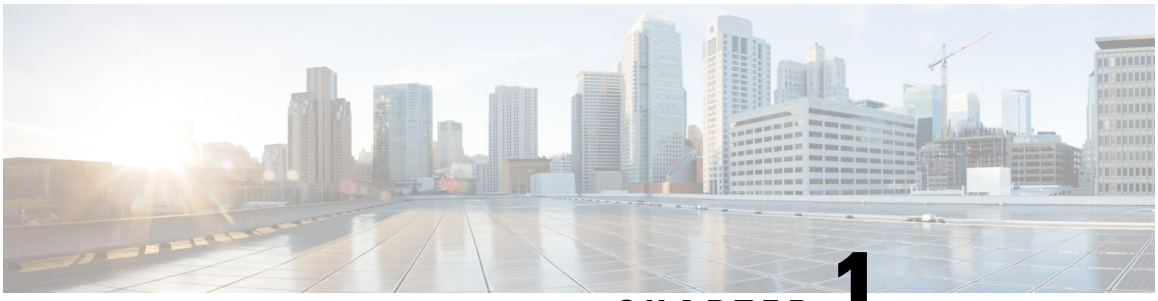
Security Configuration, Cisco Catalyst PON Series Switches

First Published: 2020-11-09

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS



CHAPTER 1

Configuring Access Control List

Information About Access Control List

An Access Control List (ACL) is an ordered set of rules that you can use to filter traffic. Each rule specifies a set of conditions that a packet must satisfy to match the rule. You can use ACLs to protect your network and specific hosts from unwanted traffic.

When an ACL applies to a packet, it tests the packet against the conditions of all rules. The first match determines whether the packet is permitted or denied. If there is no match, the device applies the applicable default rule. The device continues processing packets that are permitted and drops packets that are denied.

Types of ACL

ACL is divided into the following types, based on the purpose of application:

- **Standard ACL**: Defines the rules based on source IP addresses only. Standard ACLs control traffic by comparing the source address of the IP packet to the addresses defined in the ACL.
- **Extended ACL**: Defines the rules based on the source IP address, destination IP address, protocol type, and the protocol attributes of packets.
- **Layer 2 ACL**: Defines the rules based on the source MAC address, destination MAC address, VLAN priority, and Layer 2 protocol type.

Matching Order

An ACL consists of multiple permit or deny rules. The rules may overlap or conflict. In such cases, the matching order decides which rule is executed. ACL supports two matching orders:

- **config**: Matches the ACL rules according to the configuration order.
- **auto**: Matches the ACL rules according to the depth-first rule, wherein the longest subitem in a rule takes priority. The longest subset of a rule is matched first before the rule.

Naming Methods

An ACL is classified into the following types, based on the naming methods:

- **Numbered ACL**: The ACL is identified by the number assigned to it. You can create an ACL and assign a number to it. If you don't specifiy a number, the system assigns a number to the created ACL.

  For a Standard ACL, the numbers range from 1 through 99. You can create upto 99 Standard ACLs.

  For an Extended ACL, the numbers range from 100 through 199. You can create upto 100 Extended ACLs.

  For a Layer 2 ACL, the numbers range from 200 through 299. You can create a maximum of 100 Layer 2 ACLs.

- **Named ACL**: The ACL is identified by the name assigned to it. A named ACL consists of a name and number.

  You can create a maximum of 1000 named ACLs and also define upto 128 subrules for each ACL.

Time Range

A time-based ACL allows for access control based on time. You can create a time range to define specific times of the day and week in order to implement time-based ACLs. A time range is identified by a name and then referenced by a function. Time range can depend on the network access behavior of the users and network congestion condition.

Time range configurations include the absolute time range and periodic time range. A periodic time range configuration is in the form of days of the week. An absolute time range is in the form of start time to the end time.

How to Configure Access Control List

The following sections provide information about configuring Access Control List:

Configuring ACL Matching Order

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device#configure terminal`	Enters the global configuration mode.
Step 2	**access-list** *access-list* **match-order{ auto | config}** **Example:** `Device(config)#access-list 2 match-order config`	(Optional) Configures the ACL matching order. The default matching order is **config**.

Example Configuration for ACL Matching Order

The following example shows the **config** matching order in a two subitem ACL list.

```
Device#configure terminal
Device(config)#access-list 1 deny any

Device(config)#access-list 1 permit 1.1.1.1 0

Device(config)#show access-list config 1
Standard IP Access List 1, match-order is config, 2 rule:
0  deny    any
permit  1.1.1.1  0.0.0.0
```

The following example shows the **auto** matching order in a two subitem ACL list.

```
Device#configure terminal
Device(config)#access-list 1 match-order auto
Device(config)#access-list 1 deny any
Device(config)#access-list 1 permit 1.1.1.1 0

Device(config)#show access-list config 1
Standard IP Access List 1,  match-order is auto, 2 rule:
0  permit  1.1.1.1  0.0.0.0
1  deny    any
```

Configuring Time Range

You can configure the time range for an ACL either as an absolute time or as a periodic time. Periodic time range is in the form of days of the week. An absolute time range is in the form of start time and end time.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device#configure terminal`	Enters the global configuration mode.
Step 2	**time-range** *name* **Example:** `Device(config)#time-range timeACL`	Creates a time range and enters the time range configuration mode.
Step 3	**absolute start** *HH:MM:SS YYYY/MM/DD* [**end** *HH:MM:SS YYYY/MM/DD* **Example:** `Device(config-timerange-timeACL)#absolute start 16:00:00 2020/03/30 end 16:00:00 2020/03/31`	Configures an absolute time range. If you do not specify the start date, the time ranges from the earliest time supported by the system to the configured end date. If you do not specify the end date, the time ranges from the configured start date to 2100/12/31 23:59.
Step 4	**periodic** *days-of-the-week HH:MM:SS* **to** [*days-of-the-week*] *HH:MM:SS* **Example:**	Configures a periodic time range.

	Command or Action	Purpose
	`Device(config-timerange-timeACL)#periodic weekdays 8:00:00 to 18:00:00`	
Step 5	**no time-range** [**all** | **name** *name* **Example:** `Device(config)#no time-range all`	(Optional) Removes the specified time range configuration.

Example Configuration for Time Range

The following example configures a time range from 16:00 on March 30, 2020 to 16:00 on March 31, 2020.

```
Device#configure terminal
Device(config)#time-range b
Device(config-timerange-b)#absolute start 16:00:00 2020/03/30 end 16:00:00 2020/03/31
Device(config-timerange-b)#show  time-range name b
Current time is: 10:19:16   2020/03/30  Monday
time-range: b ( Inactive )
absolute:  start  16:00:00  2020/03/30  end  16:00:00  2020/03/31
```

The following example configures a periodic time range, which ranges from 8:00 to 18:00 and Monday to Friday.

```
Device#configure terminal
Device(config)#time-range d
Device(config-timerange-d)#periodic weekdays 8:00:00 to 18:00:00
Device(config-timerange-b)#show time-range name d
Current time is: 10:23:33   2015/03/30  Monday
time-range:d ( Inactive )
periodic:  weekdays 08:00:00  to 18:00:00
```

Configuring Standard ACL

Standard ACLs can be Named ACLs or Numbered ACLs, as described in the following sections.

Configuring a Numbered Standard ACL

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device#configure terminal`	Enters the global configuration mode.
Step 2	**access-list** *num* {**permit** | **deny** } { *source-IPv4 / IPv6 source-wildcard* | **any** | **ipv6any** } [**time-range** *name*] **Example:** `Device(config)#access-list 4 deny any`	Defines a numbered ACL. The following are the parameters of the command: • **permit** | **deny**:

	Command or Action	Purpose
		Use the keyword **permit** to allow access. Use **deny** not to allow access. • *source-IPv4 / IPv6 source-wildcard* **| any | ipv6any**: Specifies the source address of the ACL rule. *source-IPv4 / IPv6 source-wildcard* specifies the source IP address (IPv4 or IPv6) range of the packet. A *source-wildcard* with a value of zero indicates the host address. **any** specifies any IPv4 source address. **ipv6any** specifies any IPv6 source address. • **time-range** *name*: Specifies the time range in which the ACL rule takes effect.
Step 3	**no access-list** [*number* **| name** *name* **|all**] **Example:** `Device(config)#no access-list 4`	(Optional) Removes the ACL specified. If **all** is specified, removes all the ACLs.

Example Configuration for a Standard ACL

The following example defines a numbered ACL to forbid the packets that have a source address of 10.0.0.1.

```
Device#configure terminal
Device(config)#access-list 1 deny 10.0.0.1 0
```

Configuring a Named Standard ACL

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device#configure terminal`	Enters the global configuration mode.
Step 2	**access-list standard** *name* **Example:** `Device(config)#access-list standard stdacl`	Defines a Standard ACL based on name and enters the ACL configuration mode.
Step 3	{**permit | deny** } { *source-IPv4 / IPv6 source-wildcard* **| any | ipv6any** } [**time-range** *name*] **Example:**	Configures the ACL rule. The following are the parameters of the command: • **permit | deny**:

	Command or Action	Purpose
	`Device(config-std-nacl-stdacl)#deny ipv6any`	Use the keyword **permit** to allow access. Use **deny** not to allow access. • *source-IPv4 / IPv6 source-wildcard* | **any** | **ipv6any**: Specifies the source address of the ACL rule. *source-IPv4 / IPv6 source-wildcard* specifies the source IP address (IPv4 or IPv6) range of the packet. A *source-wildcard* with a value of zero indicates the host address. **any** specifies any IPv4 source address. **ipv6any** specifies any IPv6 source address. • **time-range** *name*: Specifies the time range in which the ACL rule takes effect.
Step 4	**no access-list** [*number* | **name** *name* | **all**] **Example:** `Device(config)#no access-list name stdacl`	(Optional) Removes the ACL specified.

Example Configuration for a Standard ACL

The following example defines a named ACL to forbid the packets that have a source address of 10.0.0.2.

```
Device#configure terminal
Device(config)#access-list standard stdacl
Device(config-std-nacl-stdacl)#deny 10.0.0.2 0
```

Configuring Extended ACL

Extended ACLs can be Named ACLs or Numbered ACLs, as described in the following sections.

Configuring a Numbered Extended ACL

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device#configure terminal`	Enters the global configuration mode.
Step 2	**access-list** *num* {**permit** | **deny** } [*protocol*] [**established**] { *source-IPv4 / IPv6 source-wildcard* | **any**	Defines a number-based Extended ACL. The following are the parameters of the command:

	Command or Action	Purpose
	| ipv6any } [*source-port wildcard*] { *dest-IPv4 / IPv6 dest-wildcard* **| any | ipv6any** } [*dest-port wildcard*] [*icmp-type icmp-code*] [*igmp-type*] [**traffic-class** *traffic-class*] [**precedence** *precedence*] [**tos** *tos*] [**dscp** *dscp*] [**fragments**] [**time-range** *name*] **Example:** `Device(config)#access-list 102 permit tcp 10.0.0.1 0 ftp any`	• **permit | deny**: Use the keyword **permit** to allow access. Use **deny** not to allow access. • *source-IPv4 / IPv6 source-wildcard* **| any | ipv6any**: Specifies the source address from where the packets originate. *source-IPv4 / IPv6 source-wildcard* specifies the source IP address (IPv4 or IPv6) range of the packet. A *source-wildcard* with a value of zero indicates the host address. **any** specifies any IPv4 source address. **ipv6any** specifies any IPv6 source address. • *protocol*: Specifies the type of IP protocol. It is in the range of 1–255 by number. Select from GRE, ICMP, IGMP, IPinIP, OSPF, TCP, UDP, and ICMPv6 to specify the protocol by name. • **established**: Defines the SYN flag in TCP. A value 1 indicates that the flag is active. • *dest-IPv4 / IPv6 dest-wildcard* **| any | ipv6any**: Specifies the destination address to which the packets are being sent. *dest-IPv4 / IPv6 dest-wildcard* specifies the destination IP address (IPv4 or IPv6) range of the packet. An IPv4 address is in dotted decimal notation. An IPv6 address is in hexadecimal notation. A *dest-wildcard* with a value of zero indicates the host address. **any** specifies any IPv4 source address. **ipv6any** specifies any IPv6 destination address. • *source-port / dest-port wildcard*: Specifies the TCP or UDP source and destination port numbers. • *icmp-type icmp-code*: Specifies the type of ICMP protocol packet. It is valid only when protocol is configured as **icmp** or **icmpv6**.

	Command or Action	Purpose
		• *igmp-type*: Specifies the type of IGMP protocol packet. It is valid only when protocol is configured as **igmp**. • **traffic-class**: Specifies the traffic class for IPv6. • **precedence**: Specifies the precedence priority. IP precedence ranges from 0 through 7. • **tos**: Specifies the Type of Service (ToS) priority. The values range from 0 through 15. • **dscp**: Specifies the Differentiated Services Code Point (DSCP) priority value. • **fragments**: Specifies that the ACL rule is valid for nonfirst fragmented packets. This helps prevent fragment packet attacks. • **time-range** *name*: Specifies the time range in which the ACL rule takes effect.
Step 3	**no access-list** [*number* | **all**] **Example:** `Device(config)#no access-list 102`	(Optional) Removes the ACL specified.

Example Configuration for an Extended ACL

The following example defines a number-based Extended ACL to deny FTP packets that have a source address of 10.0.0.1.

```
Device#configure terminal
Device(config)#access-list 100 deny tcp 10.0.0.1 0 ftp any
```

Configuring a Named Extended ACL

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device#configure terminal`	Enters the global configuration mode.
Step 2	**access-list extended** *name* **Example:** `Device(config)#access-list extended extacl`	Defines a name-based Extended ACLand enters the ACL configuration mode.
Step 3	{ **permit** | **deny** } [*protocol*] [**established**] { *source-IPv4 / IPv6 source-wildcard* | **any** | **ipv6any** } [*source-port wildcard*] { *dest-IPv4 / IPv6 dest-wildcard* | **any** | **ipv6any** } [*dest-port wildcard*] [*icmp-type icmp-code*] [*igmp-type*] [**traffic-class** *traffic-class*] [**precedence** *precedence*] [**tos** *tos*] [**dscp** *dscp*] [**fragments**] [**time-range** *name*] **Example:** `Device(config-ext-nacl-extacl)#deny tcp 10.0.0.1 0 ftp any`	Defines a name-based Extended ACL. The following are the parameters of the command: • **permit | deny**: Use the keyword **permit** to allow access. Use **deny** not to allow access. • *source-IPv4 / IPv6 source-wildcard* **| any | ipv6any**: Specifies the source address from where the packets originate. *source-IPv4 / IPv6 source-wildcard* specifies the source IP address (IPv4 or IPv6) range of the packet. A *source-wildcard* with a value of zero indicates the host address. **any** specifies any IPv4 source address. **ipv6any** specifies any IPv6 source address. • *protocol*: Specifies the type of IP protocol. It is in the range of 1–255 by number. Select from GRE, ICMP, IGMP, IPinIP, OSPF, TCP, UDP, and ICMPv6 to specify the protocol by name. • **established**: Defines the SYN flag in TCP. A value 1 indicates that the flag is active. • *dest-IPv4 / IPv6 dest-wildcard* **| any | ipv6any**: Specifies the destination address to which the packets are being sent. *dest-IPv4 / IPv6 dest-wildcard* specifies the destination IP address (IPv4 or IPv6) range of the

	Command or Action	Purpose
		packet. An IPv4 address is in dotted decimal notation. An IPv6 address is in hexadecimal notation. A *dest-wildcard* with a value of zero indicates the host address. **any** specifies any IPv4 source address. **ipv6any** specifies any IPv6 destination address. • *source-port / dest-port wildcard*: Specifies the TCP or UDP source and destination port numbers. • *icmp-type icmp-code*: Specifies the type of ICMP protocol packet. It is valid only when protocol is configured as **icmp** or **icmpv6**. • *igmp-type*: Specifies the type of IGMP protocol packet. It is valid only when protocol is configured as **igmp**. • **traffic-class**: Specifies the traffic class for IPv6. • **precedence**: Specifies the precedence priority. IP precedence ranges from 0 through 7. • **tos**: Specifies the Type of Service (ToS) priority. The values range from 0 through 15. • **dscp**: Specifies the Differentiated Services Code Point (DSCP) priority value. • **fragments**: Specifies that the ACL rule is valid for nonfirst fragmented packets. This helps prevent fragment packet attacks. • **time-range** *name*: Specifies the time range in which the ACL rule takes effect.
Step 4	**no access-list** [*name* | **all**] **Example:** `Device(config)#no access-list all`	(Optional) Removes the ACL specified.

Configure an Extended ACL

The following example defines a name-based Extended ACL to deny FTP packets that have a source address of 10.0.0.2.

```
Device#configure terminal

Device(config)#access-list extended extacl
Device(config)#deny tcp 10.0.0.2 0 ftp any
```

Configuring Layer 2 ACL

Layer 2 ACLs establish rules based on Layer 2 information like MAC address, VLAN priorities, and Layer 2 protocol types. Layer 2 ACLs can be name-based or number-based.

Configuring a Numbered Layer 2 ACL

SUMMARY STEPS

1. **configure terminal**
2. **access-list** *num* {**permit** | **deny** } [*protocol*] [**cos** *vlan-pri*] **ingress**{ { [**inner-vid***vid*] [*start-vlan-id end-vlan-id*] [*source-mac-addr source-mac-wildcard*] [**interface** *interface-number*] } | **any** } **egress** { { [*dest-mac-addr dest-mac-wildcard*] [**interface** *interface-number* | **cpu**] } | **any** } [**time-range** *name*]
3. **no access-list** [*number* | **all**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device#configure terminal`	Enters the global configuration mode.
Step 2	**access-list** *num* {**permit**	**deny** } [*protocol*] [**cos** *vlan-pri*] **ingress**{ { [**inner-vid***vid*] [*start-vlan-id end-vlan-id*] [*source-mac-addr source-mac-wildcard*] [**interface** *interface-number*] }	**any** } **egress** { { [*dest-mac-addr dest-mac-wildcard*] [**interface** *interface-number*	**cpu**] }	**any** } [**time-range** *name*] **Example:** `Device(config)#access-list 202 deny arp ingress 00:00:00:00:00:01 0 egress`	Defines a number-based Layer 2 ACL. The following are the parameters of the command: • **permit**	**deny**: Use the keyword **permit** to allow access. Use **deny** not to allow access. • *protocol* : Specifies the type of protocol packet carried by the Ethernet frame. In hexadecimal notation, the range is 0 through FFFF. It is optional ion case of ARP, IP, RARP. • *protocol*: Specifies the type of IP protocol.

	Command or Action	Purpose
		It is in the range of 1–255 by number. Select from GRE, ICMP, IGMP, IPinIP, OSPF, TCP, UDP, and ICMPv6 to specify the protocol by name. • **cos**: Defines the priority of the VLAN tag. • **ingress** : Specifies the rule for the incoming packets at the ingress port. • **inner-vid**: Specifies the inner VLAN ID of a double-tagged packet. • *start-vlan-id end-vlan-id*: Specifies the range of VLANs. For a double-tagged packet, it is the VLAN ID of the outer tag. • *source-mac-addr source-mac-wildcard*: Specifies the source MAC address options. *source-mac-wildcard* indicates the source MAC range. • **interface** *interface-num*: Specifies the physical port number. It can be either the ingress port or the egress port. • **CPU**: Indicates that the data will be forwarded to the CPU. • **any**: Specifies any address which can be at ingress or egress directions. • **time-range** *name*: Specifies the time range in which the ACL rule takes effect.
Step 3	**no access-list** [*number* | **all**] **Example:** `Device(config)#no access-list 102`	(Optional) Removes the ACL specified.

Example Configuration for a number-based Layer 2 ACL

The following example defines a number-based Layer 2 ACL to disable the ARP packet whose source MAC address is 00: 00: 00: 00: 00: 01.

```
Device#configure terminal
Device(config)#access-list 200 deny arp ingress 00:00:00:00:00:01 0 egress any
```

Configuring a Named Layer 2 ACL

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device#configure terminal`	Enters the global configuration mode.
Step 2	**access-list link** *name* **Example:** `Device(config)#access-list link layer2acl`	Defines a Layer 2 ACL and enters the ACL configuration mode.
Step 3	{**permit** | **deny** } [*protocol*] [**cos** *vlan-pri*] **ingress**{ { [**inner-vid***vid*] [*start-vlan-id end-vlan-id*] [*source-mac-addr source-mac-wildcard*] [**interface** *interface-number*] } | **any** } **egress** { { [*dest-mac-addr dest-mac-wildcard*] [**interface** *interface-number* | **cpu**] } | **any** } [**time-range** *name*] **Example:** `Device(config-link-nacl-layer2acl)#deny arp ingress 00:00:00:00:00:01 0 egress any`	Defines a Layer 2 ACL. The following are the parameters of the command: • **permit | deny**: Use the keyword **permit** to allow access. Use **deny** not to allow access. • *protocol* : Specifies the type of protocol packet carried by the Ethernet frame. In hexadecimal notation, the range is 0 through FFFF. It is optional in case of ARP, IP, RARP protocols. • **cos**: Defines the priority of the VLAN tag. • **ingress** : Specifies the rule for the incoming packets at the ingress port. • **inner-vid**: Specifies the inner VLAN ID of a double-tagged packet. • *start-vlan-id end-vlan-id*: Specifies the range of VLANs.

	Command or Action	Purpose
		For a double-tagged packet, it is the VLAN ID of the outer tag. • *source-mac-addr source-mac-wildcard*: Specifies the source MAC address options. *source-mac-wildcard* indicates the source MAC range. • **interface** *interface-num*: Specifies the physical port number. It can be either the ingress port or the egress port. • **CPU**: Indicates that the data is forwarded to the CPU. • **any**: Specifies any address which can be at ingress or egress directions. • **time-range** *name*: Specifies the time range in which the ACL rule takes effect.
Step 4	**no access-list** [*name* | **all**] **Example:** `Device(config)#no access-list all`	(Optional) Removes the ACL specified.

Example Configuration for a number-based Layer 2 ACL

The following example defines a name-based Layer 2 ACL to disable the ARP packet whose source MAC address is 00: 00: 00: 00: 00: 02.

```
Device#configure terminal

Device(config)#access-list link layer2acl
Device(config-link-nacl-layer2acl)#deny arp ingress 00:00:00:00:00:02 0 egress any
```

Activating an Access Control List

You have to activate an ACL for it to be effective. ACL follows the rule "First Activation, First Served".

Before you begin

To activate an ACL, ensure that the ACL is defined.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device#configure terminal`	Enters the global configuration mode.
Step 2	**access-group** [**ip-group** *name* | *num*] [**link-group** *name* | *num*] [**subitem** *num*] **Example:** `Device(config)#access-group ip-group 1 subitem 1`	Activates the specified ACL.
Step 3	**no access-group** [**ip-group** *name* | *num*] [**link-group** *name* | *num*] [**subitem** *num*] **Example:** `Device(config)#no access-group ip-group 1 subitem 1`	Deactivates the specified ACL.
Step 4	**no access-group all** **Example:** `Device(config)#no access-group all`	Deactivates all the ACLs.

Configuration Examples for Activating an ACL

The following example configures an ACL numbered 1 and activates it.

```
Device#configure terminal
Device(config)#access-list 1 deny any

Device(config)#access-list 1 permit 1.1.1.1 0
Device(config)#show access-list config 1
Standard IP Access List 1, match-order is config, 2 rule:
  0  deny    any
permit  1.1.1.1  0.0.0.0


Device(config)#access-group ip-group 1 subitem 1

Device(config)#access-group ip-group 1 subitem 0
```

In this case, because first activation takes precedence, the device allows only those packets with a source IP address of 1.1.1.1 to pass through.

The following example configures a Standard ACL with match-order **auto** and activates it.

```
Device#configure terminal
Device(config)#access-list 1 match-order auto
Device(config)#access-list 1 deny any
Device(config)#access-list 1 permit 1.1.1.1 0
Device(config)#show access-list config 1
Standard IP Access List 1, match-order is auto, 2 rule:
0  permit  1.1.1.1  0.0.0.0
1  deny    any
```

```
Device(config)#access-group ip-group 1 subitem 0
Device(config)#access-group ip-group 1 subitem 1
```

In this case, because first activation takes precedence, the device allows only those packets that have a source IP address 1.1.1.1 to pass through.

The following example configures multiple ACLs and activates them to achieve IP, MAC, and port binding.

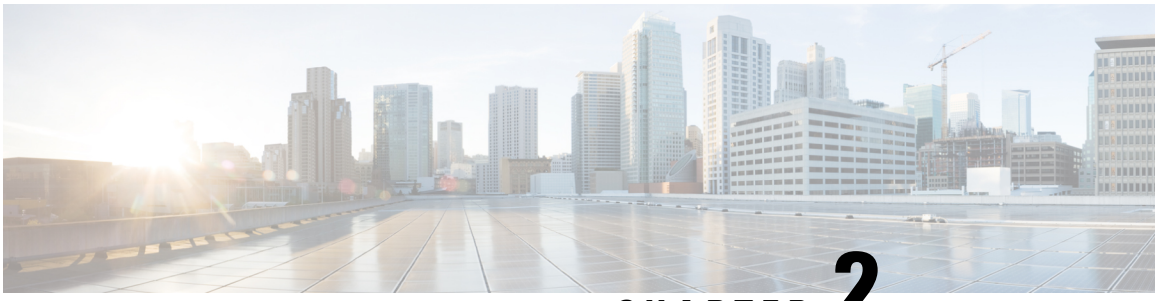
```
Device#configure terminal

Device(config)#access-list 1 permit 1.1.1.1 0
Device(config)#access-list 200 permit ingress 00:00:00:00:00:01 0 interface ethernet 0/1
egress any
Device(config)#access-group ip-group 1 link-group 200
```

Verifying Access Control List Configurations

After you have configured ACL for your devices, use the following commands to view the configurations.

Table 1: Show commands to Verify ACL Configurations

Command	Purpose
show access-list config statistic	Displays ACLstatistics. You can execute the command in any mode.
show access-list config { **all** | *num* | **name** *name* }	Displays the ACLs. You can execute the command in any mode.
show access-list runtime statistic	Displays the number of activated ACLs. You can execute the command in any mode.
show access-list runtime { **all** | *num* | **name** *name* }	Displays details of the activated ACLs. You can execute the command in any mode.



CHAPTER 2

Preventing DDOS Attack

About DDOS Attack

Denial of Service (DoS) attack does not allow the computer or network not provide normal services.

DoS attack is a simple and effective attack method which is very harmful to many network technologies. It attacks through various means to consume network bandwidth and system resources. It also attacks system defects, pralyzing the normal service of normal system. This results in the system not being able to service the users. It prevents the normal user from accessing services.

A distributed-denial-of-service (DDoS) is a form of security threat where a malicious host floods simultaneous data requests to a device or a central server. The host generates these requests from multiple compromised systems.

By flooding the device the attacker hopes to exhaust the internal RAM and affect the internet bandwidth thus disrupting the business.

You can modify the following IP packet settings to prevent a DDoS attack.

- Based on the relevant standard, the Time to Leave (TTL) field in the IP packet header must be greater than 0. By default, if a packet with TTL field equal to 0 is received, then the device discards the message as an attack. You can enable TTL monitoring to prevent DDoS attack.
- The number of fragments depend on the number of packets. If the number of packets is large, then the number of fragments is large and affects the performance of the system resources. Configuring a reasonable limit for the number of packets restricts the number of fragments. If the limit is exceeded, the message is discarded as an attack message. By default, an IP message has 800 fragments. You can limit the number of fragments allowed on a device to prevent a DDoS attack.

How to Prevent DDOS Attack

Enabling Time to Leave (TTL) Monitoring

To enable Time to Leave (TTL) monitoring, perform this procedure.

Note This procedure is optional.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	[**no**] **anti-dos ip ttl** **Example:** `Device(config)# anti-dos ip ttl`	Enables TTL monitoring. By default, messages with TTL with a value of 0 are discarded Use the **no anti-dos ip ttl** command to disable the anti-TTL attack. After configuration, normal messages are processed.
Step 4	**show anti-dos** **Example:** `Device(config)# show anti-dos`	(Optional) Displays the configuration information.

Configuring Limit for IP Fragmentation

To configure limit for IP fragmentation, perform this procedure.

Note This procedure is optional.

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	[**no**] **anti-dos ip fragment** *max-numbers* **Example:** `Device(config)#`	Allows IP fragmentations within the specified threshold value. Use the **no anti-dos ip fragment** command to restore the default value of 800.

Example: Preventing DDOS Attack

The following example shows how to verify the device can communicate with two fragments of the IP message.

```
Device> enable
Device# configure terminal
Device(config)# ping -l 2800 10.5.2.91
PING 10.5.2.91: with 2800 bytes of data:
reply from 10.5.2.91: bytes=2800 time<10ms TTL=64
reply from 10.5.2.91: bytes=2800 time<10ms TTL=64
reply from 10.5.2.91: bytes=2800 time<10ms TTL=64
reply from 10.5.2.91: bytes=2800 time<10ms TTL=64
reply from 10.5.2.91: bytes=2800 time<10ms TTL=64
----10.5.2.91 PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 0/0/0
```

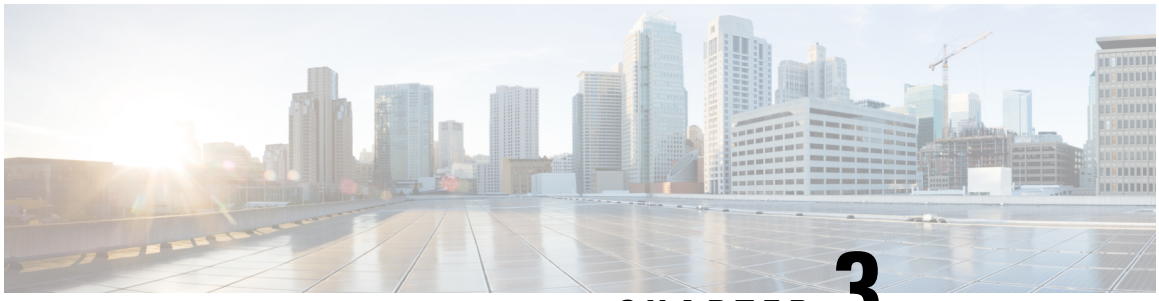
The following example shows how to verify the device unable to communicate with three fragments of the IP message.

```
Device> enable
Device# configure terminal
Device(config)# ping -l 3000 10.5.2.91
PING 10.5.2.91: with 3000 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
no answer from 10.5.2.91
```

The following example shows how to delete the IP fragmentation configuration and verify the device able to communicate with three fragments of the IP message.

```
Device> enable
Device# configure terminal
```

```
Device(config)# ping -l 3000 10.5.2.91
PING 10.5.2.91: with 3000 bytes of data:
reply from 10.5.2.91: bytes=3000 time=10ms TTL=64
reply from 10.5.2.91: bytes=3000 time<10ms TTL=64
reply from 10.5.2.91: bytes=3000 time=10ms TTL=64
reply from 10.5.2.91: bytes=3000 time<10ms TTL=64
reply from 10.5.2.91: bytes=3000 time<10ms TTL=64

----10.5.2.91 PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 0/4/10
```



CHAPTER 3

Configuring CPU CAR

About CPU Committed Access Rate (CAR)

Flooding the device with messages affects the device CPU performance. You can limit the rate of messages received on the device by configuring a limit for the CPU CAR.

The CPU CAR is enabled by default.

Note CPU CAR is not supported with the shutdown function

How to Configure CPU CAR

Configuring Limit for CPU CAR

To configure limit for the CPU CAR, perform this procedure.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. [**no**] **cpu-car** *rate*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	[**no**] **cpu-car** *rate* **Example:** `Device(config)#`	(Optional) Allows packets within the specified CPU-car rate. Use the **no cpu-car** command to restore the default value of 400pps.

Montoring CPU Performance

The commands in the following table can be used to monitor CPU performance

Table 2: CPU Performance

Command	Purpose
show cpu-car	Displays CPU-car performance.
show cpu-statistics [**ethernet** *port-number*]	Displays CPU receiving packet port statistics. Use the **clear cpu-statistics** command to clear the port statistics.
show cpu-classification [**interface ethernet***port-number*]	Displays CPU receiving packet classification statistics. Use the **clear cpu-classification** [**interface ethernet***port-number*] to clear the packet classification statistics.
show cpu-utilization	Displays CPU utilization.

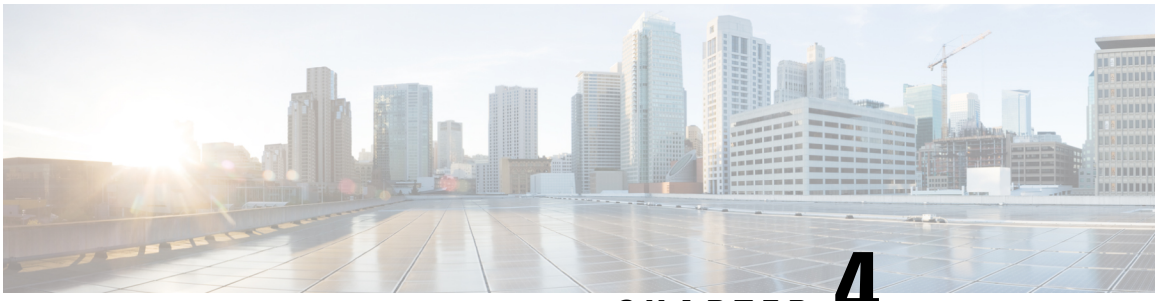
Example: Configuring CPU CAR function

The following example shows how to configure the CPU CAR speed to 50pps.

```
Device> enable
Device# configure terminal
Device(config)# interface range ethernet 1/1 ethernet 1/2
Device(config-if-ethernet-1/2)# port-car-rate 50
Device(config-if-ethernet-1/2)# exit
Device(config)# show cpu-car
Send packet to cpu rate = 50 pps.
```

Ixia A sends icmp request messages to the DUT: at a rate of 100 pps for 10 seconds, the total number of messages on the dut is 600, indicating that the cpu-car function takes effect.

```
Device> enable
Device# configure terminal
Device(config)# clear cpu-statistics
Device(config)# clear cpu-classification
Device(config)# clear interface
Device(config)# show cpu-statistics ethernet 1/2
Show packets sent to cpu  statistic information
port    64Byte  128Byte  256Byte  512Byte  1024Byte  2048Byte
e1/2  600     0        0        0        0         0
Device(config)# show cpu-classification
Type        Count       Percent(%)
Total       600         100
BPDU        0           0
ERRP        0           0
ARP         0           0
MLD         0           0
IGMP        0           0
ICMP        600         100
OSPF        0           0
RIP          0           0
DHCP        0           0
SNMP        0           0
Telnet        0           0
PIM          0           0
BGP         0           0
SSH         0           0
Other        0           0

Device(config)# show statistics interface ethernet 1/2
Port number  :  e1/2
last 5 minutes input rate 5248 bits/sec, 10 packets/sec
last 5 minutes output rate 433832 bits/sec, 771 packets/sec
64 byte packets:1048
65-127 byte packets:0
128-255 byte packets:0
256-511 byte packets:0
512-1023 byte packets:0
1024-1518 byte packets:0
1048 packets input, 67072 bytes , 0 discarded packets
1048 unicasts, 0 multicasts, 0 broadcasts
0 input errors, 0 FCS error, 0 symbol error, 0 false carrier
0 runts, 0 giants
19 packets output, 1216 bytes, 0 discarded packets
0 unicasts, 9 multicasts, 10 broadcasts
0 output errors, 0 deferred, 0 collisions
0 late collisions
Total entries: 1.
```



CHAPTER 4

Configuring Shutdown Control

About Shutdown Control

The shutdown control function allows you to configure a limit on the number of messages received on a port. If the threshold limit is exceeded, the port is shut down. This prevents bandwidth and device failure and protects other connected devices.

A shutdown port can be restored in the following ways.

- You can manually open the port using the **no shutdown** command.
- You can configure the automatic recovery mode and set a recovery time to restore the port.

How to Configure Shutdown Control

Enabling the Shutdown Control

To enable the shutdown control, perform this function.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface ethernet** *port-number* **Example:** `Device(config)#`	Enter the port configuration mode.
Step 4	[**no**] **shutdown-control** {**broadcast** | **multicast** | **unicast** } {*rate*} **Example:** `Device(config)#`	Enables shutdown control. • **broadcast** : The broadcast packets to be monitored. • **multicast** : The multicast packets to be monitored. • **unicast**: The unicast packets to be monitored. • *rate*: The threshold value of allowed number of packets Use the **no shutdown-control** {**broadcast** |**multicast** | **unicast**} command to disable the shutdown rate.

Configuring the Port Recovery Mode

To configure the port recovery mode, perform this procedure.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	[**no**] **shutdown-control-recover mode** {**automatic** | **manual**} **Example:** `Device(config)# shutdown-control-recover mode automatic`	Enables the configured recovery mode. • **automatic**: The mode is configured as automatic. • **manual**: The mode is configured as manual. Use the **no shutdown-control-recover** command to restore the default configuration.
Step 4	[**no**] **shutdown-control-recover automatic-open-time** *value* **Example:** `Device(config)#`	Restarts the port after the recovery time is expired. . Use the **no shutdown-control-recover** command to restore the default value of 480s.

Manually Restore a Shutdown Port

To manually restore a shutdown port, perform this procedure.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface ethernet** *port-number* **Example:** `Device(config)# interface ethernet 0/1`	Enters the port configuration mode.
Step 4	**no shutdown** **Example:** `Device(config-if-ethernet-0/1)# no shutdown`	Restores the shut down port. Use the **shutdown port** command to shutdown the port manually.

Monitoring Shutdown Control Configuration

The commands in the following table can be used to monitor shutdown control configurations.

Table 3: Monitoring Shutdown Control Cofiguration

Command	Purpose
show shutdown-control interface [**ethernet** *port-number*]	Displays the shutdown configuration.
show shutdown-control interface [**ethernet** *port-number*]	Displays the port recovery configuration.

Example: Configuring Shut Down Control

The following example shows how to enable the unknown unicast shut down control function and set the rate to 1000 pps

```
Device> enable
Device# configure terminal
Device(config)# interface ethernet 1/2
```

```
Device(config-if-ethernet-1/2)# shutdown-control unicast 1000
Device(config-if-ethernet-1/2)# exit
```

The following example shows how to view the configuration Information

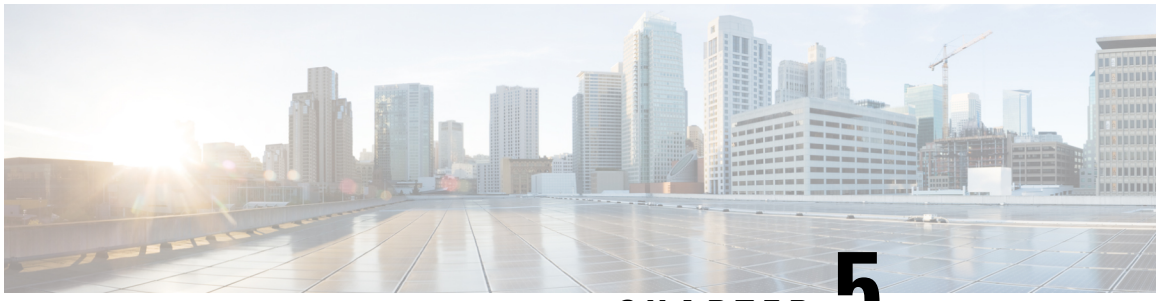
```
Device> enable
Device# configure terminal
Device(config)# show shutdown-control interface ethernet 1/2
port shutdown control recover mode : automatic
Port recover time(second)  : 480
port shutdown control information :
PortID   Broadcast Broadcast Multicast Multicast Unicast Unicast  RemainTime
status    value     status    value     status  value
e1/2   disable   -         disable   -         enable  1000     -
Total entries: 1 .
```

The following example shows how to view the configuration Information

```
Device> enable
Device# configure terminal
Device(config)# show shutdown-control interface ethernet 1/2
port shutdown control recover mode : automatic
Port recover time(second)  : 480
port shutdown control information :
PortID   Broadcast Broadcast  Multicast Multicast   Unicast Unicast    RemainTime
         status    value     status    value     status  value
e1/2   disable   -         disable   -         enable  1000     07min48sec
Total entries: 1 .
```

The following example shows how to view the configuration Information

```
Device> enable
Device# configure terminal
Device(config)# show interface brief ethernet 1/2
Port    Desc   Link  shutdn  Speed   Pri PVID  Mode TagVlan    UtVlan
e1/2     down  ERROR  auto    0   1    hyb            1
Total entries: 1 .
```



CHAPTER 5

DHCP Anti-Attack

DHCP Anti-Attack

Normally, when the DHCP client obtains an IP address from the DHCP server, the number of DHCP packets sent by the DHCP client is very small and doesn't affect the performance of the DHCP server. However, a malicious attack can cause the DHCP client to flood DHCP packets to the DHCP server, which will affect the DHCP server performance. To prevent this, you can enable DHCP monitoring on a device.

You can configure a DHCP rate threshold to monitor the packets reaching a device. If the packet rate is equal or higher than the threshold, then the packets are considered as an attack and discarded. The default packets rate threshold is 16pps.

When an attack is detected, the source MAC address of the attack packet is sent to the address table. The address table is maintained with an aging time. When the aging time expires, the table entry with the source MAC address is deleted and packets with the same source MAC address are dropped. The default aging time is 10 minutes. You can modify the aging time. Configure the aging time with a value of 0 prevents the table entry from being deleted.

By default, after an attack all ports are considered as not trustworthy. You can configure a port that does not require monitoring and is trustworthy as a trusted port.

How to Configure DHCP Anti-Attack

Enabling DHCP Packet Monitoring

To enable DHCP packet monitoring, perform this procedure.

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	[**no**] **dhcp anti-attack** **Example:** `Device(config)# dhcp anti-attack`	Enables DHCP packet monitoring. Use the **no anti-dhcp anti-attack** command to disable DHCP packet monitoring.

Configuring DHCP Rate Threshold

To configure DHCP rate threshold, perform this procedure.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	[**no**] **dhcp anti-attack threshold** *value* **Example:** `Device(config)#`	Configures the rate threshold for DHCP packets globally. Use the **no dhcp anti-attack threshold** command to set the default value of 16pps.
Step 4	**interface ethernet** *port-number* **Example:** `Device(config)#`	Enter the port configuration mode.
Step 5	[**no**] **dhcp anti-attack threshold** *value* **Example:** `Device(config-if)#`	(Optional) Configures the rate threshold for DHCP packets on an interface. Use the **no dhcp anti-attack threshold** command to set the default value of 16pps.

Configuring Recovery Function

To configure recovery function, perform this procedure.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**dhcp anti-attack recover-time** *value* **Example:** `Device(config)#`	(Optional) Configures the recovery time. The default is 10m. Configuring a value of 0 means no aging.
Step 4	**dhcp anti-attack recover** [**all** |*mac-address*] **Example:** `Device(config)#`	Configures the manual recovery. Restores the table items immediately without the need to wait for the aging time to expire.

Configuring Trusted Ports

To configure trusted ports, perform this procedure.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface ethernet** *port-number* **Example:** `Device(config)# interface ethernet 0/1`	Enter the port configuration mode.
Step 4	[**no**] **dhcp anti-attack trust** **Example:** `Device(config-ethernet-0/1)# dhcp anti-attack trust`	(Optional) Configures the port as a trusted port. Use the **no dhcp anti-attack trust** command to configure the port as not trusted.

Monitoring DHCP Anti-Attack

The commands in the following table can be used to monitor DHCP anti-attack.

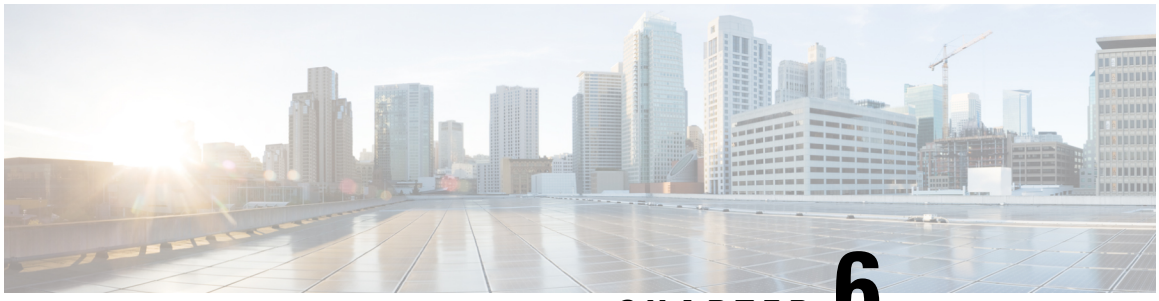
Table 4: Monitoring DHCP Anti-Attack

Command	Purpose
show dhcp anti-attack [**interface ethernet** *port-number*]	Displays the DHCP anti-attack configuration.
show dhcp anti-attack interface ethernet *port-number*	Displays the trusted port configuration.

Example: Anti-DHCP Attack

The following example shows how to configure the anti-DHCP attack.

```
Device> enable
Device# configure terminal
Device(config)# dhcp anti-attack
Device(config)# dhcp anti-attack action deny-dhcp
Device(config)# dhcp anti-attack threshold 1
Device(config)# dhcp anti-attack recover-time 3
Device(config)# logging monitor 0
Device(config)# debug dhcp
Device(config)# show dhcp anti-attack
Dhcp anti-attack: enabled
Dhcp rate limit:1pps
User recovery time:3 minutes
Reject type:DenyDHCP
DeniedSrcMAC       Port     Vlan    DenyType  RemainAgingTime(m)
00:00:00:01:11:23     e1/1     2      DenyDHCP  3

Total entry: 1.

#After 3 minutes, the attack entry is aged out

Device(config)# show dhcp anti-attack
Dhcp anti-attack: enabled
Dhcp rate limit:1pps
User recovery time:3 minutes
Reject type:DenyDHCP
DeniedSrcMAC       Port     Vlan  DenyType  RemainAgingTime(m)

Total entry: 0.
```



CHAPTER 6

Preventing ARP Spoofing and Flood Attack

Information About ARP Spoofing and Flood Attack

Overview of ARP Anti-Spoofing

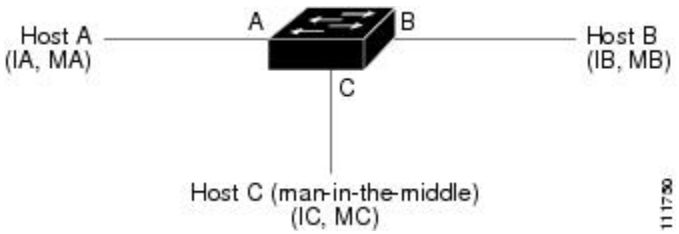
ARP provides IP communication within a broadcast domain by mapping an IP address to a MAC address. For example, host B wants to send information to host A but does not have the MAC address of host A in its ARP cache. In ARP terms, host B is the sender and host A is the target.

To get the MAC address of host A, host B generates a broadcast message for all hosts within the broadcast domain to obtain the MAC address associated with the IP address of host A. All hosts within the broadcast domain receive the ARP request, and host A responds with its MAC address.

ARP spoofing attacks occurs because ARP allows a reply from a host even if an ARP request was not received. After the attack, all traffic from the device under attack flows through the attacker's computer and then to the router, switch, or host.

An ARP spoofing attack can affect hosts, switches, and routers connected to your network by sending false information to the ARP caches of the devices connected to the subnet. Sending false information to an ARP cache is known as ARP cache poisoning. Spoof attacks can also intercept traffic intended for other hosts on the subnet.

Figure 1: ARP Spoofing



Hosts A, B, and C are connected to the device on interfaces A, B, and C, which are on the same subnet. Their IP and MAC addresses are shown in parentheses; for example, host A uses IP address IA and MAC address

MA. When host A needs to send IP data to host B, it broadcasts an ARP request for the MAC address associated with IP address IB. When the device and host B receive the ARP request, they populate their ARP caches with an ARP binding for a host with the IP address IA and a MAC address MA; for example, IP address IA is bound to MAC address MA. When host B responds, the device and host A populate their ARP caches with a binding for a host with the IP address IB and the MAC address MB.

Host C can poison the ARP caches of the device, host A, and host B by broadcasting two forged ARP responses with bindings: one for a host with an IP address of IA and a MAC address of MC and another for a host with the IP address of IB and a MAC address of MC. Host B and the device then use the MAC address MC as the destination MAC address for traffic intended for IA, which means that host C intercepts that traffic. Likewise, host A and the device use the MAC address MC as the destination MAC address for traffic intended for IB.

Because host C knows the true MAC addresses associated with IA and IB, it can forward the intercepted traffic to those hosts by using the correct MAC address as the destination. This topology, in which host C has inserted itself into the traffic stream from host A to host B, is an example of a man-in-the middle attack.

To prevent spoofing, you can enable ARP anti-spoofing. If ARP anti-spoofing is enabled, all ARP packets will be redirected to CPU for a check. The ARP packets will be verified with the entries in the static ARP table or the IP source guard static binding table or the DHCP snooping table. All ARP packets that match the entries in any one of the table will be transmitted. All incomplete ARP packets, or packets that partially match with any one of the table entries, will be discarded. Unkown ARP packets, or packets that do not match with any table entries, can be configured to either be discarded or flooded to all ports. ARP anti-spoofing attack is disabled by default.

You can configure the host protection feature to bind the IP address or MAC address and the connected port of the host together. ARP packets transmitted from this port is accepted by all other connected ports. ARP packets with the same IP address or MAC address is discarded if transmitted from any other port.

You can configure the source MAC address consistency inspection feature to check whether the ethernet source MAC address in the ARP packet is the same as the source MAC address stored in the table. If the source MAC addresses do not match, the packet is discarded. This feature is disabled by default.

A layer-3 device can configured as the gateway for certain LAN devices. An attacker host can try to add the Layer 3 device to the blocked list by sending a gratuitous ARP identfying itself as the correct gateway. You can configure the gateway anti-spoofing feature to prevent this kind of attack. This feature is disabled by default.

By default, after an attack all ports are considered as not trustworthy. You can configure a port that does not require monitoring and is trustworthy as a trusted port.

Overview of ARP Flooding Attack

An ARP spoofing attack can affect hosts, switches, and routers connected to your network by flooding packets to the CPU of the devices connected to the subnet and thus affecting device performance. Flooding the CPU on the device is known as ARP flooding attack.

To prevent ARP flood attack, the following configurations are available.

- You must enable ARP anti-flood attack to prevent ARP flood attack. The ARP packet is forwarded to the CPU. Each traffic flow is identified based on the source MAC address of the packet.
- You can configure a rate threshold to monitor the ARP flow. If the rate threshold is exceeded, then it is considered as an attack. You configure a rate threshold globally or for an interface.
- Once an attack occurs, you can configure whether to add the host's source MAC address to the blackhole address list and discard all packets, or discard only the ARP packets from the host.

- To remove hosts from the blackhole address list, you can either define a recovery time interval or manually restore the host.
- You can bind the dynamic MAC address to the static MAC address of a host in the blackhole address list. This prevents the host from transmitting any type of packets.

How to Prevent ARP Spoofing And Flood Attack

Enabling ARP Anti-Spoofing

To enable ARP anti-spoofing, perform this procedure.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. [**no**] **arp anti-spoofing**
4. **arp anti-spoofing unknown**{**discard**| **flood**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	[**no**] **arp anti-spoofing** **Example:** `Device(config)# arp anti-spoofing`	Enables ARP anti-spoofing. Use the **no** form of this command to disable ARP anti-spoofing.
Step 4	**arp anti-spoofing unknown**{**discard**| **flood**} **Example:** `Device(config)# arp anti-spoofing unknown discard`	Specifies whether to discard or flood unknown packets.

Configuring Host Protection

Configuring host protection on a port allows the port to discard unknown ARP packets.

Configure IP-port binding when you configure the device to discard the unknown ARP packets. This allows the ARP packet of this IP address to flood to the other ports only through this configured port. If the ARP packet of this IP address enters through another port, it will be discarded.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. [**no**] **host-guard bind ip** *ip-address* **interface** {**ethernet** | **gpon**} *slot_number/port_number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	[**no**] **host-guard bind ip** *ip-address* **interface** {**ethernet** | **gpon**} *slot_number/port_number* **Example:** `Device# host-guard bind ip 192.168.5.13 interface ethernet 1/2`	Configures host protection on the specified port. • *slot-number*: • GPON: The value is 0. • GE Ethernet: The value is 1. • 10GE Ethernet: The value is 2. • *port-number*: • GPON: The range is from 1 to 8. • GE Ethernet: The range is from 1 to 4. • 10GE Ethernet: The range is from 1 to 2. Use the **no** form of this command to delete host protection.

Configuring Source MAC Address Consistency Inspection

To configure source MAC address consistency inspection, perform this procedure.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. [**no**] **arp anti-spoofing valid-check**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	[**no**] **arp anti-spoofing valid-check** **Example:** `Device# arp anti-spoofing valid-check`	Enables source mac address consistency inspection. Use the **no** form of this command to disable this feature.

Configuring Gateway Anti-Spoofing

To configure gateway anti-spoofing, perform this procedure.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. [**no**] **arp anti-spoofing deny-disguiser**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	[**no**] **arp anti-spoofing deny-disguiser** **Example:** `Device# arp anti-spoofing deny-disguiser`	Enables gateway anti-spoofing. Use the **no** form of this command to disable gateway anti-spoofing.

Configuring Trust Port

To configure trust port, perform this procedur

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface ethernet** *port-number*
4. [**no**] **arp anti trust**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface ethernet** *port-number* **Example:** `Device(config)#`	Enter the port configuration mode.
Step 4	[**no**] **arp anti trust** **Example:** `Device(config)#`	(Optional) Configures the port as a trusted port. Use the **no arp anti trust** command to disable the feature.

Configuring Anti-Flood Attack

To configure anti-flood attack, perform this procedure.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. [**no**] **arp anti-flood**
4. **arp anti-flood threshold** *threshold_value*
5. **arp anti-flood action** {**deny-all** |**deny-arp**}
6. **arp anti-flood recover-time** *time*
7. **arp anti-flood recover** {*mac address* |**all**}
8. **interface ethernet** *port-number*
9. **arp anti-flood threshold** *threshold_value*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	[**no**] **arp anti-flood** **Example:** `Device(config)# arp anti-flood`	Enables anti-ARP flooding attack Use the **no** form of this command to disable this feature.
Step 4	**arp anti-flood threshold** *threshold_value* **Example:** `Device(config)# arp anti-flood`	Configure the ARP anti-flood threshold value. The default is 16pps.
Step 5	**arp anti-flood action** {**deny-all** |**deny-arp**} **Example:** `Device(config)# arp anti-flood action deny-arp`	(Optional) Specifies the type of packets to be discarded. • **deny-all**: Adds the host to a blackhole address list and discards all packets. • **deny-arp**: Discards only ARP packets
Step 6	**arp anti-flood recover-time** *time* **Example:** `Device(config)# arp anti-flood recover-time 100`	(Optional) Defines the recovery time interval after which a host is allowed to transmit again. The recovery interval is 0-1440 minutes. The default is 10 minutes. Configuring a time out interval of 0 requires the host to be manually restored.
Step 7	**arp anti-flood recover** {*mac address* |**all**} **Example:** `Device(config)# arp anti-flood recover 00:00:00:00:32:33`	(Optional) Manually restores the host to transmit again.
Step 8	**interface ethernet** *port-number* **Example:** `Device(config)#`	Enter the port configuration mode.
Step 9	**arp anti-flood threshold** *threshold_value* **Example:** `Device(config-if)# arp anti-flood`	Configure the ARP anti-flood threshold value. The default is 16pps.

Monitoring ARP Snooping and Flood Attack

The commands in the following table can be used to monitor ARP snooping and flood attack

Table 5: ARP Snooping and Flood Attack

Command	Purpose
show arp anti-snooping	Displays ARP anti-snooping configuration.
show arp anti-flood	Displays ARP anti-flood configuration and attackers list
show arp anti interface	Displays the state of interface

Example: Preventing ARP Spoofing and Flood Attack

Network Requirements

Consider a network scenario in which a switch is connected to a DHCP server and two client devices within the same VLAN. To enable anti-ARP spoofing in this scenario, enable DHCP snooping and set the port connecting the switch to the DHCP server as the trust port of DHCP snooping.

The following example shows how to enable DHCP snooping, set ethernet 1/1 as the trust port from DHCP snooping and bind the port IP to the ip-soure-guard binding table.

```
Device> enable
Device# configure terminal
Device(config)# dhcp-snooping
Device(config)# interface ethernet 1/1
Device(config-if-ethernet-1/1)# dhcp-snooping trust
Device(config-if-ethernet-1/1)# exit
Device(config)# ip-source-guard bind ip 192.168.5.10 mac 40:16:9f:f2:75:a8 interface ethernet
 1/3 vlan 1
```

The following example shows how to configure ARP anti-spoofing

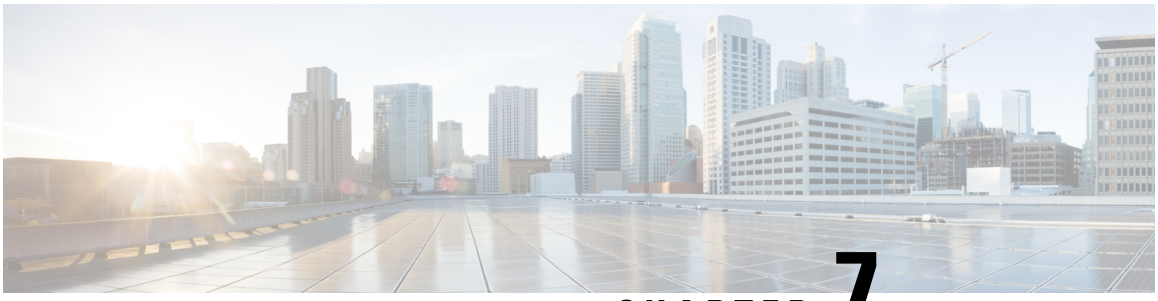
```
Device> enable
Device# configure terminal
Device(config)# arp anti-spoofing
Device(config)# arp anti-spoofing unknown discard
Device(config)# interface ethernet 1/1
Device(config-if-ethernet-1/1)# arp anti trust
```

Client A forwards arp quest message to dhcpserver, dhcpserver can be able to receive this arp quest message

Client B configure static ip=192.168.5.10 mac=40:16:9f:f2:75:a8, Client B forwards arp quest message to dhcpserver, dhcpserver can be able to receive this arp quest message

If client B enable anti-arp spoofing, source ip of arp message=Client A, the equipment will discard the message if it found this arp message is spoof message.

This instance estimates whether this arp message is spoof message or not according to dhcp-snooping clients table or ip-soure-guard bind table. In addition, ayer-3 equipment can be able to realize this function via static arp table. All of this shares the same principle, no more tautology here.



CHAPTER 7

Configuring 802.1x

Information About 802.1x

IEEE 802.1X is the accessing management protocol standard based on interface accessing control passed in June, 2001. Traditional LAN does not provide accessing authentication. User can access the devices and resources in LAN when connecting to the LAN, which is a safety loophole. For application of mobile office and CPN, device provider hopes to control and configure users connecting. There is also the need for accounting.

IEEE 802.1X is a network accessing control technology based on interface, which is the accessing devices authentication and control by physical accessing level of LAN devices. Physical accessing level here means the interface of LAN Switch devices. When authenticating, Switch is the in-between (agency) of client and authentication server. It obtains users identity from client of accessing Switch and verifies the information through authentication server. If the authentication passes, this user is allowed to access LAN resources or it will be refused.

802.1x Authentication

802.1X operates in the typical client/server model and defines three entities: supplicant system, authentication system, and authentication server system:

- Supplicant System: It is required to access the LAN, and enjoy the services provided by the Switch equipment (such as PC), the client needs to support EAPOL agreement, and the client must run the IEEE 802.1X authentication client software.
- Authentication System: In the Ethernet system, the authentication Switch is mainly used to upload and deliver user authentication information and control whether the port is available according to the authentication result. As if between the client and the authentication server to act as a proxy role.
- Authentication Server: Normally refers to the RADIUS server. RADIUS checks the identity of the client (user name and password) to determine whether the user has the right to use the network system to provide network services. After the end of the authentication, results will be sent to the Switch.

The above systems involve three basic concepts: PAE, controlled port, control direction:

- PAE: Port Access Entity (PAE) refers to the entity that performs the 802.1x algorithm and protocol operations.

  PAE is the entity responsible for performing algorithms and protocol operations in the authentication mechanism. The PAE uses the authentication server to authenticate the clients that need to access the LAN, and controls the authorized / unauthorized status of the controlled ports accordingly according to the authentication result. The client PAE responds to the authentication request from the device and sends the user authentication information to the device. The client PAE can also send the authentication request and the offline request to the device.

- Controlled port and uncontrolled port: An authenticator provides ports for supplicants to access the LAN. Each of the ports can be regarded as two logical ports: a controlled port and an uncontrolled port.
  - The uncontrolled port is always enabled in both the ingress and egress directions to allow EAPOL protocol frames to pass, guaranteeing that the supplicant can always send and receive authentication frames.
  - The controlled port is enabled to allow normal traffic to pass only when it is in the authorized state.
  - The controlled port and uncontrolled port are two parts of the same port. Any frames arriving at the port are visible to both of them.

- Control direction: In the non-authorized state, the controlled port is set to one-way controlled: the implementation of one-way controlled, prohibits the receiving frame from the client, but allows the client to send frames.
- Port controlled manner
  - Port-based authentication: As long as the first user authentication is successful under the physical ports, other access users without authentication can use the network source, when the first user is off line, other users will be refused to use network.
  - MAC-address-based authentication: All the users on the physical port need to be authenticated separately. When userA goes offline, only the userA cannot use the network.

802.1x Authentication Process

The 802.1x authentication system employs the Extensible Authentication Protocol (EAP) to exchange authentication information between the supplicant PAE, authenticator PAE, and authentication server.

At present, the EAP relay mode supports four authentication methods: EAP-MD5, EAP-TLS (Transport Layer Security), EAP-TTLS (Tunneled Transport Layer Security), and PEAP (Protected Extensible Authentication Protocol).

Switch supports EAP-Transfer mode and EAP-Finish mode to interactive with remote RADIUS server to finish the authentication.

- Authentication Process: The following takes EAP-Transfer authentication process for an example to introduce the basic service procedure.

  The authentication process is as follows:

  - When the user needs to access the network, it will input the registered user name and password through the 802.1X client and initiate the connection request (EAPOL-Start packet). At this point, the client program sends the request message to the device, start an authentication process.

- After receiving the requested data frame, the access device sends out a request frame (EAP-Request/Identity packet) to ask the user's client program for the user name.
- The client responds to the request from the device and sends the user name information to the device through the data frame (EAP-Response/Identity packet). The device encapsulates the RADIUS Access-Request packet and then sends it to the authentication server for processing after receiving the data frame packet from the client.
- After receiving the user name information from the device, the RADIUS server compares the information with the user name table in the database, finds the corresponding password information, and encrypts it with a randomly generated encryption key. And it sends the encrypted keyword to the device through a RADIUS Access-Challenge packet. The message is then forwarded by the device to the client.
- After receiving the EAP-Request/MD5 Challenge packet, the client encrypts the encrypted part (this encryption algorithm is usually irreversible) and generates the EAP-Response/MD5 Challenge packets and pass the authentication packets to the authentication server.
- The RADIUS server compares the received encrypted information (RADIUS Access-Request packet) with the local encrypted password information. If the password is the same, the RADIUS server considers the user to be a valid user and sends out the message-Accept and EAP-Success).
- After receiving the authentication message, the device changes the port to the authorized state, allowing the user to access the network through the port.

- EAP-Finsh: In this way, EAP packets are terminated at the device end and are mapped to RADIUS packets. The RADIUS server uses the standard RADIUS protocol to complete authentication, authorization, and accounting. The PAP or CHAP authentication method can be adopted between the device and the RADIUS server. Our Switch defaults to this mode. The following takes the CHAP authentication method as an example to describe the basic service flow, as shown below:

  The EAP termination mode differs from the authentication process of EAP relay mode in that a random encryption key for encrypting the user's password information is generated by the device, and then the device encrypts the user name, the random encryption key, and the encrypted password information of the client to the RADIUS server, and perform the related authentication process.

How to Configure 802.1x

This section provides information about how to configure 802.1x.

Configuring EAP

The 802.1x standard forwards the 802.1X authentication packets (Encapsulated with EAP frames) from the user to the RADIUS server without any processing. However, the traditional RADIUS server does not support the EAP feature. Therefore, the system supports the conversion of the authentication packets sent by the user to the data frames encapsulated by the standard RADIUS protocol and then forwards the packets to the RADIUS server.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **dot1x** {**eap-finish** | **eap-transfer**}
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	(Optional) Enters global configuration mode.
Step 3	**dot1x** {**eap-finish**	**eap-transfer**} **Example:** `Device(config)# dot1x eap-transfer`	Sets the protocol interaction mode between the system and the RADIUS server.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Enabling 802.1x

802.1x provides a user identity authentication scheme. However, 802.1x cannot implement the authentication scheme solely by itself. RADIUS or local authentication must be configured to work with 802.1x.

After enabling the 802.1X, the users who connected to the system can access to the LAN resources only after it had passed the authentication. When enabling the 802.1X, you should point out the whether the enabling way is based on interface authentication or MAC address authentication. The interface which does not participate in 802.1X authentication has no need to enable 802.1X authentication.

Interface configuration based on interface authentication: if one of the users under the port had passed the authentication, other users can use the network resources without authentication; However, if that user who had passed the authentication logoff, other users can not be able to use the network resources.

Interface configuration based on MAC address authentication: each user under the port should perform separate authentication. Only the user who had passed the authentication can he use the network resources. If a certain user logoff, it cannot affect other authenticated users to use the network resources.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dot1x method** {**macbased** | **portbased**} [**interface** *interface-type*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	(Optional) Enters global configuration mode.
Step 3	**dot1x method** {**macbased** | **portbased**} [**interface** *interface-type*] **Example:** `Device(config)# dot1x method macbased`	Enables 802.1x.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring 802.1x Parameters for a Port

After the interface enables the 802.1X authentication, this port needs to be authenticated by default while the uplink interface and the interface which connects to the server do not need, so you can configure the ports which do not need to be authenticated to be forceauthorized or disable their authentication functions. In addition, the interface which is banned to perform 802.1X authentication can be configured to be forceunauthorized.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dot1x port-control** {**auto** | **forceauthorized** | **forceunauthorized**} [**interface** *interface-type*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	(Optional) Enters global configuration mode.

	Command or Action	Purpose
Step 3	**dot1x port-control** {**auto** | **forceauthorized** | **forceunauthorized**} [**interface** *interface-type*] **Example:** `Device(config)# dot1x port-control forceauthorized`	Configures 802.1x parameters for a port.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring Re-authentication

In EAP-FINISH way, the port supports re-authentication. After the user is authenticated, the port can be configured to immediately re-certification, or periodic re-authentication.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dot1x** {**re-authenticate** | **re-authentication** | **timeout re-authperiod** *time*} [**interface** *interface-type*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	(Optional) Enters global configuration mode.
Step 3	**dot1x** {**re-authenticate** | **re-authentication** | **timeout re-authperiod** *time*} [**interface** *interface-type*] **Example:** `Device(config)# dot1x re-authenticate`	Enables reauthentication. • **re-authenticate**: Re-authenticates immediately. • **re-authentication**: Enables periodic re-authentication on a port. • **timeout re-authperiod** *time*: Enables periodic time configuration re-authentication on a port. The range is 1 to 3600.
Step 4	**end** **Example:**	Exits global configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config)# end`	

Configuring Watch Feature

After enabling this function, a port sends a 1x watch message periodically when no user is present, triggering the users to perform 802.1x authentication.

This triggering method is used to support clients that cannot send EAPOL-Start packets, such as 802.1X clients. Our device sends an EAP-Request/Identity packet to the client every N seconds to trigger authentication.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dot1x daemon** [**interface** *interface-type* | **time** *time*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	(Optional) Enters global configuration mode.
Step 3	**dot1x daemon** [**interface** *interface-type*	**time** *time*] **Example:** `Device(config)# dot1x daemon`	Enables the watch function. The **time** keyword configures the forwarding interval of watch packet.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring User Features

The operations mainly perform the operations, for example, the configurations for number of port users, delete users, heartbeat detection operations, etc.

Heartbeat detection: After this function is enabled, the device periodically forwards EAP-Request/Identity to the client ports, the normal online client responds with the EAP-Rsponse/Identity. If the four consecutive EAP-Request/Identity packets are not received the EAP-Rsponse/Identity packet from the client, the device considers the user to go offline, and then it will delete the session and change the port to an unauthorized state.

Quiesce function: After the user authentication fails, the device needs to quiesce for a period of time (The time can be configured through dot1x quiet-period-value. By default, no quiesced is required). During the quiesced period, the authenticator does not process the authentication request.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dot1x max-user** *number*
4. **dot1x user cut** {**mac-address** *address* | **username** *name*}
5. **dot1x detect** [**interface** *interface-type* | **interval** *time*]
6. **dot1x quiet-period-value** *time*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	(Optional) Enters global configuration mode.
Step 3	**dot1x max-user** *number* **Example:** `Device(config)# dot1x max-user 33`	Configures the maximum number of users that can pass authentication. The range is 1 to 100.
Step 4	**dot1x user cut** {**mac-address** *address* | **username** *name*} **Example:** `Device(config)# dot1x user cut username user12`	Deletes the specified online user.
Step 5	**dot1x detect** [**interface** *interface-type* | **interval** *time*] **Example:** `Device(config)# dot1x max-user 33`	Configures the Heartbeat detection time. The **interval** *time* range is 1 to 3600 seconds. The default is 25 seconds.
Step 6	**dot1x quiet-period-value** *time* **Example:** `Device(config)# dot1x max-user 33`	Configures the quiesce function. The range is 0 to 600 seconds. The default is 0 second.
Step 7	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring Host Mode Based on Port Authentication Mode

The host mode configuration only takes effect in port authentication method, please configure the port as port-based authentication; if the configuration of the host mode is the single-host, configure the port to be mac-based authentication, host mode will automatically become invalid.

- multi-hosts: Multi-hosts mode, when a user authentication is passed on the port, other users of the port can access network without authentication.
- single-host: Single-host mode, the user access network which the port allows only one authentication to pass and other users cannot access to the network, also can't go through authentication.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dot1x portbased host-mode** {**multi-hosts** | **single-host**} [**interface** *type*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	(Optional) Enters global configuration mode.
Step 3	**dot1x portbased host-mode** {**multi-hosts**	**single-host**} [**interface** *type*] **Example:** `Device(config)# dot1x portbased host-mode`	Configures host-mode based on port authentication mode.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring Guest VLAN

After enabling 1X authentication, the user can access only the network resources of the VLAN when the guest VLAN is configured on the port. Once the user authentication succeeds, the port automatically reverts to the previously configured VLAN. If the authentication server delivers a valid VLAN, the port is automatically added to the assigned VLAN. After the user goes offline, the port reverts to the guest VLAN.

To ensure that all functions can be used normally, please assign different VLAN IDs for the Config VLAN, the radius distribution VLAN, and the Guest VLAN.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dot1x guest-vlan** *vlan-id* [**interface** *type*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	(Optional) Enters global configuration mode.
Step 3	**dot1x guest-vlan** *vlan-id* [**interface** *type*] **Example:** `Device(config)# dot1x guest-vlan 120`	Configures the guest VLAN. The range is 1 to 4094.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring Radius VLAN

When 802.1X user pass the authentication via radius server, the server transmits the authentication information to the device. If the device has enabled radius function and the server has configured to distribute VLAN (adopting Tunnel-Pvt-Group-ID (81) attribute), the authentication information includes the distributed VLAN information as a consequence, what is more, the device adds the user authentication online interface to radius distributed VLAN.

Note Before using the radius VLAN distribution function, you should create the corresponding VLAN and then add the user interface to the corresponding VLAN, so does Guest VLAN and Default-active-vlan.

Radius distributes VLAN, but it does not change the interface original VLAN configuration, so does Guest VLAN and Default-active-vlan.

As to the interface-based authentication and the MAC-based authentication, radius vlan , Guest VLAN and Default-active-vlan are effective.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **aaa**
4. **radius vlan enable**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	(Optional) Enters global configuration mode.
Step 3	**aaa** **Example:** `Device(config)# aaa`	Enters Authentication, Authorization, and Accounting (AAA) configuration mode.
Step 4	**radius vlan enable** **Example:** `Device(config-aaa)# radius vlan enable`	Enables radius VLAN distribution function. This is disabled by default.
Step 5	**end** **Example:** `Device(config-aaa)# end`	Exits AAA configuration mode and returns to privileged EXEC mode.

Configuring EAPOL Transmission

When a port disables 802.1x authentication, it requires to transmit user 802.1x EAPOL message. So the equipment works as the relay, users can perform 802.1x authentication in the upper equipment. This function can only handle EAPOL packet forwarded to CPU. For packets that do not forward to CPU, the packets are processed by the hardware and are not subject to this configuration. You can configure EAPOL transparent transmission port and the corresponding uplink port only when the 802.1x authentication is disabled. That is, you cannot configure transparent transmission function when the 802.1x authentication is enabled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dot1x eapol-relay** [**interface** *type* | **uplink**]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	(Optional) Enters global configuration mode.
Step 3	**dot1x eapol-relay** [**interface** *type* | **uplink**] **Example:** `Device(config)# dot1x eapol-relay`	Enables port EAPOL message transmission function.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Monitoring 802.1x

This section provides the list of commands that are used to monitor the 802.1x configuration. Run the commands in the following table in global configuration mode:

Command	Purpose
show dot1x	Displays the status of 802.1x authentication function.
show dot1x daemon	Displays the configuration of 802.1x authentication interface watch function .
show dot1x interface	Displays interface configuration, such as the interface control mode, re-authentication state, the maximum number of users for the interface authentication.
show dot1x session	Displays 802.1x session.
show dot1x eapol-relay	Displays EAPOL pass through configuration.
show dot1x detect	Displays heartbeat detection configuration.
show dot1x guest-vlan	Displays guest VLAN information.
show dot1x port-auth	Displays whether the interface authentication is enabled or disabled.
show dot1x quiet-period-value	Displays the quiet period.
debug dot1x	Debugs dot1x receive packet and transmit packet as well as module processing.

Configuration Examples for 802.1x

This section provides configuration examples for 802.1x.

The following example shows how to enable the 802.1x authentication of Ethernet port 1/1.

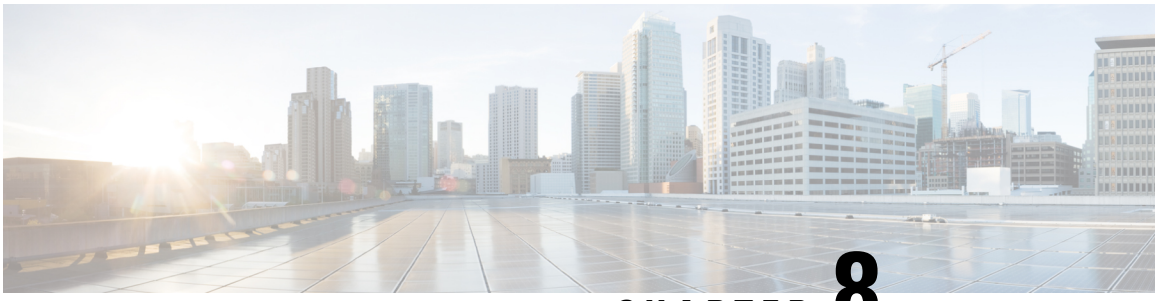
```
Device> enable
Device# configure terminal
Device(config)# dot1x method macbased interface ethernet 1/1
```

This example shows how to configure the basic function of RADIUS server.

```
Device> enable
Device# configure terminal
Device(config)# aaa
Device(config-aaa)# radius host 1
Device(config-aaa-radius-1)# primary-auth-ip 10.1.1.1 1812
Device(config-aaa-radius-1)# primary-acct-ip 10.1.1.2 1813
Device(config-aaa-radius-1)# auth-secret-key 123456
Device(config-aaa-radius-1)# acct-secret-key 123456
Device(config-aaa-radius-1)# exit
Device(config-aaa)# domain abc.com
Device(config-aaa-domain-abc.com)# radius host binding 1
Device(config-aaa-domain-abc.com)# state active
Device(config-aaa)# default domain-name enable abc.com
```

The following is a sample output of the **show dot1x session** command.

```
Device(config)# show dot1x session

port    vid    mac             username      login time
1/1     1      2001:DB8::1     u1@abc.com    2000/01/01 05:13:42
```



CHAPTER 8

Configuring RADIUS

Information About RADIUS

AAA Overview

AAA stands for Authentication, Authorization and Accounting.

AAA is actually a management of network security. Here, the network security mainly refers to the access control, including the users who can access the network server; what services are available to users with access rights; and how users are using network resources for billing.

AAA generally adopts the client/server structure: the client runs on the managed resource side, and the server stores the user information centrally. Therefore, the AAA framework has good scalability, and easy to achieve the centralized management of user information.

AAA Realization

There are two ways to realize AAA:

- via NAS.
- via RADIUS, TACACS +, etc.

RADIUS Overview

RADIUS creates a unique user database, stores the user name and password of the user to authenticate, and stores the service type and corresponding configuration information that is passed to the user to complete the authorization. After the user is authorized, the RADIUS server performs the function of accounting for user accounts.

RADIUS stands for Remote Authentication Dial in User Service.

- RADIUS is an AAA protocol for applications such as Network Access or IP Mobility.
- It works in both situations, Local and Mobile.
- It uses Password Authentication Protocol (PAP), Challenge Handshake Authentication Protocol (CHAP), or Extensible Authentication Protocol (EAP) protocols to authenticate users.
- It looks in text file, LDAP Servers, Database for authentication.
- After authentication services parameters passed back to NAS.
- It notifies when a session starts and stop. This data is used for Billing or Statistics purposes.
- SNMP is used for remote monitoring.
- It can be used as a proxy.

Here is a list of all the key features of Radius:

- Client/Server Model
  - NAS works as a client for the Radius server.
  - Radius server is responsible for getting user connection requests, authenticating the user, and then returning all the configuration information necessary for the client to deliver service to the user.
  - A Radius server can act as a proxy client to other Radius servers.
- Network Security
  - Transactions between a client and a server are authenticated through the use of a shared key. This key is never sent over the network.
  - Password is encrypted before sending it over the network.
- Flexible Authentication Mechanisms
  - Point-to-Point Protocol (PPP)
  - Password Authentication Protocol (PAP)
  - Challenge Handshake Authentication Protocol (CHAP)
  - Simple UNIX Login
- Extensible Protocol
  - Radius is extensible; most vendors of Radius hardware and software implement their own dialects.

How to Configure RADIUS

The following sections provide information about configuring RADIUS:

Configuring RADIUS Server

RADIUS server saves valid user's identity. When authentication, system transfers user's identity to RADIUS server and transfers the validation to user. User accessing to system can access LAN resources only after authentication of RADIUS server.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa**
4. **radius host** *name*
5. **primary-auth-ip** *ip-address port*
6. **second-auth-ip** *ip-address port*
7. **primary-acct-ip** *ip-address port*
8. **second-acct-ip** *ip-address port*
9. **auth-secret-key** *keystring*
10. **acct-secret-key** *keystring*
11. **nas-ipaddress** *ip-address*
12. **username-format** {**with-domain** | **without-domain**}
13. **realtime-account**
14. **realtime-account interval** *time*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	(Optional) Enters global configuration mode.
Step 3	**aaa** **Example:** `Device(config)# aaa`	Enters AAA configuration mode.
Step 4	**radius host** *name* **Example:** `Device(config-aaa)# radius host mmm`	Creates a RADIUS scheme and enters RADIUS scheme mode.
Step 5	**primary-auth-ip** *ip-address port* **Example:** `Device(config-aaa-radius-mmm)# primary-auth-ip 10.0.0.10 300`	Configures primary RADIUS. The authentication port range is 1 to 65535.

	Command or Action	Purpose
Step 6	**second-auth-ip** *ip-address port* **Example:** `Device(config-aaa-radius-mmm)# primary-auth-ip 10.0.0.11 400`	(Optional) Configures second RADIUS. The authentication port range is 1 to 65535.
Step 7	**primary-acct-ip** *ip-address port* **Example:** `Device(config-aaa-radius-mmm)# primary-acct-ip 10.1.1.10 333`	(Optional) Configures primary accounting server. The accounting port range is 1 to 65535.
Step 8	**second-acct-ip** *ip-address port* **Example:** `Device(config-aaa-radius-mmm)# primary-acct-ip 10.1.1.11 444`	(Optional) Configures second accounting server. The accounting port range is 1 to 65535.
Step 9	**auth-secret-key** *keystring* **Example:** `Device(config-aaa-radius-mmm)# auth-secret-key key1`	Configures the shared key of primary RADIUS.
Step 10	**acct-secret-key** *keystring* **Example:** `Device(config-aaa-radius-mmm)# acct-secret-key key2`	(Optional) Configures the shared key of second RADIUS.
Step 11	**nas-ipaddress** *ip-address* **Example:** `Device(config-aaa-radius-mmm)# nas-ipaddress 10.1.0.10`	(Optional) Configures the NAS-RADIUS address.
Step 12	**username-format** {**with-domain** | **without-domain**} **Example:** `Device(config-aaa-radius-mmm)# username-format with-domain`	(Optional) Specifies whether the user name is to be carried with the domain name when the system passes the packet to the current RADIUS server.
Step 13	**realtime-account** **Example:** `Device(config-aaa-radius-mmm)# realtime-account`	(Optional) Configures the realtime accounting.
Step 14	**realtime-account interval** *time* **Example:** `Device(config-aaa-radius-mmm)# realtime-account interval 20`	(Optional) Configures the realtime accounting time interval in minutes. The range is 1 to 255.

Configuring Radius Master Server and Radius Slave Server Shift

RADIUS offers master/slave server redundancy function, that is, if both the master server and slave server can be able to perform the regular work, it can only perform the authentication via master server; if there is something wrong with the master server, the slave server will be enabled; if the master server recovers normal again, the slave server will be disabled, and then the master server will be enabled.

Realization Mechanisms: When in radius authentication, if the master server cannot perform the regular work, just configure the master server as down, then the slave server will begin to work; if the master server is found had recovered the regular work, preemption timer will be enabled (time is configured as preemption-time). When the timer timeout, the master server will be configured as up, that is to say, you can perform the authentication operations via master server.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa**
4. **radius host** *name*
5. **preemption-time** *preemption-time*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	(Optional) Enters global configuration mode.
Step 3	**aaa** **Example:** `Device(config)# aaa`	Enters AAA configuration mode.
Step 4	**radius host** *name* **Example:** `Device(config-aaa)# radius host mmm`	Creates a RADIUS scheme and enters RADIUS scheme configuration mode.
Step 5	**preemption-time** *preemption-time* **Example:** `Device(config-aaa-radius-mmm)# primary-auth-ip 10.0.0.10 300`	Configures the preemption timer in minutes. The range is 0 to 1440, and the default value is 0.
Step 6	**end** **Example:**	Exits RADIUS scheme configuration mode and enters privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-aaa-radius-mmm)# end`	

Configuring Local User

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa**
4. **local-user username** *name* **password** *pwd* [**vlan** *vlan-id*]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	(Optional) Enters global configuration mode.
Step 3	**aaa** **Example:** `Device(config)# aaa`	Enters AAA configuration mode.
Step 4	**local-user username** *name* **password** *pwd* [**vlan** *vlan-id*] **Example:** `Device(config-aaa)# local-user username name1 password pass1 vlan 220`	Configures a local user. The VLAN ID range is 1 to 4094.
Step 5	**end** **Example:** `Device(config-aaa)# end`	Exits AAA configuration mode and enters privileged EXEC mode.

Configuring Domain

A username and password must be provided during authentication. Username usually contains the corresponding user ISP information, domain and ISP. The most important information of the domain is the RADIUS server authentication and accounting for the users in the domain.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa**
4. **default domain-name enable** *domain-name*
5. **domain** *name*
6. **scheme radius**
7. **scheme local**
8. **scheme radius local**
9. **radius host binding** *name*
10. **access-limit enable** *number*
11. **state active**
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	(Optional) Enters global configuration mode.
Step 3	**aaa** **Example:** `Device(config)# aaa`	Enters AAA configuration mode.
Step 4	**default domain-name enable** *domain-name* **Example:** `Device(config-aaa)# default domain-name enable domain1`	(Optional) Configures the default domain name.
Step 5	**domain** *name* **Example:** `Device(config-aaa)# domain eee`	Creates and enters a domain scenario.
Step 6	**scheme radius** **Example:** `Device(config-aaa-eee)# scheme radius`	(Optional) Configures to use radius server authentication.
Step 7	**scheme local** **Example:** `Device(config-aaa-eee)# scheme local`	Configures to use local user authentication.

	Command or Action	Purpose
Step 8	**scheme radius local** **Example:** `Device(config-aaa-eee)# scheme radius local`	Configures to use local authentication if the radius authentication fails.
Step 9	**radius host binding** *name* **Example:** `Device(config-aaa-eee)# radius host binding radius1`	(Optional) Selects the RADIUS server for the current domain.
Step 10	**access-limit enable** *number* **Example:** `Device(config-aaa-eee)# access-limit enable 30`	(Optional) Enables the number limit of authentication users in the domain and sets the number limit of allowed users. The range is from 1 to 640.
Step 11	**state active** **Example:** `Device(config-aaa-eee)# state active`	Activates the current domain.
Step 12	**end** **Example:** `Device(config-aaa-eee)# end`	Exits domain scenario mode and enters privileged EXEC mode.

Configuring RADIUS Features

- accounting-on: After the device reboots, it sends an Accounting-On packet to the RADIUS server to notify the RADIUS server to force the user of the device to go offline.
- RADIUS distributes port priority: After this function is enabled, if the user authenticates, the priority of the port where the user is located is modified. This function is carried out through the 77 attribute number in the Vendor Specific by default, which can be modified by using the radius config-attribute.
- RADIUS distributes port PVID: After this function is enabled, if the user passes the authentication, the PVID of the port where the user is located will be modified. This function is carried out by using the tunnel-Pvt-Group-ID. The value of this attribute is a string. Use this string to find the VLAN name descriptor that matches the VLAN value.
- RADIUS distributes number limit of MAC address: After this function is enabled, if the user passes the authentication, the MAC address learning limit of the port where the user resides is modified. This function is carried out through the 50 attribute number in the Vendor Specific by default, which can be modified by using the radius config-attribute.
- RADIUS distributes bandwidth control: After this function is enabled, if the user passes the authentication, the bandwidth control of the port where the user is located will be modified. The uplink bandwidth control is carried out through the 75 attribute number in the Vendor Specific by default, which can be modified by using theradius config-attribute; the downlink bandwidth control is carried out through the 76 attribute number in the Vendor Specific by default, which can be modified by using the radius config-attribute. The unit value defaults to kbps and can be modified through the radius config-attribute access-bandwidth unit.

- RADIUS distributes ACL: This function has no control commands. It is enabled by default. Configure via 11 attributes of Filter-Id.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa**
4. **accounting-on** {**enable** *sen-num* | **disable**}
5. **radius accounting**
6. **radius server-disconnect drop 1x**
7. **radius 8021p enable**
8. **radius vlan enable**
9. **radius mac-address-number enable**
10. **radius bandwidth-limit enable**
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	(Optional) Enters global configuration mode.
Step 3	**aaa** **Example:** `Device(config)# aaa`	Enters AAA configuration mode.
Step 4	**accounting-on** {**enable** *sen-num* | **disable**} **Example:** `Device(config-aaa)# accounting-on enable 40`	(Optional) Configures accounting-on function.
Step 5	**radius accounting** **Example:** `Device(config-aaa)# radius accounting`	(Optional) Enables accounting function .
Step 6	**radius server-disconnect drop 1x** **Example:** `Device(config-aaa)# radius server-disconnect drop 1x`	(Optional) If the accounting packet does not respond, the user is shut down.

	Command or Action	Purpose
Step 7	**radius 8021p enable** **Example:** `Device(config-aaa)# radius 8021p enable`	(Optional) Configures RADIUS to distribute port priority.
Step 8	**radius vlan enable** **Example:** `Device(config-aaa)# radius vlan enable`	(Optional) Configures RADIUS to distribute port PVID.
Step 9	**radius mac-address-number enable** **Example:** `Device(config-aaa)# radius mac-address-number enable`	(Optional) Configures RADIUS to distribute number limit of MAC address.
Step 10	**radius bandwidth-limit enable** **Example:** `Device(config-aaa)# radius bandwidth-limit enable`	(Optional) Configures RADIUS to distribute bandwidth control.
Step 11	**end** **Example:** `Device(config-aaa)# end`	Exits AAA configuration mode and enters privileged EXEC mode.

Monitoring RADIUS

Command	Purpose
show radius attribute	Displays the radius attribute.
show radius config-attribute	Displays the radius attribute.
show radius host	Displays the radius service configuration information.
debug radius	Enables the radius debugging function.

Example: Configuring RADIUS

This example shows how to configure the related services of RADIUS, and configure ACLs.

```
Device> enable
Device# configure terminal
Device(config)# aaa
Device(config-aaa)# radius host ngn
Device(config-aaa-radius-ngn)# primary-auth-ip 10.1.1.1 1812
Device(config-aaa-radius-ngn)# primary-acct-ip 10.1.1.2 1813
Device(config-aaa-radius-ngn)# auth-secret-key 123456
Device(config-aaa-radius-ngn)# acct-secret-key 123456
Device(config-aaa-radius-ngn)# exit
```

```
Device(config-aaa)# domain ngn.com
Device(config-aaa)# domain ngn.com
Device(config-aaa-domain-ngn.com)# radius host binding ngn
Device(config-aaa-domain-ngn.com)# state active
Device(config-aaa-domain-ngn.com)# exit
Device(config-aaa)# default domain-name enable ngn.com
Device(config-aaa)# exit
Device(config)# access-list 100 deny any 10.0.0.10 0.0.0.255
Device(config)# access-list 100 permit any any
```

After authentication succeeds, the user can access the external network normally. The information of the online users can be found on the device. The command of **show dot1x radius-acl** displays the status of the acl100 as enable, and the bandwidth of the ingress direction of the 0/ 0/1 port is limited to 2048 while the egress direction is limited to 1024.

The following is a sample output of the **show dot1x session** command.

```
Device(config)# show dot1x session

port    vid   mac                username       login time
e1/1    1     c8:3a:35:d3:e3:99  test@ngn.com   2000/12/11 15:07:00
Total [1] item(s).
```

The following is a sample output of the **show dot1x radius-acl** command.

```
Device(config)# show dot1x radius-acl

The format of radius acl is string.
The prefix of radius acl is assignacl-.
Port    acl  Status
e1/1  100  enable
Total entries: 1.
```

The following is a sample output of the **show bandwidth-control interface ethernet** command.

```
Device(config)# show bandwidth-control interface ethernet 1/1

port    Ingress bandwidth control  Egress bandwidth control
e1/1    2048 kbps                  1024 kbps
Total entries: 1.
```