID can be displayed with the **display snmp-agent local-engineid** command as shown in this example:

#

```
[HP]display snmp-agent local-engineid
  SNMP local EngineID: 800063A203000FE2000002
```

#

Note that if the engine ID is changed, all SNMP user accounts must be reconfigured. The next step is to configure an SNMPv3 group. This command configures an HP Comware device for SNMPv3 with an SNMP server group *AUTHGROUP* and enables only authentication for this group by using the **authentication** keyword:

#

```
snmp-agent group v3 AUTHGROUP authentication
```

#

This command configures an HP Comware device for SNMPv3 with an SNMP server group *PRIVGROUP* and enables both authentication and encryption for this group by using the **privacy** keyword:

#

```
snmp-agent group v3 PRIVGROUP privacy
```

#

This command configures an SNMPv3 user snmpv3user with an MD5 authentication password of authpassword and a 3DES encryption password of privpassword:

#

```
snmp-agent usm-user v3 snmpv3user PRIVGROUP authentication-mode md5 authpas
sword privacy-mode 3des privpassword
```

#

Additionally, it is recommended that SNMPv1/v2 be disabled whenever SNMPv3 is configured for an additional level of security. For more information, see “SNMP” in the *Network Management and Monitoring Command Reference Guide*.

Logging best practices

Event logging provides you with visibility into the operation of an HP Comware device and the network into which it is deployed. HP Comware software provides several flexible logging options that can help achieve an organization’s network management and visibility goals.

These sections provide some basic logging best practices that can help an administrator leverage logging successfully while minimizing the impact of logging on an HP Comware device.

Send logs to a central location

You are advised to send logging information to a remote syslog server. By doing so, it becomes possible to correlate and audit network and security events across network devices more effectively. Note that syslog messages are transmitted unreliably by UDP and in cleartext. For this reason, any protections that a network affords to management traffic (for example, encryption or out-of-band access) should be extended to include syslog traffic.

The following configuration example configures an HP Comware device to send logging information to a remote syslog server:

#

```
info-center loghost <ip-address>
```

#

For more information on log correlation, see “Information Center” in the *Network Management and Monitoring Configuration Guide*.

Logging level

Each log message that is generated by an HP Comware device is assigned one of eight severity levels that range from level 0 (emergencies) through level 7 (debug). Unless specifically required, you are advised to avoid logging at level 7. Logging at level 7 produces an elevated CPU load on the device that can lead to device and network instability.

The system-view configuration command **info-center source default channel loghost log level** is used to specify which logging messages are sent to remote syslog servers. The level specified indicates the lowest severity message that is sent. For buffered logging, the **info-center source default channel logbuffer log level** command is used.

This configuration example limits log messages that are sent to remote syslog servers and the local log buffer to severities 6 (informational) through 0 (emergencies):

#

```
info-center source default channel logbuffer log level informational
info-center source default channel loghost log level informational
```

#

For more information, see “Information Center” in the *Network Management and Monitoring Command Reference Guide*.

Do not log to console or monitor sessions

With HP Comware software, it is possible to send log messages to monitor sessions and to the console. Monitor sessions are interactive management sessions in which the EXEC command **terminal monitor** has been issued. However, sending such messages can elevate the CPU load of a Comware device and therefore is not recommended.

Instead, you are advised to send logging information to the local log buffer, which can be viewed by using the **display logbuffer** command.

Use the system-view configuration commands **info-center source default channel console log state off** and **info-center source default channel monitor log state off** to disable logging to the console and monitor sessions. The following configuration example shows the use of these commands:

```
#
  info-center source default channel console log state off
  info-center source default channel monitor log state off
#
```

server:

```
#
info-center loghost <ip-address>
#
```

For more information on log correlation, see “Information Center” in the *Network Management and Monitoring Configuration Guide*.

Use buffered logging

HP Comware software supports the use of a local log buffer so that an administrator can view locally generated log messages. The use of buffered logging is highly recommended versus logging to either the console or monitor session.

There are two configuration options that are relevant when configuring buffered logging: the logging buffer size and the message severities that are stored in the buffer. The size of the logging buffer is configured with the system-view configuration command **info-center logbuffer size**. The lowest severity included in the buffer is configured using the **info-center source default channel logbuffer log level** command. An administrator is able to view the contents of the logging buffer through the **display logbuffer** EXEC command.

The following configuration example includes the configuration of a logging buffer of 1,024 items, as well as a severity of 6 (informational), indicating that messages at levels 0 (emergencies) through 6 (informational) are stored:

```
#
info-center logbuffer size 1024
info-center source default channel logbuffer log level informational
#
```

For more information, see “Information Center” in the *Network Management and Monitoring Command Reference Guide*.

Configure logging source interface

In order to provide an increased level of consistency when collecting and reviewing log messages, you are advised to statically configure a logging source interface. Accomplished by using the **info-center loghost source interface** command, statically configuring a logging source interface helps ensure that the same IP address appears in all logging messages that are sent from an individual HP Comware device. For added stability, you are advised to use a loopback interface as the logging source.

The following configuration example illustrates the use of the **info-center loghost source** command to specify that the IP address of the loopback 0 interface be used for all log messages:

```
#
info-center loghost source Loopback 0
#
```

For more information, see “Information Center” in the *Network Management and Monitoring Command Reference Guide*.

Configure logging timestamps

Configuring logging timestamps helps you correlate events across network devices. It is important to implement a correct and consistent logging timestamp configuration to ensure that you are able to correlate logging data. Logging timestamps should be configured to include the date and time with millisecond precision and to include the time zone in use on the device.

The following example includes the configuration of logging timestamps with the time from system boot:

```
#
info-center timestamp log boot
#
```

Use the **info-center timestamp loghost** command to configure the format of logging timestamps sent to the log host.

For more information, see "Information Center" in the *Network Management and Monitoring Command Reference Guide*.

HP Comware software configuration management

HP Comware software includes several features that can enable a form of configuration management on an HP Comware device. Such features include functionality to archive configurations and to roll back the configuration to a previous version as well as create a detailed configuration change log.

Configuration Replace and Configuration Rollback

Beginning in HP Comware Software Release 5.20, the Configuration Replace and Configuration Rollback features allow HP Comware device configurations to be archived on the device. Stored manually or automatically, the configurations in this archive can be used to replace the current running configuration by using the **configuration replace file** command.

You are advised to enable this feature on all HP Comware devices in the network. Once enabled, an administrator can cause the current running configuration to be added to the archive by using the **archive configuration** command. The archived configurations can be viewed by using the **display archive configuration** command.

The following example illustrates the configuration of automatic configuration archiving. This example instructs the HP Comware device to store archived configurations as files named *archived-config-N* on the cfa0:/config file system, to maintain a maximum of 10 backups, and to archive once per day (1,440 minutes):

```
#
  archive configuration location cfa0:/config filename-prefix archived-config
  archive configuration interval 1440
  archive configuration max 10
#
```

Although the configuration archive functionality can store up to 10 backup configurations, you are advised to consider the space requirements before using the **maximum** command.

For more information, see "Configuration File Management" in the *Fundamentals Command Reference Guide*.

Backup configuration file and boot file

The following example configures backup configuration and boot files that are used when the primary ones are unavailable:

```
#
boot-loader file cfa0:/mainmsr20_backup.bin backup
startup saved-configuration startup_backup.cfg backup
#
```

For more information, see "Configuration File Management" and "Device Management" in the *Fundamentals Command Reference Guide*.

Configuration change notification

The configuration change notification feature can log the configuration changes made to an HP Comware device. You can display the change trap with the **display trapbuffer** command. Use the **snmp-agent trap enable** command to enable configuration change notification.

```
#
[HP]display trapbuffer
Trapping buffer configuration and contents:enabled
Allowed max buffer size : 1024
Actual buffer size : 1024
Channel number : 3 , channel name : trapbuffer
Dropped messages : 0
Overwritten messages : 0
Current messages : 31

#Aug 27 04:01:50:785 2010 HP DEVM/4/SYSTEM WARM START:
 Trap 1.3.6.1.4.1.25506.6.8.5: system warm start.

#Aug 27 04:01:54:374 2010 HP SHELL/4/LOGIN:
 Trap 1.3.6.1.4.1.25506.2.2.1.1.3.0.1: login from Console
#Aug 27 04:02:10:277 2010 HP CFGMAN/4/TRAP:
          1.3.6.1.4.1.25506.2.4.2.1 configure changed:
          EventIndex=1,CommandSource=1,ConfigSource=2,ConfigDestination=4
#
```

Control plane

Control plane functions consist of the protocols and processes that communicate between network devices to move data from source to destination, including routing protocols such as the Border Gateway Protocol, as well as protocols like ICMP and the Resource Reservation Protocol (RSVP).

It is important that events in the management and data planes do not adversely affect the control plane. If a data plane event such as a DoS attack impacts the control plane, the entire network can become unstable. This information about HP Comware software features and configurations can help ensure the resilience of the control plane.

General control plane hardening

Protection of a network device's control plane is critical because the control plane helps ensure that the management and data planes are maintained and operational. If the control plane were to become unstable during a security incident, it can be impossible for you to recover the stability of the network.

In many cases, disabling the reception and transmission of certain types of messages on an interface can reduce the amount of CPU load that is required to process unneeded packets.

IP ICMP redirects

An ICMP redirect message can be generated by a router when a packet is received and transmitted on the same interface. In this situation, the router forwards the packet and sends an ICMP redirect message back to the sender of the original packet. This behavior allows the sender to bypass the router and forward future packets directly to the destination (or to a router closer to the destination). In a properly functioning IP network, a router sends redirects only to hosts on its own local subnets. In other words, ICMP redirects should never go beyond a Layer 3 boundary.

There are two types of ICMP redirect messages: redirect for a host address and redirect for an entire subnet. A malicious user can exploit the ability of the router to send ICMP redirects by continually sending packets to the router, forcing the router to respond with ICMP redirect messages. This produces an adverse impact on the CPU and on the performance of the router. In order to prevent the router from sending ICMP redirects, use the **undo ip redirects** command.

For more information on ICMP redirects, see “IP Performance Optimization” in the *Layer-3 IP Services Command Reference Guide*.

ICMP unreachables

Generating ICMP unreachable messages can increase CPU load on the device. ICMP unreachable message generation can be disabled using the **undo ip unreachables** command.

ICMP TTL-expiry

Generating ICMP timeout messages can increase CPU load on the device. ICMP TTL timeout message generation can be disabled using the **undo ip ttl-expires** command.

Proxy ARP

Proxy ARP is the technique in which one device, usually a router, answers ARP requests that are intended for another device. By "faking" its identity, the router accepts responsibility for routing packets to the real destination. Proxy ARP can help machines on a subnet reach remote subnets without configuring routing or a default gateway. Proxy ARP is defined in RFC 1027.

There are several disadvantages to utilizing proxy ARP. Doing so can result in an increase in the amount of ARP traffic on the network segment, as well as resource exhaustion and man-in-the-middle attacks. Proxy ARP presents a resource exhaustion attack vector because each proxied ARP request consumes a small amount of memory. An attacker can exhaust all available memory by sending a large number of ARP requests.

Man-in-the-middle attacks enable a host on the network to spoof the MAC address of the router, resulting in unsuspecting hosts sending traffic to the attacker. Proxy ARP can be disabled using the **undo proxy-arp enable** command in interface view.

For more information on this feature, see “ARP Configuration” in the *Layer-3 IP Services Command Reference Guide*.

Network time protocol

Network Time Protocol (NTP) is not an especially dangerous service, but any unneeded service can represent an attack vector. If NTP is used, it is important to explicitly configure a trusted time source and to use proper authentication. Accurate and reliable time is required for syslog purposes, such as during forensic investigations of potential attacks, as well as for successful VPN connectivity when depending on certificates for Phase 1 authentication.

NTP time zone—When you configure NTP, the time zone needs to be configured so that timestamps can be accurately correlated. There are usually two approaches to configuring the time zone for devices in a network with a global presence. One method is to configure all network devices with the Coordinated Universal Time (UTC—previously Greenwich Mean Time [GMT]). The other approach is to configure network devices with the local time zone. More information on this feature can be found in “clock timezone” in the HP product documentation.

NTP maximum dynamic sessions—Use the **ntp-service max-dynamic-sessions** command to set the maximum number of dynamic NTP sessions that are allowed to be established locally. Please see “NTP” in the *Network Management and Monitoring Configuration Guide and Command Reference Guide*.

NTP access control—Configure the access control right to restrict the NTP peers. The access control right mechanism provides only a minimum degree of security protection for the system running NTP. A more secure method is identity authentication. For more information, see “NTP” in the *Network Management and Monitoring Configuration Guide and Command Reference Guide*.

NTP authentication—Configuring NTP authentication provides some assurance that NTP messages are exchanged between trusted NTP peers. For more information on how to configure NTP authentication, see “NTP” in the *Network Management and Monitoring Configuration Guide and Command Reference Guide*.

Limiting the CPU impact of control plane traffic

Protecting the control plane is critical. Because application performance and the end-user experience can suffer without the presence of data and management traffic, the survivability of the control plane helps ensure that the other two planes are maintainable and operational.

Understanding control plane traffic

To properly protect the control plane of HP Comware devices, it is essential to understand the types of traffic that is processed by the CPU. CPU-processed traffic normally consists of two different types of traffic. The first type of traffic is directed to the HP Comware device and must be handled directly by the HP Comware device CPU. This traffic consists of traffic to the device:

- **Traffic to the device**
  This kind of unicast traffic matches FIB entries that either have a next hop of "127.0.0.1" or an outbound interface of InLoop0 (displayed with the **display fib** command), such as traffic destined to interface IP addresses. Some multicast traffic or broadcast traffic may also need to be processed by the device.

The second type of traffic that is handled by the CPU is data plane traffic with a destination beyond the HP Comware device itself. This traffic requires special processing by the CPU. Although not an exhaustive list of CPU-impacting data plane traffic, these types of traffic are processed by the CPU and can therefore affect the operation of the control plane:

- **IP options**
  Any IP packets with options must be processed by the CPU.
- **Fragmentation**
  Any IP packet that requires fragmentation must be passed to the CPU for processing.
- **Time-to-live (TTL) expiry**
  Packets that have a TTL value less than or equal to 1 require Internet Control Message Protocol Time Exceeded (ICMP Type 11, Code 0) messages to be sent, which results in CPU processing.
- **ICMP unreachables**
  Packets that result in ICMP unreachable messages due to routing, MTU, or filtering are processed by the CPU.
- **ICMP redirects**
  Packets received and transmitted on the same interface are processed by the CPU.
- **Traffic requiring an ARP request**
  Destinations for which no ARP entry exists require processing by the CPU.
- **Non-IP traffic**
  All non-IP traffic is processed by the CPU. The **display fib** command can be used to check the prefix and next-hop information.

FTP and TFTP ACLs

An FTP server can deny the FTP requests from some FTP clients and only permit the access of clients allowed by the ACL rules. The command to configure this feature is **ftp server acl**. For more information, see "FTP" and "TFTP" in the *Fundamentals Configuration Guide*.

The **tftp-server acl** command can be used to control the device's access to a specific TFTP server using an ACL.

User interface ACLs

You can use ACLs to control access from telnet/SSH users to VTYs. The following gives an example configuration:

```
#
acl number 2001
  rule permit source 192.168.1.26 0
#
user-interface vty 0 4
 acl [ ipv6 ] acl-number { inbound | outbound }
#
```

For more information about ACL, see “ACL” in the *Security Command Reference Guide*.

HTTPS ACLs

Use the **ip https acl** command to control HTTPS access with an ACL. Only the clients permitted by the ACL can access the HTTPS server on the device.

Control plane protection

The control plane policing feature allows you to configure a quality of service (QoS) policy that manages control plane packets to protect the control plane from denial-of-service (DoS) attacks. In this way, the control plane can help maintain packet forwarding and protocol states despite an attack or heavy traffic load on the router or switch.

#

```
system-view
control-plane [ slot slot-number ]
```

#

Apply a QoS policy. For more information, see “QoS” in the *ACL and QoS Configuration Guide*.

```
qos apply policy policy-name { inbound | outbound }
```

#

Rate limiting packets on network management interfaces

This feature is used to limit the rate of incoming packets on a network management interface to prevent DoS attacks. When the rate exceeds the threshold, excessive packets are discarded.

TCP SYN Cookie and protection against Naptha attacks

To prevent TCP connection attacks, the device provides the following features:

- SYN Cookie
- Protection against Naptha attacks

In an SYN flood attack, the attacking host sends a large number of SYN messages to the server to establish TCP connections, but it never makes any response. The server establishes a large number of incomplete TCP connections and is unable to handle services normally.

The SYN Cookie feature can prevent SYN flood attacks. After receiving a TCP connection request, the server directly returns a SYN ACK message, instead of establishing an incomplete TCP connection. Only after receiving an ACK message from the client can the server establish a connection, and then enter the ESTABLISHED state.

Follow these steps to enable the SYN Cookie feature:

#

```
  tcp syn-cookie enable
```

#

A Naptha attack uses the six TCP connection states (CLOSING, ESTABLISHED, FIN_WAIT_1, FIN_WAIT_2, LAST_ACK, and SYN_RECEIVED), and while an SYN flood attack uses only the SYN_RECEIVED state.

The Naptha attacker controls a huge number of hosts to establish TCP connections with the server, keep these connections in the same state (any of the six), and request for no data so as to exhaust the memory resources of the server. As a result, the server cannot process normal services.

Follow these steps to enable the protection against Naptha attacks:

#

```
  tcp anti-naptha enable
  tcp state { closing | established | fin-wait-1 | fin-wait-2 | last-ack | syn-received }
connection-number number
```

#

For more information on these two features, see “TCP” and “ICMP Attack Protection” in the *Security Configuration Guide*.

Securing BGP

Border Gateway Protocol (BGP) is the routing foundation of the Internet. As such, any organization with more than modest connectivity requirements often finds itself utilizing BGP. BGP is often targeted by attackers because of its ubiquity and the “set-and-forget” nature of BGP configurations in smaller organizations. However, there are many BGP-specific security features that can be leveraged to increase the security of a BGP configuration.

The following section provides an overview of the most important BGP security features. Where appropriate, configuration recommendations are made.

Generalized TTL Security Mechanism

The Generalized TTL Security Mechanism (GTSM) is designed to protect a router's IP-based control plane from CPU utilization–based attacks. In particular, while cryptographic techniques can protect the router-based infrastructure from a wide variety of attacks, many attacks based on CPU overload can be prevented by GTSM. Note that the same technique protects against other scarce-resource attacks involving a router's CPU, such as attacks against processor-line card bandwidth.

GTSM for BGP is enabled using the **ttl-security** option for the **peer** command in BGP view. The following example illustrates the configuration of this feature:

```
#
bgp <asn>
peer <ip-address> as-number <remote-asn>
peer <ip-address> ttl-security hops <hop-count>
#
```

When BGP packets are received, the TTL value is checked and must be greater than 255 minus the hop-count specified.

For more information, see “Configuring GTSM for BGP in BGP” in the *Layer-3 IP Routing Configuration Guide*.

BGP peer authentication with MD5

Peer authentication using MD5 creates an MD5 digest of each packet sent as part of a BGP session. Specifically, portions of the IP and TCP headers, TCP payload, and a secret key are used to generate the digest.

The created digest is then stored in TCP option Kind 19, which was created specifically for this purpose by RFC 2385. The receiving BGP speaker uses the same algorithm and secret key to regenerate the message digest. If the received and computed digests are not identical, the packet is discarded.

Peer authentication with MD5 is configured by using the **password** option in the **peer** command in BGP view. The use of this command is illustrated as follows:

```
#
bgp <asn>
peer <ip-address> as-number <remote-asn>
peer <ip-address> password cipher <secret>
#
```

For more information, see “Enabling MD5 Authentication for TCP Connections in BGP” in the *Layer-3 IP Routing Configuration Guide*.

Configuring maximum prefixes

BGP prefixes are stored by a router in memory. The more prefixes that a router must hold results in BGP consuming more memory. In some configurations, a subset of all Internet prefixes can be stored, such as in configurations that leverage only a default route or routes for a provider’s customer networks.

In order to prevent memory exhaustion, it is important to configure the maximum number of prefixes that is accepted on a per-peer basis. It is recommended that a limit be configured for each BGP peer.

When configuring this feature using the **peer route-limit** command in BGP view, one argument is required: the maximum number of prefixes that are accepted before a peer is shut down. Optionally, a number from 1 to 100 can also be entered. This number represents the percentage of the maximum prefix value at which point a log message is sent.

```
#
bgp <asn>
peer <ip-address> as-number <remote-asn>
peer <ip-address> route-limit <shutdown-threshold> <log-percent>
#
```

For more information, see "Limiting Prefixes Received from a Peer/Peer Group in BGP" in the *Layer-3 IP Routing Configuration Guide*.

Filtering BGP prefixes with prefix lists

Prefix lists allow a network administrator to permit or deny specific prefixes that are sent or received via BGP. Prefix lists should be used where possible to help ensure that network traffic is sent over the intended paths. Prefix lists should be applied to each eBGP peer in both inbound and outbound directions.

Configured prefix lists limit the prefixes that are sent or received to those specifically permitted by a network's routing policy. If this is not feasible due to the large number of prefixes received, a prefix list should be configured to specifically block known bad prefixes. These known bad prefixes include unallocated IP address spaces and networks that are reserved for internal or testing purposes by RFC 3330. Outbound prefix lists should be configured to specifically permit only the prefixes that an organization intends to advertise.

The configuration example that follows uses prefix lists to limit the routes that are learned and advertised. Specifically, only a default route is allowed in bound by prefix list BGP-PL-INBOUND, and the prefix 192.168.2.0/24 is the only route allowed to be advertised by BGP-PL-OUTBOUND.

```
#
ip ip-prefix BGP-PL-INBOUND index 5 permit 0.0.0.0 0
ip ip-prefix BGP-PL-OUTBOUND index 5 permit 192.168.2.0 24
#
bgp <asn>
peer <ip-address> ip-prefix BGP-PL-INBOUND import
peer <ip-address> ip-prefix BGP-PL-OUTBOUND export
#
```

For more information, see "Configuring BGP Route Distribution/Reception Filtering Policies in BGP" in the *Layer-3 IP Routing Configuration Guide*.

Filtering BGP prefixes with autonomous system path access lists

BGP autonomous system (AS) path access lists allow you to filter received and advertised prefixes based on the AS path attribute of a prefix. This can be used in conjunction with prefix lists to establish a robust set of filters.

The configuration example that follows uses AS path access lists to restrict inbound prefixes to those originated by the remote AS and to restrict outbound prefixes to those originated by the local autonomous system. Prefixes that are sourced from all other autonomous systems are filtered and are not installed in the routing table.

```
#
ip as-path 1 permit ^65501$
ip as-path 2 permit ^$
#
bgp <asn>
peer <ip-address> as-number 65501
```

```
peer <ip-address> as-path-acl 1 import
peer <ip-address> as-path-acl 2 export
#
```

Securing Interior Gateway Protocols

The ability of a network to properly forward traffic and recover from topology changes or faults is dependent on an accurate view of the topology. Running an Interior Gateway Protocol (IGP) can often provide this view. By default, IGPs are dynamic and discover additional routers that communicate with the particular IGP in use. IGPs also discover routes that can be used during a network link failure.

These subsections provide an overview of the most important IGP security features. Recommendations and examples that cover Routing Information Protocol Version 2 (RIPv2), open shortest path first (OSPF), and Intermediate System to Intermediate System (IS-IS) are provided when appropriate.

Routing protocol authentication and verification with MD5

Failure to secure the exchange of routing information allows an attacker to introduce false routing information into the network. By using password authentication with routing protocols between routers, you can aid the security of the network. However, because this authentication is sent as cleartext, it can be simple for an attacker to subvert this security control.

By adding MD5 hash capabilities to the authentication process, routing updates no longer contain cleartext passwords, and the entire content of the routing update is more resistant to tampering. However, MD5 authentication is still susceptible to brute force and dictionary attacks if weak passwords are chosen. You are advised to use passwords with sufficient randomization. Because MD5 authentication is much more secure when compared to password authentication, these examples are specific to MD5 authentication.

An example of MD5 router authentication configuration for RIPv2 follows. RIPv1 does not support authentication.

```
#
interface <interface>
rip authentication-mode md5 rfc2543 <password>
#
```

For more information, see “Configuring RIPv2 Message Authentication in RIP” in the *Layer-3 IP Routing Configuration Guide*.

Following is an example configuration for OSPF router authentication using MD5:

```
#
interface <interface>
ospf authentication-mode md5 <key-id> <password>
#
ospf <process-id>
area 0
authentication-mode md5
#
```

For more information, see “Configuring OSPF Authentication in OSPF” in the *Layer-3 IP Routing Configuration Guide*.

Following is an example configuration for IS-IS router authentication using MD5:

```
#
interface <interface>
isis authentication-mode md5 <password>
#
isis <process-id>
```

```
area-authentication-mode md5 <password>
domain-authentication-mode md5 <password>
#
```

For more information, see "Enhancing IS-IS Network Security in ISIS" in the *Layer-3 IP Routing Configuration Guide*.

Silent-interface commands

Information leaks, or the introduction of false information into an IGP, can be mitigated through use of the **silent-interface** command, which assists in controlling the advertisement of routing information. You are advised not to advertise any information to networks that are outside your administrative control.

The following example demonstrates usage of this feature:

```
#
ospf <process-id>
silent-interface all
undo silent-interface <interface>
#
```

Route filtering

To reduce the possibility of introducing false routing information to the network, you must utilize route filtering. Unlike the **silent-interface** command, routing occurs on interfaces once route filtering is enabled, but the information that is advertised or processed is limited.

For RIP, using the **filter-policy** command with the **export** key word limits what information is advertised, while use of the **import** key word limits what updates are processed. The **filter-policy** command is available for OSPF, but it does not prevent a router from propagating filtered routes. Instead, the **filter** command can be used.

The following RIP example filters outbound advertisements with the **filter-policy** command and a prefix list:

```
#
ip ip-prefix <list-name> index 10 permit <ip-address> <mask-length>
#
rip <process-id>
silent-interface all
undo silent-interface <interface>
filter-policy ip-prefix <list-name> export <interface>
#
```

The following RIP example filters inbound updates with a prefix list:

```
#
ip ip-prefix <list-name> index 10 permit <ip-address> <mask-length>
#
rip <process-id>
silent-interface all
undo silent-interface <interface>
filter-policy ip-prefix <list-name> import <interface>
#
```

For more information, see "Configuring Inbound/Outbound Route Filtering in RIP" in the *Layer-3 IP Routing Configuration Guide*.

The following OSPF example utilizes a prefix list with the OSPF-specific **filter** command:

```
#
```

```
ip ip-prefix <list-name> index 10 permit <ip-address> <mask-length>
#
ospf <process-id>
area <area-id>
filter ip-prefix <list-name> import
#
```

For more information on OSPF Area Border Router (ABR) Type 3 link-state advertisements filtering, see “Configuring ABR Type-3 LSA Filtering in OSPF” in the *Layer-3 IP Routing Configuration Guide*.

Securing Virtual Router Redundancy Protocol

Virtual Router Redundancy Protocol (VRRP) provides resiliency and redundancy for devices that are acting as default gateways.

By default, VRRP communicates using unauthenticated communication. This kind of communication can allow an attacker to pose as a fake device to assume a network’s default gateway role. This takeover would allow an attacker to perform a man-in-the-middle attack and intercept all user traffic that exits the network.

In order to prevent this type of attack, VRRP supported by HP Comware software includes an authentication capability using either MD5 or plain text. Because of the threat posed by unauthenticated VRRPs, it is recommended that instances of these protocols use MD5 authentication. The following configuration example demonstrates the use of VRRP MD5 authentication:

```
#
interface Ethernet0/1/0
 vrrp vrid virtual-router-id authentication-mode md5 <key>
#
```

For more information, see “VRRP” in the *High Availability Configuration Guide*.

Data plane

Although the data plane is responsible for moving data from source to destination, within the context of security, the data plane is the least important of the three planes. It is for this reason that when you are securing a network device, it is important to protect the management and control planes in preference over the data plane.

However, within the data plane itself, there are many features and configuration options that can help secure traffic. The sections that follow detail these features and options so that you can more easily secure your network.

General data plane hardening

The vast majority of data plane traffic flows across the network as determined by the network’s routing configuration. However, IP network functionality exists to alter the path of packets across the network. Features such as IP Options, specifically the source routing option, form a security challenge in today’s networks.

The use of Transit ACLs is also relevant to the hardening of the data plane. For more information, see the “Filtering transit traffic with Transit ACLs” section of this document.

Disable ICMP redirects

ICMP redirects are used to inform a network device of a better path to an IP destination.

In some situations, it may be possible for an attacker to cause the device to send many ICMP redirect messages, resulting in an elevated CPU load. For this reason, it is recommended that the transmission of ICMP redirects be disabled. By default, HP Comware software does not send a redirect if it receives a packet that must be routed through the interface it was received from.

ICMP redirects are disabled using the **undo ip redirects** command in system view, as shown in the following example configuration:

```
#
undo ip redirects
#
```

For more information on the **undo ip redirects** command, see “IP Performance Optimization” in the *Layer-3 IP Services Configuration Guide*.

Disable or limit IP Directed broadcasts

IP Directed Broadcasts make it possible to send an IP broadcast packet to a remote IP subnet. Once it reaches the remote network, the forwarding IP device sends the packet as a Layer 2 broadcast to all stations on the subnet. This directed broadcast functionality has been leveraged as an amplification and reflection aid in several attacks, including the Smurf attack.

Current versions of HP Comware products have this functionality disabled by default; however, it can be enabled via the **ip forward-broadcast** command.

If a network absolutely requires directed broadcast functionality, its use should be controlled. This is possible using an access control list as an option to the **ip forward-broadcast** command. The following configuration example limits directed broadcasts to those UDP packets originating at a trusted network, 192.168.1.0/24:

```
#
acl number 3001
 rule 0 permit udp source 192.168.1.0 0.0.0.255
#
interface Ethernet 0/1/0
 ip forward-broadcast acl 3001
#
```

For more information about the **ip forward-broadcast** command, see “IP Performance Optimization Configuration” in the *Layer-3 IP Services Configuration Guide*.

Filtering transit traffic with Transit ACLs

ICMP packet filtering

The Internet Control Message Protocol (ICMP) was designed as a control protocol for IP. As a result, the messages it conveys can have far-reaching ramifications on TCP and IP in general. ICMP is used by the network troubleshooting tools ping and traceroute, as well as by Path MTU Discovery; however, external ICMP connectivity is rarely needed for the proper operation of a network.

HP Comware software provides functionality to specifically filter ICMP messages by name or type and code.

The following example ACL allows ICMP from trusted networks while blocking all ICMP packets from other sources:

```
#
acl number 3000 name ACL-TRANSIT-IN
#
# Permit ICMP packets from trusted networks only
#
rule permit icmp source <trusted-networks>
#
# Deny all other ICMP traffic.
#
rule deny icmp
#
```

Filtering IP fragments

As detailed previously in the “Limiting access to the network with infrastructure ACLs” section of this document, the filtering of fragmented IP packets can pose a challenge to security devices.

Because of the nonintuitive nature of fragment handling, IP fragments are often inadvertently permitted by ACLs. Fragmentation is also often used in attempts to evade detection by intrusion detection systems. It is for these reasons that IP fragments are often used in attacks and should be explicitly filtered at the top of any configured traffic ACLs. The example ACL that follows includes comprehensive filtering of IP fragments. The functionality illustrated in this example must be used in conjunction with the functionality of the previous examples:

```
#
acl number 3000 name ACL-TRANSIT-IN
#
# Deny IP fragments using protocol-specific ACEs to aid in classification of attack traffic.
#
rule deny tcp fragment
rule deny udp fragment
rule deny icmp fragment
rule deny ip fragment
#
```

Anti-spoofing protections

Many attacks utilize source IP address spoofing to be effective or to conceal the true source of an attack and hinder accurate traceback. HP Comware provides Unicast Reverse Path Forwarding (URPF) and IP Source Guard (IPSG) to deter attacks that rely on source IP address spoofing. In addition, ACLs and null routing are often deployed as a manual means of spoofing prevention.

IP Source Guard is effective at reducing spoofing for networks that are under direct administrative control by performing switch port, MAC address, and source address verification. URPF provides source network verification and can reduce spoofed attacks from networks that are not under direct administrative control. Port security can be used in order to validate MAC addresses at the access layer. ARP detection mitigates attack vectors that utilize ARP poisoning on local segments.

URPF

URPF enables a device to verify that the source address of a forwarded packet can be reached through the interface that received the packet. You must not rely on URPF as the only protection against spoofing. Spoofed packets could enter the network through a URPF-enabled interface if an appropriate return route to the source IP address exists.

URPF can be configured in one of two modes: loose or strict. In cases where there is asymmetric routing, loose mode configuration is preferred because strict mode is known to drop packets in these situations.

The following example illustrates configuration of this feature:

```
#
interface Ethernet0/1/0
 ip urpf loose
#
```

URPF can be configured globally or on the interface, depending on the device model.

For more information about the configuration and use of URPF, see “URPF” in the *Security Configuration Guide*.

IP source guard

IP source guard is an effective means of spoofing prevention in Layer 2 access mode.

After receiving a packet, an IP source guard-enabled port obtains the key attributes (source IP address, source MAC address, and VLAN tag) of the packet and then looks up the binding entries of the IP source guard for a match. If there is a match, the port forwards the packet; otherwise, the port discards the packet. You can enable this feature on a port connected to terminals to block illegal access (such as IP spoofing) and improve port security.

HP IP source guard supports static and dynamic entries. You can configure static entries in scenarios where there are only a few hosts in a LAN and their IP addresses are manually configured. For example, you can configure a static entry on a port that connects a server so that the port receives and sends packets from/to only the server.

Following is an example of static entry configuration:

#

Configure Ethernet 1/2 on Device B to allow only packets from host A with MAC address 00010203-0406 and IP address 192.168.1.1 to pass.

```
<DeviceB> system-view

[DeviceB] interface ethernet 1/2

[DeviceB-Ethernet1/2] user-bind ip-address 192.168.0.1 mac-address 0001-0203-0406
[DeviceB-Ethernet1/2] quit
```

#

Dynamic IP source guard entries are generated dynamically according to client entries on the DHCP snooping or DHCP relay agent device. They are suitable for scenarios where many hosts are in a LAN and DHCP is used to allocate IP addresses to the hosts. Once DHCP allocates an IP address to a client, IP source guard automatically adds the entry to allow the client to access the network. A person using an IP address not obtained through DHCP cannot access the network. Dynamic IPv6 source guard entries are obtained from client entries on the ND snooping device.

Following is a dynamic entry configuration example (DHCP snooping must have been enabled.)

#

Enable dynamic entry generation on Ethernet 1/1.

```
[Device] interface ethernet 1/1

[Device-Ethernet1/1] ip check source ip-address mac-address

[Device-Ethernet1/1] quit
```

#

Port security

Port security is a MAC address-based network access control mechanism. It is an extension to IEEE 802.1X and MAC authentication. It prevents access from unauthorized devices by checking the source MAC address of inbound traffic and access to unauthorized devices by checking the destination MAC address of outbound traffic.

With port security enabled, frames whose source MAC addresses cannot be learned by the device in a security mode are considered illegal. The events that users do not pass IEEE 802.1X authentication or MAC authentication are considered illegal.

Upon detection of illegal frames or events, the device takes the predefined action automatically. While enhancing the system security, this reduces your maintenance burden greatly. The illegal packets include:

- Packets whose source MAC addresses are not learned
- Packets failing authentication

The following table describes the port security modes.

Port security mode	Description
noRestrictions	In this mode, port security is disabled on the port and access to the port is not restricted.
autoLearn	The port in this mode adds learned and configured secure MAC address entries into the secure MAC address table. When the maximum number of secure MAC addresses is reached, the port changes to secure mode.
secure	In this mode, the port does not learn new MAC addresses, and permits only packets whose source MAC address matches a secure MAC address entry to pass.
userLogin	A port in this mode performs 802.1X authentication and implements port-based access control. The port can service multiple 802.1X users. If one 802.1X user passes authentication, all the other 802.1X users of the port can access the network without authentication.
userLoginSecure	A port in this mode performs 802.1X authentication and implements MAC-based access control. The port services only one user passing 802.1X authentication.
userLoginWithOUI	This mode is similar to the userLoginSecure mode. The difference is that a port in this mode also permits frames from a MAC address that contains a specified organizationally unique identifier (OUI).
macAddressWithRadius	A port in this mode performs MAC address authentication on users.
macAddressOrUserLoginSecure	This mode is the combination of the macAddressWithRadius and userLoginSecure modes. For wired users, the port performs MAC authentication upon receiving non-802.1X frames and performs 802.1X authentication upon receiving 802.1X frames.
macAddressElseUserLoginSecure	This mode is the combination of the macAddressWithRadius and userLoginSecure modes. For non-802.1X frames, a port in this mode performs only MAC authentication. For 802.1X frames, it performs MAC authentication and then, if the MAC authentication fails, 802.1X authentication.
userLoginSecureExt	A port in this mode performs MAC-based 802.1X authentication and allows multiple 802.1X users to have access.
macAddressOrUserLoginSecureExt	This mode is similar to the macAddressOrUserLoginSecure mode, except that a port in this mode allows multiple 802.1X and MAC users to have access.
macAddressElseUserLoginSecureExt	This mode is similar to the macAddressElseUserLoginSecure mode, except that a port in this mode allows multiple 802.1X and MAC users to have access.

The following configuration example enables MAC address learning on a port and sets the maximum number of MAC addresses the port can learn to 10:

```
#
[HP]port-security enable
 Please wait.......................... Done.
  [HP-Ethernet0/4/1]port-security max-mac-count 10
[HP-Ethernet0/4/1]port-security port-mode autolearn
[HP-Ethernet0/4/1]di th
#
interface Ethernet0/4/1
 port link-mode bridge
 port-security max-mac-count 10
 port-security port-mode autolearn
#
```

ARP Detection

ARP Detection can be utilized to mitigate ARP poisoning attacks on local segments. An ARP poisoning attack is a method in which an attacker sends falsified ARP information to a local segment. This information is designed to corrupt the ARP cache of other devices. Often an attacker uses ARP poisoning in order to perform a man-in-the-middle attack.

ARP Detection intercepts and validates the IP-to-MAC address relationship of all ARP packets on untrusted ports. In DHCP environments, ARP Detection utilizes the data that is generated by the DHCP snooping feature. In 802.1X environments, ARP Detection can use the user data generated by the 802.1x feature. ARP packets that are received on trusted interfaces are not validated and invalid packets on untrusted interfaces are discarded.

In non-DHCP or non-802.1X environments, the configuration of static client entries is required. Even if in DHCP environments, there may be some users such as servers or printers that use manually configured IP addresses. In such environments, static client entries are also the requisites when enabling ARP Detection.

The following command enables DHCP snooping:

```
#
dhcp-snooping
#
```

Once DHCP snooping has been enabled, the following commands enable ARP Detection:

```
#
vlan 1
arp detection enable
#
```

In non-DHCP or non-802.1x environments, static client entries on a port are required to enable ARP Detection. The following example demonstrates the basic configuration of a static client entry:

```
#
interface Ethernet0/4/0
user-bind ip-address <X.X.X.X> mac-address <H-H-H> vlan <VLAN ID>
#
```

For more information on how to configure ARP Detection, see “ARP Attack Protection” in the *Security Configuration Guide*.

Anti-spoofing ACLs

Manually configured ACLs can provide static anti-spoofing protection against attacks that utilize known unused and untrusted address space. Commonly, these anti-spoofing ACLs are applied to ingress traffic at network boundaries as a component of a larger ACL. Anti-spoofing ACLs require regular monitoring because they can frequently change. Spoofing can be reduced in traffic originating from the local network by applying outbound ACLs that limit the traffic to valid local addresses.

The following example demonstrates how ACLs can be used to limit IP spoofing. This ACL is applied in bound on the desired interface:

```
#
acl number 3001 name ACL-ANTISPOOF-IN
rule deny ip source 10.0.0.0 0.255.255.255
rule deny ip source 192.168.0.0 0.0.255.255
#
# For Switch, port ACL command is “packet-filter”
interface <interface>
packet-filter name ACL-ANTISPOOF-IN inbound
# For Router, port ACL command is “firewall packet-filter”
interface <interface>
```

```
firewall packet-filter name ACL-ANTISPOOF-IN inbound
#
```

Limiting the CPU impact of data plane traffic

The primary purpose of routers and switches is to forward packets and frames to their final destinations. These packets, which transit the devices deployed throughout the network, can impact a device's CPU operations. The data plane, which consists of traffic transiting the network device, should be secured to help ensure the operation of the management and control planes. If transit traffic can cause a device to process switch traffic, the control plane of a device can be affected, which may lead to an operational disruption.

Features and traffic types that impact the CPU

Although not exhaustive, this list includes types of data plane traffic that require special CPU processing and are process switched by the CPU:

- **IP options**
  Any IP packets with options included must be processed by the CPU.
- **Fragmentation**
  Any IP packet that requires fragmentation must be passed to the CPU for processing.
- **Time-to-Live (TTL) expiry**
  Packets that have a TTL value less than or equal to 1 require Internet Control Message Protocol Time Exceeded (ICMP Type 11, Code 0) messages to be sent, which results in CPU processing.
- **ICMP unreachables**
  Packets that result in ICMP unreachable messages due to routing, MTU, or filtering are processed by the CPU.
- **Traffic requiring an ARP request**
  Destinations for which an ARP entry does not exist require processing by the CPU.
- **Non-IP traffic**
  All non-IP traffic is processed by the CPU.

For more information about data plane hardening, see the "General data plane hardening" section of this document.

Traffic identification and traceback

At times, you need to quickly identify and traceback network traffic, especially during incident response or poor network performance. NetStream and classification ACLs are two primary methods to accomplish this. Using HP Comware, NetStream can provide visibility into all network traffic. Additionally, NetStream can be implemented with collectors that can provide long-term trending and automated analysis. Classification ACLs are a component of ACLs and require preplanning to identify specific traffic and manual intervention during analysis. The sections that follow provide a brief overview of each feature.

NetStream

NetStream identifies anomalous and security-related network activity by tracking network flows. NetStream data can be viewed and analyzed via the CLI, or the data can be exported to a NetStream collector and data analyzer for aggregation and analysis. A NetStream data analyzer, through long-term trending, can provide network behavior and usage analysis. NetStream, which can be configured on both routers and switches, functions by performing analysis on specific attributes within IP packets and creating flows. Version 9 is the most flexible format and allows users to define templates with different statistics fields.

The following example illustrates the basic configuration of this feature. NetStream can be enabled on an interface, or through QoS policy or port mirroring. Different devices choose one of the approaches based on device model:

Approach I, enable NetStream on an interface.

```
#
interface Ethernet0/1/0
ip netstream { inbound | outbound }
#
```

Approach II, enable NetStream through QoS policy.

#

```
ip netstream { inbound | outbound }
```

#

```
traffic behavior <behavior-name>
mirror-to interface net-stream <interface-number>
```

#

Approach III, enable NetStream through port mirroring.

#

```
ip { inbound | outbound }
```

#

```
interface Ethernet0/1/0
ip netstream mirror-to interface net-stream <interface-numbr>
```

#

Following is an example of NetStream output from the CLI. The `If(Direc)` attribute can be beneficial in traceback:

#

```
<Sysname> display ip netstream cache
IP netstream cache information:
  Stream active timeout (in minutes)         : 60
  Stream inactive timeout (in seconds)       : 10
  Stream max entry number                    : 1000
    IP active stream entry number            : 1
  MPLS active stream entry number            : 2
    L2 active stream entry number            : 1
  IPL2 active stream entry number            : 1
IP stream entries been counted           : 10
  MPLS stream entries been counted           : 20
    L2 stream entries been counted           : 10
  IPL2 stream entries been counted           : 20
Last statistics reset time                 : 01/01/2000, 00:01:02

IP packet size distribution (1103746 total packets):
1-32   64   96  128  160  192  224  256  288  320  352  384  416  448  480
.249 .694 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000

512  544  576 1024 1536 2048 2560 3072 3584 4096 4608 >4608
.000 .000 .027 .000 .027 .000 .000 .000 .000 .000 .000 .000

Protocol          Total    Packets    Stream  Packets Active(sec) Idle(sec)
                 Streams        /Sec     /Sec   /stream     /stream     /stream

TCP-Telnet     2656855      372         4       86        49        27
TCP-FTP        5900082       86         9        9        11        33
```

```
TCP-FTPD       3200453     1006          5      193         45         33
TCP-WWW      546778274    11170        887       12          8         32
TCP-other     49148540     3752         79       47         30         32
UDP-DNS      117240379      570        190        3          7         34
UDP-other     45502422     2272         73       30          8         37
ICMP          14837957      125         24        5         12         34
IP-other         77406        5          0       47         52         27

Type DstIP(Port)           SrcIP(Port)           Pro ToS If(Direc)   Pkts
    DstMAC(VLAN)           SrcMAC(VLAN)
    TopLblType(IP/MASK)    Lbl-Exp-S-List

IP 11.1.1.1(1024)          11.1.1.2(21)            6  128 ET1/0(I)   42996
L2 0012-3f86-e94c(10)      0012-3f86-e86a(0)              ET1/4/0(I)  1253
MPLS LDP(3.3.3.3/24)       1:18-6-0                       ET1/1(O)     291
                              2:24-6-0
                              3:30-6-1
IP& 192.168.123.1(2048)    192.168.1.1(0)           1    0 ET1/1(O)      10
L2 0012-3f86-e95d(0)       0012-3f86-e116(1008)

IP& 172.16.1.1(68)         172.16.2.1(67)          17   64 ET1/2(I)    1848
MPLS LDP(4.4.4.4/24)       1:55-6-0
                               2:16-6-1
```

#

For more information on NetStream capabilities, see “NetStream” in the *Network Management and Monitoring Configuration Guide*.

sFlow

sFlow is a traffic monitoring technology used to collect and analyze traffic statistics.

The sFlow system involves an sFlow agent and a remote sFlow collector. The sFlow agent collects traffic statistics and packet information from sFlow-enabled interfaces, and encapsulates them into sFlow packets. When the sFlow packet buffer is full, or the age time of sFlow packets is reached, the sFlow agent sends the packets to a specified sFlow collector. The sFlow collector analyzes the sFlow packets and displays the results.

sFlow has the following two sampling mechanisms:

- Flow sampling—packet-based sampling used to obtain packet content information
- Counter sampling—time-based sampling used to obtain port traffic statistics

sFlow has the following advantages:

- Supporting traffic monitoring on Gigabit Ethernet and higher-speed networks
- Providing good scalability to allow one sFlow collector to monitor multiple sFlow agents
- Saving costs by embedding the sFlow agent in a device, instead of using a dedicated sFlow agent device; only the sFlow agent is supported on HP Comware devices

Only the sFlow agent is supported on HP Comware devices.

The following example shows the basic configuration of sFlow.

Specify an sFlow collector.

#

```
sflow collector < collector-id > ip < ip-address >
```

#

Specify an IP address for the sFlow agent, and the sFlow version.

#

```
sflow agent { ip ip-address | ipv6 ipv6-address }
sflow version { 4 | 5 }
```

#

Configure flow sampling:

#

```
interface Ethernet0/1/0
sflow sampling-mode { determine | random }
sflow sampling-rate < rate >
sflow flow max-header < length >
sflow flow collector < collector-id >
```

#

Configure counter sampling:

#

```
interface Ethernet0/1/1
sflow counter interval < seconds >
sflow counter collector < collector-id >
```

#

For more information about sFlow, see "sFlow" in the *Network Management and Monitoring Configuration Guide*.

Classification ACLs

Classification ACLs provide visibility into traffic that traverses an interface. Classification ACLs do not alter the security policy of a network and are typically constructed to classify individual protocols, source addresses, or destinations. For example, a match item that permits all traffic could be separated into specific protocols or ports. This more granular classification of traffic into specific match items can help provide an understanding of the network traffic because each traffic category has its own hit counter. An administrator may also separate the implicit deny at the end of an ACL into granular match items to help identify the types of denied traffic.

An administrator can expedite an incident response by using classification ACLs with **display acl** and **reset acl counter** commands.

The following example illustrates the configuration of a classification ACL to identify traffic:

#

```
acl number 3002 name ACL-SMB-CLASSIFY
description Classification of SMB specific TCP traffic
rule deny tcp destination-port eq 139
rule deny tcp destination-port eq 445
rule deny ip
```

#

Use the **display acl** command to view the configuration of ACL 3002. (The ACL counters can be cleared by using the **reset acl counter** command.)

#

```
[HP] display acl 3002
Advanced ACL 3002, named ACL-SMB-CLASSIFY, 3 rules,
Classification of SMB specific TCP traffic
ACL's step is 5
 rule 0 deny tcp destination-port eq 139 (10 times)
 rule 5 deny tcp destination-port eq 445 (10 times)
 rule 10 deny ip (205 times)
#
```

Access control with VLAN QoS policy and port access control lists

VLAN access control lists (VACLs), or VLAN QoS policy and port ACLs (PACLs), provide the capability to enforce access control on non-routed traffic closer to endpoint devices than ACLs applied to routed interfaces.

The sections that follow provide an overview of the features, benefits, and potential usage scenarios of VACLs and PACLs.

Access control with VLAN QoS policy

VACLs, or VLAN QoS policies that apply to all packets that enter the VLAN, provide the capability to enforce access control on intra-VLAN traffic. This is not possible using ACLs on routed interfaces. For example, a VLAN QoS policy may be used to prevent hosts that are contained within the same VLAN from communicating with each other, thereby reducing opportunities for local attackers or worms to exploit a host on the same network segment. In order to deny packets from using a VLAN QoS policy, you can create an ACL that matches the traffic and, in the VLAN QoS policy, set the action to drop. Once a VLAN QoS policy is configured, all packets that enter the LAN are sequentially evaluated against the configured VLAN QoS policy.

The following example utilizes an extended named access list that illustrates the configuration of this feature:

```
#
acl number 3003 name <acl-name>
rule permit <protocol> <source-address> <source-port> <destination-address>
<destination-port>
#
[HP]traffic behavior<name>
[HP-behavior-b1] <permit|deny>
[HP]traffic classifier <name>
[HP-classifier-b1] if-match <acl-name>
[HP]qos policy <name>
[HP-qospolicy-c1]classifier <name> behavior <name>
#
[HP]qos vlan-policy <policy-name> vlan 100 inbound
#
```

Access control with PACLs

PACLs can only be applied to the inbound direction on Layer 2 physical interfaces of a switch. The syntax for creating PACLs, which take precedence over VLAN QoS policies and router ACLs, is the same as it is for router ACLs. An ACL applied to a Layer-2 interface is referred to as a PACL. Configuration involves creating an IPv4, IPv6, or MAC ACL and applying it to the Layer 2 interface.

The following example utilizes an extended ACL to illustrate the configuration of this feature:

```
#
acl number 3003 name <acl-name>
```

```
rule permit <protocol> <source-address> <source-port> <destination-address>
<destination-port>
#
interface <interface>
packet-filter name <acl-name> inbound
```

Access control with MAC

A MAC ACL can be applied on an IP network and instructs the forwarding engine to not inspect the IP header. The result is that you are able to use a MAC access list in the IP environment.

Using private VLANs

Private VLANs (PVLANs) are a Layer 2 security feature that limits connectivity between workstations or servers within a VLAN. Without PVLANs, all devices on a Layer 2 VLAN can communicate freely.

Networking situations exist where security can be aided by limiting communication between devices on a single VLAN. For example, PVLANs are often used to prohibit communication between servers in a publicly accessible subnet. Should a single server become compromised, the lack of connectivity to other servers due to the application of PVLANs may help limit the compromise to the one server.

There are three types of private VLANs: isolated VLANs, community VLANs, and primary VLANs. The configuration of PVLANs makes use of primary and secondary VLANs. The primary VLAN contains all promiscuous ports, which are described later, and includes one or more secondary VLANs, which can be either isolated or community VLANs.

Note: Only some product models support insolated VLANs and promiscuous ports.

Isolated VLANs

The configuration of a secondary VLAN as an isolated VLAN completely prevents communication between devices in the secondary VLAN. There may only be one isolated VLAN per primary VLAN, and only promiscuous ports may communicate with ports in an isolated VLAN. Isolated VLANs should be used on untrusted networks such as networks that support guests.

This configuration example configures VLAN 11 as an isolated VLAN and associates it to the primary VLAN, VLAN 20. The following example also configures interface GigabitEthernet1/0/1 as an isolated port in VLAN 11:

```
#
vlan 11
 isolated-vlan enable
#
vlan 20
 isolate-user-vlan enable
#
interface GigabitEthernet1/0/1
description *** Port in Isolated VLAN ***
port isolate-user-vlan host
port access vlan 11
#
isolate-user-vlan 20 secondary 11
#
```

Community VLANs

A secondary VLAN that is configured as a community VLAN allows communication among members of the VLAN as well as with any promiscuous ports in the primary VLAN. However, no communication is possible between any two community VLANs or from a community VLAN to an isolated VLAN. Community VLANs must be used to group servers that need connectivity with one another, but where connectivity to all other devices in the VLAN is not required. This scenario is common in a publicly accessible network or anywhere that servers provide content to untrusted clients.

The following example configures a single community VLAN and configures switch port GigabitEthernet1/0/2 as a member of that VLAN. The community VLAN, VLAN 12, is a secondary VLAN to primary VLAN 20.

Note: A secondary VLAN is considered a community VLAN by default.

```
#
vlan 12
#
vlan 20
 isolate-user-vlan enable
#
interface GigabitEthernet1/0/2
description *** Port in Community VLAN ***
 port isolate-user-vlan host
port access vlan 12
#
isolate-user-vlan 20 secondary 12
#
```

Promiscuous ports

Switch ports that are placed into the primary VLAN are known as *promiscuous* ports. Promiscuous ports can communicate with all other ports in the primary and secondary VLANs. Router or firewall interfaces are the most common devices found on these VLANs.

The following configuration example combines the previous isolated and community VLAN examples and adds the configuration of interface GigabitEthernet1/0/12 as a promiscuous port:

```
#
vlan 11
 isolated-vlan enable
#
vlan 12
#
vlan 20
 isolate-user-vlan enable
#
interface GigabitEthernet1/0/1
description *** Port in Isolated VLAN ***
port isolate-user-vlan host
port access vlan 11
#
interface GigabitEthernet1/0/2
description *** Port in Community VLAN ***
 port isolate-user-vlan host
```

```
port access vlan 12
#
interface GigabitEthernet1/0/12
port link-mode bridge
description *** Promiscuous Port ***
port isolate-user-vlan promiscuous
port link-type hybrid
port hybrid vlan 11 to 12 20 tagged
#
isolate-user-vlan 20 secondary 11 12
#
```

When implementing PVLANs, it is important to ensure that the Layer 3 configuration in place supports the restrictions that are imposed by PVLANs and does not allow the PVLAN configuration to be subverted.

Layer 3 filtering using a router ACL or firewall can prevent the subversion of the PVLAN configuration.

Port isolation

Port isolation is another Layer 2 security feature that limits connectivity between workstation or servers within a VLAN. Without port isolation, all devices on a Layer 2 VLAN can communicate freely.

Networking situations exist where security can be aided by limiting communication between devices on a single VLAN. For example, port isolation is often used to prohibit communication between servers in a publicly accessible subnet. Should a single server become compromised, the lack of connectivity to other servers due to the application of port isolation may help limit the compromise to the one server.

Port isolation includes isolated ports, uplink port, and isolation groups. HP Comware supports creating multiple isolation groups, each of which can contain multiple isolated ports and one uplink port.

Layer 2 traffic is isolated among member ports in an isolation group.

Note: Not all product models support uplink and isolation group functions.

Isolated ports

The configuration of some ports in a VLAN as isolated ports completely prevents communication between devices in the VLAN. Isolated ports should be used on untrusted networks that only access the Internet without needing to communicate with each other.

The following configuration example configures all the ports of VLAN 20 as isolated ports. Interface GigabitEthernet1/0/10 and GigabitEthernet1/0/11 are in VLAN 20:

```
#
interface GigabitEthernet1/0/10
description *** Isolated Port ***
port access vlan 20
port-isolate enable
#
interface GigabitEthernet1/0/11
description *** Isolated Port ***
port access vlan 20
port-isolate enable
#
```

Uplink port

The uplink port of an isolation group can communicate with isolated ports in the group so that the isolated ports can access other networks through the uplink port without needing Layer 3 forwarding. If your device does not support an uplink port feature, the isolated ports in a Layer 2 VLAN need Layer 3 forwarding to access other networks. The following configuration example configures G1/0/10 and G1/0/11 in VLAN 20 as isolated ports, and configures Ten-GigabitEthernet1/0/49 as the uplink port.

```
#
interface GigabitEthernet1/0/10
description *** Isolated Port ***
port access vlan 20
port-isolate enable
#
interface GigabitEthernet1/0/11
description *** Isolated Port ***
port access vlan 20
port-isolate enable
#
interface Ten-GigabitEthernet1/0/49
description *** Uplink Port ***
port access vlan 20
port-isolate uplink-port
#
```

Isolation groups

This configuration example configures G1/0/10 and G1/0/11 in VLAN 20 as isolated ports in isolation group 1, and configures Ten-GigabitEthernet1/0/49 as the uplink port of isolation group 1; configures G1/0/20 and G1/0/21 in VLAN 20 as isolated ports in isolation group 2; and configures TenGigabitEthernet1/0/50 as the uplink port of isolation group 2.

```
#
interface GigabitEthernet1/0/10
description *** Isolated Port of Group1 ***
port access vlan 20
port-isolate enable group 1
#
interface GigabitEthernet1/0/11
description *** Isolated Port of Group1 ***
port access vlan 20
port-isolate enable group 1
#
#
interface GigabitEthernet1/0/20
description *** Isolated Port of Group2 ***
port access vlan 20
port-isolate enable group 2
#
interface GigabitEthernet1/0/21
```

```
description *** Isolated Port of Group2 ***
port access vlan 20
port-isolate enable group 2
#
interface Ten-GigabitEthernet1/0/49
description *** Uplink Port of Group1 ***
port access vlan 20
port-isolate uplink-port group 1
#
interface Ten-GigabitEthernet1/0/50
description *** Uplink Port of Group2 ***
port access vlan 20
port-isolate uplink-port group 2
#
```

For more information about port isolation, see “Port Isolation” in the *Layer-2 LAN Switching Configuration Guide*.

Keywords: secure, management plane, control plane, data plane

Abstract: This document describes how to secure HP Comware devices.

Acronyms:

Acronym	Full spelling
AAA	authentication, authorization, and accounting
TFTP	Trivial File Transfer Protocol
SFTP	Security FTP
NTP	Network Time Protocol
UDP	User Datagram Protocol
DoS	denial of service
ACL	access control list
TTY	true type terminal
VTY	virtual type terminal
LDAP	Lightweight Directory Access Protocol
SNMP	Simple Network Management Protocol
RIP	Routing Information Protocol
OSPF	open shortest path first
IGP	Interior Gateway Protocol
MD5	Message Digest 5
ABR	Area Border Router
BGP	Border Gateway Protocol
ACE	access control entry

4AA4-4160ENW, Created October 2012