ZXR10 8900 Series
10G Routing Switch
User Manual (FW Volume)

Version 2.8.02.C

ZTE CORPORATION
ZTE Plaza, Keji Road South,
Hi-Tech Industrial Park,
Nanshan District, Shenzhen,
P. R. China
518057
Tel: (86) 755 26771900 800-9830-9830
Fax: (86) 755 26772236
URL: http://support.zte.com.cn
E-mail: doc@zte.com.cn

Revision History

Revision No.	Revision Date	Revision Reason
R1.0	20090630	First Release

Serial Number: sjzl20093843

Contents

About This Manual

This manual is ZXR10 8900 Series (V2.8.02.C) 10G Routing Switch User Manual (FW Volume) and applies to ZXR10 8902/8905/8908/8912 10G routing switch (V2.8.02.C).

ZXR10 8900 series 10G routing switch has the following related manuals:

Manual	Summary
ZXR10 8900 Series (V2.8.02.C) 10G Routing Switch Hardware Installation Manual	This manual describes installation preparation, 19-inch cabinet installation, main device installation, power cable connection, cable connection and hardware inspection.
ZXR10 8900 Series (V2.8.02.C) 10G Routing Switch Hardware Manual	This manual describes device functions, technical characteristics and parameters, working principle, hardware structure, MCS, LIC, power module and fan plug-in box.
ZXR10 8900 Series (V2.8.02.C) 10G Routing Switch User Manual (Basic Configuration Volume)	This manual describes using and operation of device, system management, CLI privilege ranking configuration, port configuration, network protocol configuration, DHCP configuration, VRRP configuration, ACL configuration, QoS configuration, DOTIX configuration, cluster management configuration, network management configuration, IPTV configuration, VBAS configuration, CPU guard, URPF configuration and UDLD configuration.
ZXR10 8900 Series (V2.8.02.C) 10G Routing Switch User Manual (Ethernet Switching Volume)	This manual describes device VLAN configuration, STP configuration, MAC address table operation, link aggregation configuration, IGMP Snooping configuration, link protection configuration, Ethernet OAM configuration and EPON OLT configuration.

Manual	Summary
ZXR10 8900 Series (V2.8.02.C) 10G Routing Switch User Manual (IPv4 Routing Volume)	This manual describes static routing configuration, RIP configuration, OSPF configuration, IS-IS configuration, BGP configuration, load balancing configuration, multicast routing configuration, IP/LDP FRR configuration and BFD configuration.
ZXR10 8900 Series (V2.8.02.C) 10G Routing Switch User Manual (MPLS Volume)	This manual describes device MPLS configuration, MPLS L3VPN configuration and MPLS L2VPN configuration.
ZXR10 8900 Series (V2.8.02.C) 10G Routing Switch User Manual (IPv6 Volume)	This manual describes device IPv6 address configuration, IPv6 neighbor discovery protocol configuration, IPv6 tunnel configuration, IPv6 static routing configuration, RIPng configuration, OSPFv3 configuration, IS-ISv6 configuration and BGP+ configuration.
ZXR10 8900 Series (V2.8.02.C) 10G Routing Switch User Manual (DPI Volume)	This manual describes device signature symbol configuration, signature entry configuration, policy configuration, subservice configuration, service configuration and DPI-template configuration.
ZXR10 8900 Series (V2.8.02.C) 10G Routing Switch User Manual (FW Volume)	This chapter describes system management configuration, resource management configuration, FW function management, packet filtering and access control rule configuration, NAT configuration, protocol filtering configuration, intrusion prevention configuration, high availability configuration, and log and alarm configuration.
ZXR10 8900 Series (V2.8.02.C) 10G Routing Switch Command Manual (Command Index Volume)	This manual describes volume and section corresponding to each command in ZXR10 8900 series 10G routing switch.
ZXR10 8900 Series (V2.8.02.C) 10G Routing Switch Command Manual (IPv6 Volume)	This manual describes IPv6-related commands in ZXR10 8900 series 10G routing switch.
ZXR10 8900 Series (V2.8.02.C) 10G Routing Switch Command Manual (IP Routing Volume I)	This manual describes RIP, OSPF and IS-IS-related commands in ZXR10 8900 series 10G routing switch.

Manual	Summary
ZXR10 8900 Series (V2.8.02.C) 10G Routing Switch Command Manual (IP Routing Volume II)	This manual describes BGP, route map and routing policy-related commands in ZXR10 8900 series 10G routing switch.
ZXR10 8900 Series (V2.8.02.C) 10G Routing Switch Command Manual (MPLS Volume)	This manual describes MPLS-related commands in ZXR10 8900 series 10G routing switch.
ZXR10 8900 Series (V2.8.02.C) 10G Routing Switch Command Manual (QoS Volume)	This manual describes QoS-related commands in ZXR10 8900 series 10G routing switch.
ZXR10 8900 Series (V2.8.02.C) 10G Routing Switch Command Manual (Security Volume)	This manual describes security configuration-related commands in ZXR10 8900 series 10G routing switch.
ZXR10 8900 Series (V2.8.02.C) 10G Routing Switch Command Manual (Basic Configuration Volume I)	This manual describes system management, file management, user interface, log statistics, FTP/TFTP server and IPvr-related commands in ZXR10 8900 series 10G routing switch.
ZXR10 8900 Series (V2.8.02.C) 10G Routing Switch Command Manual (Basic Configuration Volume II)	This manual describes interface configuration, DHCP and VRRP-related commands in ZXR10 8900 series 10G routing switch.
ZXR10 8900 Series (V2.8.02.C) 10G Routing Switch Command Manual (Basic Configuration Volume III)	This manual describes NAT, Time Range, stack and DEBUG-related commands in ZXR10 8900 series 10G routing switch.
ZXR10 8900 Series (V2.8.02.C) 10G Routing Switch Command Manual (Network Management Volume)	This manual describes network management-related commands in ZXR10 8900 series 10G routing switch.
ZXR10 8900 Series (V2.8.02.C) 10G Routing Switch Command Manual (Ethernet Switching Volume)	This manual describes MAC, VLAN, SuperVLAN, STP, link aggregation, VBAS, MAC PING and UDLD-related commands in ZXR10 8900 series 10G routing switch.
ZXR10 8900 Series (V2.8.02.C) 10G Routing Switch Command Manual (Voice and Video Volume)	This manual describes VOIP and IPTV-related commands in ZXR10 8900 series 10G routing switch.
ZXR10 8900 Series (V2.8.02.C) 10G Routing Switch Command Manual (Multicast Volume)	This manual describes multicast protocol-related commands in ZXR10 8900 series 10G routing switch.

Manual	Summary
ZXR10 8900 Series (V2.8.02.C) 10G Routing Switch Command Manual (DPI Volume)	This manual describes DPI-related commands in ZXR10 8900 series 10G routing switch.
ZXR10 8900 Series (V2.8.02.C) 10G Routing Switch Command Manual (FW Volume)	This manual describes FW-related commands in ZXR10 8900 series 10G routing switch.

Commands supported by ZXR10 8900 series (V2.8.02.C) 10G routing switch are based on uniform platform ZXROS V4.8.22.

ZXR10 8900 Series (V2.8.02.C) 10G Routing Switch User Manual (FW Volume) contains the following chapters:

Chapter	Summary
Chapter 1 FW Overview	This chapter describes FW functional principle and management mode.
Chapter 2 System Management Configuration	This chapter describes basic concept, configuration and configuration example of FW system management.
Chapter 3 Resource Management Configuration	This chapter describes basic concept, configuration and configuration example of FW resource management.
Chapter 4 ZXR10 FW Function Management	This chapter describes basic concept, configuration and configuration example of FW function management.
Chapter 5 Packet Filtering and Access Control Rule Configuration	This chapter describes basic concepts, configurations and configuration examples of FW packet filtering and access control rule.
Chapter 6 NAT Configuration	This chapter describes basic concept, configuration and configuration example of FW NAT.
Chapter 7 Protocol Filtering Configuration	This chapter describes basic concept, configuration and configuration example of FW protocol filtering.
Chapter 8 Intrusion Prevension Configuration	This chapter describes basic concept, configuration and configuration example of FW intrusion prevention.
Chapter 9 High Availability Configuration	This chapter describes basic concept, configuration and configuration example of FW high availability.
Chapter 10 Log and Alarm Configuration	This chapter describes basic concept, configuration and configuration example of log and alarm.
Glossary	This part lists glossaries used in this manual.

Chapter 1

Firewall Overview

Table of Contents:

Function Overview

ZXR10 8900 Series Switch firewall (FW) service card has the following basic functions:

- Supporting routing and hybrid working modes;
- Supporting object-based network access control, including access control of network layer, application layer and other layers;
- Supporting NAT of multiple types of network addresses;
- Supporting built-in IDS module, which prevents Land, Smurf, TearOfDrop, Ping of Death, SynFlood, Targa3, IpSweep and another few attacks and has the function of anti-DOS/DDOS.
- Supporting hot standby between FW cards;
- Supporting FTP, TFTP, MMS, H.323, SIP, RSTP, SQLNET, and PPTP protocols.

ZXR10 8900 Series Switch FW has the following features:

- Adopting the design of multi-interfaces, providing sound network application scalability.
- Providing high-efficiency application layer access control. Proxy technology is used for traditional access control on application layer. System needs to switch among core layer, application layer and processes frequently, which consumes a lot system resources and influences performance.
- Showing flexible management. Network administrator can access FW through various interfaces for central management.
- Using a brand new management port protocol. With this protocol, multiple management services can be enabled on the unique service interface of FW.
- Providing high-performance content filtering. Core layer of system provides restore and security inspection to transmitted packets and implements high-performance content security protocol.

Working Principle

Data Flow Processing Flow

Generally, FW is used to control access from external untrusted networks (such as Internet) to internal trusted networks and mutual accesses among different areas within internal network. OS platform used by ZXR10 8900 Series Switch FW is the latest modular OS. By uploading a series of functional modules such as FW module and packet filtering module, FW module can control data flow traversing security device by setting access rules, packet filtering rules, interface properties and other mechanisms. ZXR10 8900 Series Switch FW takes the following basic steps to process packets:

1. Fast Forwarding

   As for a newly received legal packet, FW firstly searches session table to see if this packet has belonged to one existed session. If so, FW processes this packet according to corresponding session in the session table. When the packet matches access rule and address translation policy of this session, FW processes this packet fast. If the session is unavailable, it indicates this packet belongs to one new session. FW will retrieve routing table, address translation policy table and access rule table to collect policies related to this packet, that is entering "Receiving and Processing" flow.

2. Receiving and Processing

   ZXR10 8900 Series Switch FW module invokes related functional module and conducts initial processing to received packets. The following functional modules are invoked:

   - IDS Module, used to perform intrusion detection to packets. If the received packet matches IDS rule, it is regarded illegal and dropped.
   - IP-MAC binding module. If IP address and MAC address data contained in header of received packet break rules in IP-MAC binding table, the packet will be dropped.

3. Rule Matching

   At this step, FW matches the packet passing through receiving and processing step with a series of rules. The following modules are invoked:

   - PF module. PF module not only conducts L2/L3 protocol filtering to the packet, but also checks if the packet belongs to the service that can pass through.
   - Address translation module. Address translation policy gives processing method of received packet. ZXR10 8900 Series Switch FW module supports four address translation policies:
     - Forwarding directly

       FW doesn’t process packet and the packet is forwarded directly. This is default address translation policy of ZXR10 8900 Series Switch FW.

     - Translating source address

FW translates source IP address (or port id) of the received packet to preset IP address (or port id), and then forwards the packet whose source address is modified.

- Translating destination address

  FW translates destination IP address or port id of the received packet (FW interface address in usual cases) to preset IP address or port id (actual IP address or port id), and then forwards the packet whose destination address is modified.

- Bi-directional NAT

  FW translates source address and destination address (or port id) of the packet at the same time.

- Access control module. Access control rule defines if FW permits the packets matching rules to pass through. When receiving one packet, FW matches it with rules in access rule table one by one according to policy sequence number and processes the packet according to operation (permit or deny) specified by corresponding policy. If corresponding access policy fails to be matched, the packet will be forwarded to destination interface. ZXR10 8900 Series SwitchFW will proces this packet according to default property (permit or deny) of the area where destination interface locates.

4. Session Establishment

   As for the packet with no session for matching, ZXR10 8900 Series SwitchFW will create one new record in session table according to packet processing information in steps 1-3, including packet destination address, source address, route, address translation policy, access rule and other information. Packets of this session received after this new record will be processed according to record in the session table.

5. Processing before Routing

   When policy changes during communication process, FW will re-invoke packet filtering module and access control module to match the packet with policy.

6. Route Querying

   ZXR10 8900 Series SwitchFW module selects packet forwarding interface according to routing table or MAC address table learned on each interface. If packet address is translated, FW will search NAT table to find the actual address for routing.

Matching Access Control Rules

Access control rules are a set of policies customized by user. These rules can define what packets (meeting certain conditions) can pass FW and what packets (meeting some other conditions) will be denied by FW. Data contained in each access policy include: source address and destination address of the packet, service (protocol type and port id) and operations (forwarding or dropping) performed to the packets meeting conditions.

In access policy, policy source defines the source of packet, which can be one or multiple objects (such as host, subnet, scope and so on). When source address of the packet belongs to the scope of policy source, it is believed to meet constraint conditions of policy source.

Policy destination defines the scope of destination address. The same as policy source, policy destination can contain one or multiple hosts, subnet, scope and multiple areas (or VLAN).

Policy service defines network protocol used by packet and specific port id.

Access control defines FW operations to the packet meeting policies, including permit (permit the packet to pass through) and deny (drop this packet).

In the case that a packet matches one access policy, it indicates source address of the packet is within the scope defined by policy source, destination address of the packet is within the scope defined by policy destination, port id corresponding to the packet is contained in policy service, and packet receiving time meet the requirement of policy access time (if access time is defined). Only when one packet meets all conditions required by the policy, this policy matches this packet.

It shall be noted that one content filtering policy and one application identity policy shall be defined for each access rule to filter and inspect data at application layer. FW searches the access policy matching a packet according at the following steps:

ZXR10 8900 Series Switch FW module retrieves access control rule table according to sequence of access policies and matches policies with packet one by one. Once an access policy is found to match the packet, FW stops checking matched access policy and processes the packet (permit or deny) according to rules defined in the first matched access policy. If no access policy is found to match this packet, FW will process this packet according to default access control properties on packet sending interface.

If the packet is forbidden to be forwarded, it will be dropped; if the packet is permitted to be forwarded, check if this policy defines DPI policy or application identity policy.

If application identity policy is defined in the policy, check to see if any protocol of the application identity policy is used in application layer of the packet. If corresponding protocol is used, process this packet according to operations defined by this application identity policy.

Working Modes

ZXR10 8900 Series Switch FW protects VLAN interfaces and supports two working modes: routing mode and hybrid mode.

Route Mode In this mode, ZXR10 8900 Series Switch FW protects L3 packets on protected vlan interface. All L3 packets passing through protected vlan are forwarded only after being processed by FW module. This mode is applicable to the case when each area is in a separate network segment. Similar to router, IP address shall be configured for each vlan interface in routing mode or hybrid mode according to area planning.

Hybrid Mode In this mode, ZXR10 8900 Series Switch FW protects L2 and L3 packets on protected vlan interface. No matter internal L2 packets of the protected vlan or L3 packets cross-vlans are forwarded after being processed by FW module.

Management Modes

Network administrator can manage ZXR10 8900 Series Switch FW module in many ways, including:

- Through CONSOLE (perform local management through CONSOLE port)
- Through TELNET (perform remote management by logging into FW through Telnet)
- Through WEBUI (perform remote management by logging into FW through browser)

Logging into FW through Console Port

Context It is available to log into FW module through CONSOLE port and conduct some basic settings on FW.

Steps

1. Using one serial console cable (included in factory accessories) to connect serial port of PC (assume that com1 is available) and console port of FW.
2. Setting properties of serial port according to the following parameters:

Parameter	Description
Bits per Second (baud rate):	9600
Data Bits:	8
Parity Check:	Null
Stop Bits:	1

3. Logging into switch main board, accessing FW template configuration node under config node, and inputting the following command to access FW card.

Command	Function
`ZXR10(fw-template-1)#`**session**	It accesses FW card from main board.

4. Logging into ZXR10 8900 Series Switch FW module by inputting system default username. User can perform configuration management through command line after accessing FW module.

Tip:

Both username and password are case sensitive.

END OF STEPS.

Logging into FW through Telnet

Context It is available to log into FW module through Telnet and conduct some basic settings on FW.

Steps

1. Selecting the interface of vlan where administrator locates and configuring IP address for the interface.

Command	Function
ZXR10(config-if-vlan1)#**ip address** < *ipaddress* >< *maskaddress* >	It configures IP address and subnet mask for L3 vlan 1.

Parameter Description:

Parameter	Description
< *ipaddress* >	It is IP address, in form of A.B.C.D.
< *maskaddress* >	It is subnet mask, such as 255.255.255.0.

2. Accessing FW configuration node and configuring IP address of VLAN interface to management IP of managed FW card.

Command	Function
ZXR10(config-fw)#**bind mng-ip** <*slot number* >< *ipaddress* >	It configures IP address of VLAN interface to management IP of managed FW card.

Parameter Description:

Parameter	Description
<*slot number* >	It is the number of slot where FW card locates.
< *ipaddress* >	It corresponds to above IP address of L3 vlan interface, in form of A.B.C.D.

3. Running telnet < ipaddress >, that is IP address configured in step 2, on administrator PC to access configuration interface of FW card.

END OF STEPS.

Example The following steps show how to log into FW through Telnet:

1. Binding fw-template 1 with slot number.

```
ZXR10(config) # fw ZXR10(config-fw) # fw-template 1
ZXR10(config-fw-template-1) # bind slot 2
ZXR10(config-fw-template-1) # exit ZXR10(config-fw) # exit
```

2. Configuring IP address for L3 vlan interface.

```
ZXR10(config) # vlan 2 ZXR10(config-vlan2) # exit
ZXR10(config) # int vlan 2
ZXR10(config-if-vlan2) ip addr  10.2.2.1  255.255.255.0
```

3. Accessing FW configuration node and configuring IP address of VLAN interface to management IP of managed FW card.

```
ZXR10(config-if-vlan2) exit ZXR10(config) fw
ZXR10(config- fw) bind mng-ip 2  10.2.2.1
```

4. Accessing FW card through terminal telnet.

```
telnet 10.2.2.1
```

Logging into FW through Browser

Context It is available to log into FW module through browser and conduct some basic settings on FW.

Steps

1. Selecting the interface of vlan where administrator locates and configuring IP address for the interface.

Command	Function
ZXR10(config-if-vlan1)#**ip address** < *ipaddress* >< *maskaddress* >	It configures IP address and subnet mask for L3 vlan 1.

Parameter Description:

Parameter	Description
< *ipaddress* >	It is IP address, in form of A.B.C.D.
< *maskaddress* >	It is subnet mask, such as 255.255.255.0.

2. Accessing FW configuration node and configuring IP address of VLAN interface to management IP of managed FW card.

Command	Function
ZXR10(config-fw)#**bind mng-ip** <*slot number* >< *ipaddress* >	IP address of vlan 1 interface is management IP of FW card in slot 2.

Parameter Description:

Parameter	Description
<*slot number* >	It is the number of slot where FW card locates.
< *ipaddress* >	It is IP address, in form of A.B.C.D.

3. Administrator inputs FW management URL (such as https://< ipaddress >) on browser of management host and login interface pops up.

END OF STEPS.

Example The following steps show how to log in FW through browser https.

1. Binding fw-template 1 with slot number.

```
ZXR10(config) # fw ZXR10(config-fw) # fw-template 1
ZXR10(config-fw-template1) # bind slot 2
```

2. Configuring IP address for L3 vlan interface.

```
ZXR10(config) # vlan 2 ZXR10(config-vlan2) # exit
ZXR10(config) # int vlan 2 ZXR10(config-if-vlan2) ip addr
10.2.2.1  255.255.255.0
```

3. Accessing FW configuration node and configuring IP address of VLAN interface to management IP of managed FW card.

```
ZXR10(config-if-vlan2) exit ZXR10(config) fw
ZXR10(config- fw) bind mng-ip 2  10.2.2.1
```

4. Logging in FW card through https.

```
https : // 10.2.2.1
```

Chapter 2

System Management Configuration

Table of Contents:

System Management Overview

In system command module, user can configure basic information of ZXR10 8900 Series Switch FW service card, such as version display, clock management, NTP setting, system configuration management, system upgrade, authentication user management, administrator information, FW reboot command and so on.

To access command module, execute the following command:

#**system**

To exit from this command module, execute the following command:

#**exit**

After logging into FW and accessing this command module, CLI administrator can execute corresponding component management commands. The following parts will introduce all component management commands under this command module. The format of command in the example is that after accessing this command module.

Querying System Basic Information

User can search model, software platform version, system current configuration and other information of current device in system command module.

1. Displaying system version information

Command	Function
ZXR10_FW.system #**version**	It displays system version information.

2. Displaying running statuses of system services

Command	Function
ZXR10_FW.system #**service status**	It displays running statuses of system services (such as server state and if various services are enabled).

3. Displaying system name

Command	Function
ZXR10_FW.system #**devname show**	It displays system name.

4. Displaying current configuration of system

Command	Function
ZXR10_FW.system #**config show_running**	It displays current configuration of system.

Querying System Running Information

Running information indicates current system CPU, memory, other occupation information of system resources, and connection information established through FW.

1. Viewing current running status of device.

Command	Function
ZXR10_FW.system #**information**	It views current running status of device, including memory information, CPU utilization and other information.

2. Showing network connection information.

Command	Function
ZXR10_FW.system #**netstat**	It shows network connection, routing table and network interface information, and thus user can learn which network connections are being used currently.

Configuring System Management

Setting System Parameters

User can set administrator login parameter and connection time-out parameter, and view session statistics on **authset** command module. System parameter specifies the max login failures for the same one administrator and concurrent administrators, and managing login site. Once the login failure number of an administrator exceeds threshold, system will lock the login to prevent illegal users logging into ZXR10 8900 Series Switch FW service card through brute force of password.

To access **authset** command module, execute the following command:

#**authset**

To exit from this command module, execute the following command:

#**exit**

1. Setting max authentication failure-related parameter.

Command	Function
ZXR10_FW.system authset #**authfail set maxnum** <*number*>	This sets max authentication failure-related parameter. This can prevent brute force of password.

Parameter Description:

Parameter	Description
set maxnum	Setting system name
<*number*>	This is the max number, in range of 1-10.

Example:

To set max authentication failure number to 5, execute the following command:

```
ZXR10_FW.system .authset # authfail set maxnum 5
```

2. Showing max authentication failure number.

Command	Function
ZXR10_FW.system authset #**authfail show**	This shows max authentication failure number.

Example:

To show max authentication failure number, execute the following command:

```
ZXR10_FW.system .authset # authfail show
```

3. Setting authentication faillock time-related parameter.

Command	Function
ZXR10_FW.system authset #**faillock set time**<*number*>	This sets authe ntication faillock time-related parameter.

Parameter Description:

Parameter	Description
set time	This sets authentication faillock time.
<*number*>	This is the faillock time, in range of 60-3600, in seconds.

Example:

To set authentication faillock time, execute the following command:

```
ZXR10_FW.system .authset #  faillock set time 60
```

4. Showing authentication faillock time.

Command	Function
ZXR10_FW.system authset #**faillock show**	This shows authentication faillock time.

5. Setting max concurrent management site-related parameter.

Command	Function
ZXR10_FW.system authset #**managermaxlogin set maxnum**<*number*>	This sets max concurrent management site-related parameter, that is, setting the max number of sites (IP) from which the same administrator name can log into the same one device. The default value is 5, in range of 1-32. For example, one device is configured with multiple IP addresses and this command limits the number of IP addresses, with which one user can log in.

Parameter Description:

Parameter	Description
set maxnum	This sets max concurrent management site-related parameter.
<*number*>	This is the max number, in range of 1-32.

Example:

To set max concurrent management sites to 16, execute the following command:

```
ZXR10_FW.system .authset # managermaxlogin set maxnum 16
```

6. Showing max concurrent management site-related parameter.

Command	Function
ZXR10_FW.system authset #**managermaxlogin show**	This shows max concurrent management site-related parameter.

7. Setting max concurrent administrator number.

Command	Function
ZXR10_FW.system authset #**maxonlineadm set maxnum**<*number*>	It sets max concurrent administrator number.

Parameter Description:

Parameter	Description
set maxnum	This sets max concurrent administrator number.
<*number*>	This is the max number, in range of 1-256.

Example:

To set the max concurrent administrator number to 16, execute the following command:

```
ZXR10_FW.system .authset # maxonlineadm set maxnum 16
```

8. Showing max concurrent administrator number.

Command	Function
ZXR10_FW.system authset #**maxonlineadm show**	This shows max concurrent administrator number.

9. Setting all authentication parameters to default values.

Command	Function
ZXR10_FW.system authset #**set default**	This sets all authentication parameters to default values.

10. Setting the max login number for the same one user.

Command	Function
ZXR10_FW.system authset #**usermaxlogin set maxnum** <*number*>	This sets the max login number for the same one user.

Parameter Description:

Parameter	Description
set maxnum	This sets the max login number for the same one user.
<*number*>	This is the max number, in range of 1-2000.

Example:

To set the max login number for the same one user to 4, execute the following command:

```
ZXR10_FW.system .authset # usermaxlogin set maxnum 4
```

11. Showing the max concurrent login user number set on system.

Command	Function
ZXR10_FW.system authset #**usermaxlogin show**	This shows the max concurrent login user number set on system.

Managing System Services

System services indicate management/monitoring services conducted by FW device, including monitoring service, SSH service, Telnet service, HTTP service and NTP service.

- Monitoring service is used by user to remotely monitor running status of this device.
- SSH service is used by user to remotely manage device through SSH protocol.
- Telnet service is used by user to remotely manage device through Telnet protocol.
- HTTP service is used by user to remotely manage device through HTTP protocol.
- NTP service is used to synchronize system time through NTP protocol.

In factory configuration, only HTTP service is in running status. To disable this system service, user will fail to manage system through WebUI. FW system provides control (enable and disable) over these functions. The detailed commands are as follows:

1. Eabling monitoring service.

Command	Function
ZXR10_FW.system#**monitord start**	This enables monitoring service.

2. Disabling monitoring service.

Command	Function
ZXR10_FW.system#**monitord stop**	This disables monitoring service.

3. Enabling SSH service.

Command	Function
ZXR10_FW.system#**sshd start**	This enables SSH service.

4. Disabling SSH service.

Command	Function
ZXR10_FW.system#**sshd stop**	This disables SSH service.

5. Enabling Telnet service.

Command	Function
ZXR10_FW.system#**telnetd start**	This enables Telnet service.

6. Disabling Telnet service.

Command	Function
ZXR10_FW.system#**telnetd stop**	This disables Telnet service.

7. Enabling HTTP service.

Command	Function
ZXR10_FW.system#**httpd start**	This enables HTTP service.

8. Disabling HTTP service.

Command	Function
ZXR10_FW.system#**httpd stop**	This disables HTTP service.

Note:

When enabling corresponding system service, system will enable corresponding service program on background to provide this service. However, to use this service to manage or monitor device, it is necessary to add corresponding service control rule in Open Service. For details, please refer to section 2.4.3.

Only HTTP service is enabled by default in factory configuration.

Setting Open Services

To improve device security, system provides granularity access control to communication management between user and device and among devices. User can strengthen access control by setting open service control rules. System can manage the following service types:

Service	Description
GUI	It allows user to configure and manage device through ZXR10 8900 Series Switch FW service card manager.
WEBUI	It allows user to configure and manage device through WEBUI.
MONITOR	It allows user to monitor running status of device according to preset conditions.
PING	It allows user to ping physical interface address of device, vlan virtual interface address and sub-interface addresses.

Service	Description
Telnet	It allows user to configure and manage device through TELNET.
IDS	It allows interaction with IDS device.

User can implement simple L2/L3 access control by setting packet filtering policy. When receiving one packet, device will match it with packet filtering policy sequentially. In case no policy is matched, the packet will be processed according to default rule. According to factory configuration, all packets can pass FW by default.

To access this command module, execute the following command:

#**pf**

To exit from this command module, execute the following command:

#**exit**

After logging into GW and accessing this command module, CLI administrator can execute corresponding component management commands. The following parts will introduce all component management commands under this command module. The format of command in the example is that after accessing this command module.

1. Adding one open service rule.

Command	Function
ZXR10_FW.pf #**service add name** <*gui*|**snmp**|**ssh**|**monitor**|**ping**|**telnet**|**ids**|**auth**|**ntp**|**update**|**dhcp**|**rip**|**l2tp**|**pptp**|**webui**|**ipsecvpn**|**sdmi** > **area** <*string1*><*addressid* <*number1*>| **addressname** <*string2*>>	This adds one open service rule.

Parameter Description:

Parameter	Description
add	This adds one open service rule.
name	This selects the name of service opened by GW.
gui	It is GUI service.
snmp	It is SNMP service.
ssh	It is SSH service.
monitor	It is MONITOR service.
ping	It is PING service.
telnet	It is telnet service.
ids	It is IDS service.
auth	It is AUTH service.
ntp	It is NTP service.
update	It is upgrade service.

Parameter	Description
dhcp	It is DHCP service.
rip	It is RIP service.
l2tp	It is L2TP service.
pptp	It is PPTP service.
webui	It manages GW through WEBUI.
ipsecvpn	It is the service opened when establishing IPSEC tunnel.
sdmi	It is security management service. It allows to manage FW device through military security management platform.
area	It selects the area from which service request is sent. The area must be selected from existing ones. For configuration and illustration of area, please refer to section **Configuring Area Resource**.
<*string1*>	It is a string, the name of area.
addressid	It sets ID for permitted address object.
<*number1*>	It is a number.
addressname	It sets name for permitted address object.
<*string2*>	It is a string. It must be a preset host, subnet or scope address object.

Command Illustration:

Parameters addressid and addressname can be used at the same time. However, it must be confirmed that the objects corresponding to addressid and addressname are unique, or services will fail to be added.

Example:

To open webui service for area_intervlan0 (where area_intervlan0 is the preset area object), execute the following command:

```
ZXR10_FW.pf # service add name webui area
area_intervlan0 addressname any
```

2. Modifying one open service rule.

Command	Function
ZXR10_FW.pf #**service modify id** <*number*>[**name** <*gui*|**snmp**|**ssh**|**monitor**|**ping**|**telnet**|**ids**|**auth**|**ntp**|**update**|**dhcp**|**rip**|**l2tp**|**pptp**|**webui**|**ipsecvpn**>][**area** <*string1*>][**addressid** <*number1*>][**addressname** <*string2*>]	This modifies one open service rule.

Parameter Description:

Parameter	Description
modify	This modifies one open service rule.

Parameter	Description
id	This specifies ID for the rule.
<*number*>	This is a number and must be the id of a rule that has been added.

Command Illustration:

Service rule can be modified except for id.

Example:

To modify the service whose id is 8361 and open gui service, execute the following command:

```
ZXR10_FW.pf #service modify id 8361 name gui
area area_intervlan0 addressname any
```

3. Showing one open service rule.

Command	Function
ZXR10_FW.pf #**service show** [**name** <*gui*|**snmp**|**ssh**|**monitor** |**ping**|**teln et**|**ids**|**auth**|**ntp**|**update**|**dhcp** |**rip** | **l2tp**|**pptp**|**webui** |**ipsecvpn** >]	This shows one open service rule.

Parameter Description:

Parameter	Description
show	This shows one open service rule.
name	This selects name of the service to be viewed.

Command Illustration:

When type is selected, the setting of specified service type is shown; when type is not selected, all rules are shown.

Example:

To show gui open service rule, execute the following command:

```
ZXR10_FW.pf #service show name  gui
```

4. Deleting one open service rule.

Command	Function
ZXR10_FW.pf #**service delete id** <*number*>	This deletes one open service rule.

Parameter Description:

Parameter	Description
delete	This deletes one open service rule.
id	This selects id of service opened by GW.
<*number*>	It is a number.

Command Illustration:

To view the id needed for deleting one service, execute command **service show**.

Setting WEBUI Authentication

WEBUI authentication means administrator can access ZXR10 8900 Series Switch FW service card only after passing both certificate authentication and username/password authentication.

1. Restoring WEBUI system root certificate.

Command	Function
ZXR10_FW.system #**webui cert** <*restore*>	This restores WEBUI system root certificate.

Parameter Description:

Parameter	Description
cert	Setting WEBUI system root certificate.
<*restore*>	This restores root certificate.

Tip:

- In WEBUI authentication, FW system must import PEM format certificate and client must import PCKS#12 format certificate. Client can obtain this certificate from CA. For details, please contact enterprise certificate administrator.
- When enabling WEBUI authentication, to log into WEBUI management interface, administrator must provide corresponding certificate for passing authentication.
- Before authentication, administrator needs to import personal certificate into Internet browser. For details, please contact enterprise certificate administrator.

2. Showing WEBUI setting.

Command	Function
ZXR10_FW.system #**webui show**	This shows WEBUI setting.

3. Seting WEBUI timeout time.

Command	Function
ZXR10_FW.system #**webui idle-timeout** <*number*>	This sets WEBUI timeout time.

Parameter Description:

Parameter	Description
idle-timeout	This sets WEBUI timeout time.
<*number*>	This is an interval, in range of 30-3600 or 0, in seconds. The default value is 180. As for number 0, it indicates WEBUI will never timeout.

Example:

To set WEBUI timeout time to 60 seconds, execute the following command:

```
ZXR10_FW.system # webui idle-timeout 60
```

Configuration Maintenance

Maintenance includes operations of viewing, uploading and downloading configuration file of system, restoring factory configuration, restarting device and other functions.

Configuration Maintenance Overview

System configuration indicates configurations and files of all functional modules in the entire FW, including FW configuration (including network basic configuration), VPN configuration and AV configuration.

There are two types of system configuration:

- The first one is saving configuration, which is the configuration file manually saved on the device by user for the last time. When system reboots, this configuration file will be loaded automatically.
- The second one is running configuration, which shows configuration when device is in running state. This configuration can be dynamically adjusted according to operations of users. But when system reboots, this configuration will get invalid. Running configuration is different from saving configuration. For example, after one user adds some rules, these rules join running configuration and get valid immediately, but will not join saving configuration until user saves them and these rules will get invalid after system reboots.

System provides maintenance to configuration of FW device. User can perform some maintenance operations on device, such as viewing saving configuration and running configuration, uploading and downloading system configuration file (that is import and export all system configurations for one time) and others. System also enables administrator to restore configuration to factory ones for user reconfiguration.

Configuring Maintenance

1. Validating the configurations newly added to system.

Command	Function
ZXR10_FW.system #**config implement**	This validates the configurations newly added to system.

Command Illustration:

With this command, the newly added configurations get valid on device immediately but they are not saved. To apply these configurations next time when enabling FW, it is needed to save configurations.

2. Loading default configuration.

Command	Function
ZXR10_FW.system #**config reset**	This loads default configuration.

3. Saving current system configuration.

Command	Function
ZXR10_FW.system #**config save**	This saves current system configuration.

4. Showing configurations previously saved on system.

Command	Function
ZXR10_FW.system #**config show**	This shows configurations previously saved on system.

5. Showing current system configuration.

Command	Function
ZXR10_FW.system #**config show_running**	This shows current system configuration.

Restoring System

Command	Function
ZXR10_FW.system #**service default**	This restores default system service factory configuration. All services are enabled by default.

Rebooting System

Command	Function
ZXR10_FW.system #**reboot**	This reboots FW.

Configuring System Manager

System Manager Overview

ZXR10 8900 Series Switch FW service card supports management and operation by multiple users. Different users have different operation privileges. Root system manager has global privilege to configuration information and can view configuration information of all public interface factors corresponding to this privilege. Where superman is the unique super manager in system and has all management privileges in ZXR10 8900 Series Switch FW service card.

Configuring System Manager

As for ZXR10 8900 Series Switch FW service card, only super manager can configure manager account and add another manager.

1. Adding device manager information: name, password and privilege information.

Command	Function
ZXR10_FW. system #**admininfo add** **input manager's name:**<*string1*> **new password:**<*string2*> **re_enter password:** <*string2*> **choose manager's privilege**[**audit**|**config**|**vs**]:<*audit*|*config*|*vs*> **input the comment**[**y/n**]: <*y*|*n*> **input the comment:** <*string3*>	This adds device manager information: name, password and privilege information.

Parameter Description:

Parameter	Description
add	This adds device manager.
name	This is the name of device manager.
<*string1*>	This is a name string.
password	This sets password.
<*string2*>	This is a password string.
privilege	This sets privilege.
audit| **config** |**vs**	Manager has the following types: security audit|security management|virtual system. Manager of security audit type can view system security and configuration information, but cannot modify configuration. Manager of security management type has additional right of modifying configuration.
comment	It is the comment.
y|**n**	It specifies whether to give comment.
<*string3*>	It the content of comment.

Command Illustration:

When manager is of security management type, the command prompt is "%" after accessing command line interface.

When manager is of security audit type, the command prompt is "$" after accessing command line interface.

2. Modifying system manager information: comment, name, password, privilege and the type of system that manager belongs to.

Command	Function
ZXR10_FW. system #**admininfo modify** **input manager's name:**<*string1*> **new password:**<*string2*> **re_enter password:** <*string2*> **choose manager's privilege**[**audit** | **config**]: <*audit*|*config*> **input the comment**[**y/n**]: <*y*|*n*> **input the comment:** <*string3*>	This modifies system manager information: comment, name, password, privilege and the type of system that manager belongs to.

Parameter Description:

Parameter	Description
modify	This modifies information about device manager.
name	This is the name of device manager.
<*string1*>	This is a name string.
<*string2*>	This is a password string.
new_password	This sets password.
privilege	This sets privilege.
audit | **config**]	Manager has the following types: security audit|security management.
comment	It is the comment.

Command Illustration:

Only super manager can modify the name of manager.

3. Deleting information of a manager in database.

Command	Function
ZXR10_FW. system #**admininfo delete_db_manager name** <*string*>	This deletes information of a manager in database.

Parameter Description:

Parameter	Description
delete_db_manager	This deletes information of a manager in database.
name	This is the name of device manager.
<*string*>	This is a name string.

4. Showing name, privilege, comments and other information of manager in database.

Command	Function
ZXR10_FW. system #**admininfo showdb**	This shows name, privilege, comments and other information of manager in database.

5. Showing names, login addresses and online time of online managers.

Command	Function
ZXR10_FW. system #**admininfo showonline**	This shows names, login addresses and online time of online managers.

System Manager Configuration Example

1. Adding device configuration security management manager "test".

   ```
   # admininfo add input manager’s name:test new
   password: re_enter password: choose manager’s
   privilege[audit | config ]:config input the comment[y/n]:y
   input the comment:config_test_user
   ```

   It prompts manager is added successfully.

   ```
   Add this manager successfully
   ```

2. Modifying information of manager test, set it to security audit manager, and modify password, comments and other information.

   ```
   #admininfo modify input manager’s name:test
   modify the name(only super admin can change name)[y/n]:y
   input new name:audittest new password:22222222 re_enter
   password:22222222 modify the privilege[y/n]:y choose
   manager’s privilege[audit | config ]:audit modify the
   comment[y/n]:y input the comment:audit_user
   ```

3. Deleting the manager named test.

   ```
   # admininfo delete_db_manager name test
   ```

Chapter 3

Resource Management Configuration

Table of Contents:

Resource Management Overview

ZXR10 8900 Series SwitchMost functions of FW service card are based on resource, such as access control policy, address translation policy, server load balancing policy, authentication management and so on. It is necessary to define resources of various types before manager configures ZXR10 8900 Series Switch FW service card.

The using of concept resource simplifies management to ZXR10 8900 Series Switch FW service card. When one resource changes, manager only needs to modify properties of resource and doesn't need to modify all policies and rules related to this resource.

As for ZXR10 8900 Series Switch FW service card, user can customize the following resource types:

- Address resource: It includes host resource, address range resource, subnet resource and address group.
- Property resource: It includes property resource and property group. Proper resource can get valid only when bound with other resources (such as interface resource, sub-interface resource, area resource and so on).
- Area resource: It defines area access privilege by being bound with property resource.
- Time resource: It includes time resources for multi-cycles and single-cycle.
- Service resource: It includes system-defined service resource, customized service resource and service group.

Note:

The following special characters cannot be present in resource name: space, "'", """, "\", ";", "'", "$", "&", "@", "%", "|", "~", "<", ">", "#", "+", "!", "=", "^", "?", "`" (the key under "~").

ZXR10 8900 Series SwitchIt is available to rename resource on FW service card.

Configuring Address Resource

Address Resource Configuration Overview

Configuration of address resource is the most basic one in resource management. It needs to select different address resources when defining access control rules and address translation rules. User can set address resources of various types, such as host resource, address range resource and subnet resource, and meanwhile can define address group to add all these address resources into address group.

For setting of various address resources and address group, please refer to the following sections.

User can perform management and configuration to above resources in DEFINE module of ZXR10 8900 Series Switch FW card.

To access this command module, execute the following command:

#**define**

To exit this command module, execute the following command:

#**end**

Setting Host Resource

1. Adding host.

Command	Function
ZXR10_FW.define#**host add** <*name <string1>*>[**ipaddr** <*string2*>][**macaddr** <*macaddress*>][**session** <*number1*>][**halfsession** <*number2*>]	This adds one host.

Parameter Description:

Parameter	Description
add	This adds one host.
name	This sets name for the host to be added.
<*string1*>	This is one string, indicating name of the host.
ipaddr	This sets IP address for the host.
<*string2*>	This is one string, indicating IP address, in format of 192.168.1.6. It can be one or more IP addresses. As for multiple IP addresses, space is used between each two IP addresses and all addresses are quoted with single quotes.
macaddr	This sets MAC address for the host.
<*macaddress*>	This is one string, indicating MAC address, in format of 00:00:00:00:00:00.
session	This sets the number of session.
<*number1*>	This is one number, indicating the number of sessions on host.
halfsession	This sets the number of half-sessions.
<*number2*>	This is one number, indicating the number of half-sessions on host.

Command Illustration:

Multiple IP addresses (no more than 120) can be added to single-host resource to control multi-IP user.

Example:

To add host1 and set its IP addresses to 192.168.1.8 and 192.168.1.9, mac address to 1a:21:7b:13:11:5c, the number of session on host to 1 and half-session to 1, execute the following command:

```
ZXR10_FW.define# host add name host1 ipaddr
‘192.168.1.8  192.168.1.9’ macaddr 1a:21:7b:13:11:5c
session 1 halfsession 1
```

2. Modifying one host.

Command	Function
ZXR10_FW.define#**host modify name** <*string1*>[**ipaddr** <*string2*>][**macaddr** <*macaddress*>][**session** <*number1*>][**halfsession** <*number2*>]	This modifies one host.

Parameter Description:

Parameter	Description
modify	This modifies one host.
name	This specifies the name of host to be modified.
<*string1*>	This is one string, indicating name of the host.

Parameter	Description
ipaddr	This specifies one new IP address.
<*string2*>	This is one string, indicating IP address.
macaddr	This specifies one new MAC address.
<*macaddress*>	This is one string, indicating MAC address.
session	This specifies the new number of max sessions.
<*number1*>	This is one number, indicating the number of max sessions.
halfsession	This modifies the number of half-sessions.
<*number2*>	This is one number, indicating the number of half-sessions on host.

Example:

To modify host1 and set its IP addresses to 192.168.1.8 and 192.168.1.9, mac address to 1a:21:7b:13:11:5c, the number of session on host to 1 and half-session to 1, execute the following command:

```
ZXR10_FW.define# host modify name host1
ipaddr ‘192.168.1.8  192.168.1.9’ macaddr
1a:21:7b:13:11:5c session 1 halfsession 1
```

3. Renaming one host.

Command	Function
ZXR10_FW.define#**host rename oldname** <*string1*> **newname** <*string2*>	This renames one host.

Parameter Description:

Parameter	Description
rename	This renames one host.
oldname	This specifies the name of host to be renamed.
<*string1*>	This is one string, indicating the name of host (the host name has been defined).
newname	This specifies new name for one host.
<*string2*>	This is one string, indicating new name of the host.

Example:

To modify the name of one host from host1 to host2, execute the following command:

```
ZXR10_FW.define#host rename oldname host1
newname host2
```

4. Deleting one host.

Command	Function
ZXR10_FW.define#**host delete** [**id** <*number1*>][**name** <*string*>]	This deletes one host.

Parameter Description:

Parameter	Description
delete	This deletes one host.
id	This specifies ID of the host to be deleted.
<*number1*>	This is one number, indicating ID of host.
name	This specifies the name of host to be deleted.
<*string*>	This is one string, indicating name of the host.

Command Illustration:

To delete one host, it is available to delete the host according to host name, host id or both. However, in case host id and host name are inconsistent, host name shall apply.

When no parameter is given, the host not quoted by policy is deleted.

Example:

To delete the host whose name is host1, execute the following command:

```
ZXR10_FW.define# host delete name host1
```

5. Deleting all hosts not quoted by policy.

Command	Function
ZXR10_FW.define#**host clean**	This deletes all hosts not quoted by policy.

6. Viewing all hosts.

Command	Function
ZXR10_FW.define#**host show**	This views all hosts.

Setting Address Range Resource

1. Adding address configuration range.

Command	Function
ZXR10_FW.define#**range add name** <*string1*> **ip1** <*string2*> **ip2** <*string3*>[**except** <*string4*>][**session** <*number1*>]	This adds address configuration range.

Parameter Description:

Parameter	Description
add	This adds address range.
name	This sets name for address range.
<*string1*>	This is one string, indicating the name of address range.
ip1	This sets start IP address for address range.
<*string2*>	This is one string, indicating IP address, in format of 0.0.0.0.
ip2	This sets end IP address for address range.
<*string3*>	This is one string, indicating IP address, in format of 0.0.0.0.
except	This sets except IP address in address range.
<*string4*>	This is one string, indicating IP address, in format of 0.0.0.0.
session	This sets the number of session.
<*number1*>	This is one number, indicating the number of sessions.

Command Illustration:

The value of ip1 mustn't be larger than that of ip2, or it will report error. The value of parameter **Except** shall be within the range between Ipaddress1 and Ipaddress2.

The default range configuration for ZXR10 8900 Series Switch FW service card is any0.0.0.0-255.255.255.255. At the same moment, the number of connections of individual addresses within the address range cannot exceed the number of max sessions.

Example:

To add address range1 and set the range to 192.16.1.10-192.16.2.81, execute the following command:

```
ZXR10_FW.define# range add name range1 ip1
192.16.1.10  ip2  192.16.2.81
```

2. Modifying address configuration range.

Command	Function
ZXR10_FW.define#**range modify name** <*string1*> **ip1** <*string2*> **ip2** <*string3*>[**except** <*string4*>][**session** <*number1*>]	User can add, modify and delete address range in management of address range of FW.

Parameter Description:

Parameter	Description
modify	This modifies address range.
name	This sets name for address range to be modified.
<*string1*>	This is one string, indicating the name of address range.

Parameter	Description
ip1	This sets start IP address for address range.
<*string2*>	This is one string, indicating IP address, in format of 0.0.0.0.
ip2	This sets end IP address for address range.
<*string3*>	This is one string, indicating IP address, in format of 0.0.0.0.
except	This sets except IP address in address range.
<*string4*>	This is one string, indicating IP address, in format of 0.0.0.0.
session	This sets the number of session.
<*number1*>	This is one number, indicating the number of sessions.

Command Illustration:

At the same moment, the total number of connections of all hosts within the address range cannot exceed the number of max sessions.

Example:

To modify address range to 192.16.1.11—192.16.2.82 after adding range1 with address 192.16.2.1 excepted, execute the following command:

```
ZXR10_FW.define# range modify name range1
ip1 192.16.1.11 ip2  192.16.2.82 except 192.16.2.1
```

3. Renaming address configuraiton range.

Command	Function
ZXR10_FW.define#**range rename oldname** <*string1*> **newname** <*string2*>	This renames address range.

Parameter Description:

Parameter	Description
rename	This renames address range.
oldname	This specifies the name of address range to be renamed.
<*string1*>	This is one string, indicating the name of address range (the name of address range has been defined).
newname	This specifies new name for address range.
<*string2*>	This is one string, indicating the new name of address range.

Example:

To rename address range1 to range2, execute the following command:

```
ZXR10_FW.define# range rename oldname
range1 newname range2
```

4. Deleting one address range.

Command	Function
ZXR10_FW.define#**range delete** [**id** <*number1*>][**name** <*string*>]	This deletes one address range.

Parameter Description:

Parameter	Description
delete	This deletes address range.
id	This specifies ID of the address range to be deleted.
<*number1*>	This is one number, indicating ID of address range.
name	This specifies name for address range to be deleted.
<*string*>	This is one string, indicating the name of address range.

Command Illustration:

To delete address range, it is available to delete the address range according to address range name, address range id or both. However, in case address range id and address range name are inconsistent, address range name shall apply.

When no parameter is given, the address range not quoted by policy is deleted.

Example:

To delete address range1, execute the following command:

```
ZXR10_FW.define# range delete name range1
```

5. This deletes all address ranges not quoted by policy.

Command	Function
ZXR10_FW.define#**range clean**	This deletes all address ranges not quoted by policy.

6. Showing all address ranges.

Command	Function
ZXR10_FW.define#**range show**	This shows all address ranges.

Setting Subnet Resource

1. Adding one subnet.

Command	Function
ZXR10_FW.define#**subnet add name** <*string1*> **ipaddr** <*ipaddress*> **mask** <*netmask*>[**except** <*string2*>][**session** <*number1*>]	This adds one subnet.

Parameter Description:

Parameter	Description
add	This adds one subnet.
name	This sets name for subnet.
<*string*>	This is one string, indicating name of the subnet.
ipaddr	This sets address for subnet.
<*ipaddress*>	This is one string, indicating ip address of subnet, such as 192.168.8.0.
mask	This sets subnet mask.
<*netmask*>	This is one string, indicating subnet mask, such as 255.255.255.0.
except	This sets except address in subnet.
<*string2*>	This is one string, indicating excepted IP address, in format of 0.0.0.0.
session	This sets the number of sessions.
<*number1*>	This is one number, indicating the number of sessions.

Command Illustration:

At the same moment, the number of connections of individual addresses within the subnet cannot exceed the number of max sessions.

Example:

To add subnet1 with subnet address to be 192.168.10.0 and mask to be 255.255.255.0, execute the following command:

```
ZXR10_FW.define# subnet add name subnet1
ipaddr 192.168.10.0 mask 255.255.255.0
```

2. Modifying one subnet.

Command	Function
ZXR10_FW.define#**subnet modify name** <*string1*>[**ipaddr** <*ipaddress*>][**mask** <*netmask*>][**except** <*string2*>][**session** <*number1*>]	This modifies one subnet.

Parameter Description:

Parameter	Description
modify	This modifies one subnet.
name	This specifies the name of subnet to be modified.
<*string*>	This is one string, indicating name of the subnet.
ipaddr	This sets new address for subnet.
<*ipaddress*>	This is one string, indicating ip address of subnet, such as 192.168.8.0.
mask	This sets new subnet mask.
<*netmask*>	This is one string, indicating subnet mask, such as 255.255.255.0.
except	This sets new except address in subnet.
<*string2*>	This is one string, indicating excepted IP address, in format of 0.0.0.0.
session	This sets new number of session.
<*number1*>	This is one number, indicating the number of sessions.

Command Illustration:

At the same moment, the total number of connections of all hosts within the subnet cannot exceed the number of max sessions.

Example:

To modify IP address of subnet1 to 192.168.20.0, execute the following command:

```
ZXR10_FW.define# subnet modify name
subnet1 ipaddr 192.168.20.0
```

3. Renaming one subnet.

Command	Function
ZXR10_FW.define#**subnet rename oldname** <*string1*> **newname** <*string2*>	This renames one subnet.

Parameter Description:

Parameter	Description
rename	This renames one subnet.
oldname	This specifies the name of subnet to be renamed.
<*string1*>	This is one string, indicating the name of subnet (the subnet name has been defined).
newname	This specifies new name for the subnet.
<*string2*>	This is one string, indicating new name of the subnet.

Command Illustration:

At the same moment, the total number of connections of all hosts within the subnet cannot exceed the number of max sessions.

Example:

To rename subnet1 to subnet2, execute the following command:

```
ZXR10_FW.define# subnet rename oldname
subnet1 newname subnet2
```

4. Deleting one subnet.

Command	Function
ZXR10_FW.define#**subnet delete** [**id** <*number1*>][**name** <*string*>]	This deletes one subnet.

Parameter Description:

Parameter	Description
delete	This deletes one subnet.
id	This specifies ID of the subnet to be deleted.
<*number*>	This is one number, indicating ID of subnet.
name	This specifies the name of subnet to be deleted.
<*string*>	This is one string, indicating name of the subnet.

Command Illustration:

To delete one subnet, it is available to delete the subnet according to subnet name, subnet id or both. However, in case subnet id and subnet name are inconsistent, subnet name shall apply.

When no parameter is given, the subnet not quoted by policy is deleted.

Example:

To delete subnet1, execute the following command:

```
ZXR10_FW.define# subnet delete name subnet1
```

5. This deletes all subnets not quoted by policy.

Command	Function
ZXR10_FW.define#**subnet clean**	This deletes all subnets not quoted by policy.

6. Showing all subnets.

Command	Function
ZXR10_FW.define#**subnet show**	This shows all subnets.

Setting Address Group

Different address resources can be combined to one address group to define policy destination or policy source. With address group, resource management is more flexible.

1. Adding one address group.

Command	Function
ZXR10_FW.define#**group_address add name** <*string1*>[**member** <*string2*>]	This adds one address group.

Parameter Description:

Parameter	Description
add	This adds one address group.
name	This sets name for address group.
<*string1*>	This is one string, indicating the name of address group.
member	This sets member in address group.
<*string2*>	This is one string, indicating address object, which can be defined host object, subnet object or address range object.

Command Illustration:

Before defining one address group, define address object. For details, please refer to related sections.

Example:

To add groupaddr1 and set defined host1 as the group member, execute the following command:

```
ZXR10_FW.define# group_address add name
groupaddr1 member host1
```

2. This adds member to defined address group.

Command	Function
ZXR10_FW.define#**group_address addmember groupname** <*string1*> **member** <*string2*>	This adds member to defined address group.

Parameter Description:

Parameter	Description
addmember	This adds member to address group.
name	This specifies name for address group to which member will be added.
<*string1*>	This is one string, indicating the name of address group.

Parameter	Description
member	This specifies members to be added.
<*string2*>	This is one string, indicating address object, which can be host object, subnet object or address range object.

Example:

```
ZXR10_FW.define # group_address addmember
groupame groupaddr1 member subnet1
```

3. Renaming address group.

Command	Function
ZXR10_FW.define#**group_address rename oldname** <*string1*> **newname** <*string2*>	This renames one subnet.

Parameter Description:

Parameter	Description
rename	This renames address group.
oldname	This specifies the name of address group to be renamed.
<*string1*>	This is one string, indicating the name of address group (the name of address group has been defined).
newname	This specifies new name for address group.
<*string2*>	This is one string, indicating the new name of address group.

Example:

This renames groupaddr1 to groupaddr2.

```
ZXR10_FW.define # group_address rename
oldname groupaddr1 newname groupaddr2
```

4. This deletes one address group.

Command	Function
ZXR10_FW.define#**group_address delete** [**id** <*number1*>][**name** <*string*>]	This deletes one address group.

Parameter Description:

Parameter	Description
delete	This deletes one address group.
id	This specifies ID of the address group to be deleted.
<*number1*>	This is one number, indicating ID of address group.

Parameter	Description
name	This specifies the name of address group to be deleted.
<*string*>	This is one string, indicating the name of address group.

Command Illustration:

To delete address group, it is available to delete the address group according to address group name, address group id or both. However, in case address group id and address group name are inconsistent, address group name shall apply.

When no parameter is given, all address groups not quoted by policy is deleted.

Example:

To delete groupaddr1, execute the following command:

```
ZXR10_FW.define# group_addrees delete name groupaddr1
```

5. This deletes one member in address group.

Command	Function
ZXR10_FW.define#**group_address delmember groupname** <*string1*> **member** <*string2*>	This deletes one member in address group.

Parameter Description:

Parameter	Description
delmember	This deletes one member in address group.
groupname	This specifies name for address group whose member is to be deleted.
<*string1*>	This is one string, indicating the name of address group.
member	This specifies the member to be deleted in the address group.
<*string2*>	This is one string, indicating address name, which can be host object, subnet object or address range object.

Example:

To delete member subnet1 in groupaddr1, execute the following command:

```
ZXR10_FW.define # group_address delmember
groupname groupaddr1 member subnet1
```

6. Deleting all address groups not quoted by policy.

Command	Function
ZXR10_FW.define#**group_address clean**	This deletes all address groups not quoted by policy.

7. Deleting all members in one address group.

Command	Function
ZXR10_FW.define#**group_address cleanmember groupname** <*string*>	This deletes all members in one address group.

Parameter Description:

Parameter	Description
cleanmember	This deletes all members in one address group.
groupname	This specifies the name of address group, all of whose members are to be deleted.
<*string*>	This is one string, indicating the name of address group.

Example:

To delete all members in groupaddr1, execute the following command:

```
ZXR10_FW.define # group_address cleanmember
groupame groupaddr1
```

8. Showing all address groups:

Command	Function
ZXR10_FW.define#**group_address show**	This shows all address groups:

Configuring Area Resource

Area Resource Configuration Overview

In FW area management, user can add, modify and delete one area and set default access privilege for area as well. Access control rule uses area for access control. In case no access control rule matches, ZXR10 8900 Series Switch FW service card will process this packet according to the privilege of area where destination interface locates.

Configuring Area Resource

1. Adding one area.

Command	Function
ZXR10_FW.define#**area add name** <*string1*>[**access** <*on*|*off*>][**attribute** <*string2*>][**comment** <*string3*>]	This adds one area.

Parameter Description:

Parameter	Description
add	This adds one area.
name	This specifies area name.
<*string1*>	This is a string, indicating the name of area.
access	This specifies privilege for accessing one area.
on	This permits accessing this area.
off	This denies accessing this area.
attribute	This specifies new attribute or attribute group bound to this area.
<*string2*>	This is one string, which can be one or more pre-defined attributes or attribute groups. As for multiple ones, single quotes are used, and space is used between each two (such as 'aa bb'); to view and define attribute or attribute group, perform the operations in attribute and attribute-group of network module.
comment	This sets comment.
<*string3*>	This s one string, indicating the content of comment.

Command Illustration:

Area is section of network space with similar security attribute. As for ZXR10 8900 Series Switch FW service card, access control rule uses area to control access.

Example:

To add area_gei_1/1 bound with attribute gei_1/1 and permit access to this area, execute the following command:

```
ZXR10_FW.define# area add name area_gei_1/1
access on attribute gei_1/1 comment
```

2. Modifying one area.

Command	Function
ZXR10_FW.define#**area modify name** <*string1*>[**access** <*on*|*off*>][**attribute** <*string2*>][**comment** <*string3*>]	This modifies one area.

Parameter Description:

Parameter	Description
modify	This modifies one area.
name	This specifies the name of area to be modified.

Parameter	Description
<*string1*>	This is a string, indicating the name of area.
access	This specifies new privilege for accessing one area.
on	This permits accessing this area.
off	This denies accessing this area.
attribute	This specifies new attribute or attribute group bound to this area.
<*string2*>	This is one string, which can be one or more pre-defined attributes or attribute groups. As for multiple ones, single quotes are used, and space is used between each two (such as 'aa bb'); to view and define attribute or attribute group, perform the operations in attribute and attribute-group of network module.
comment	This sets new comment. Tips: If the value "none" is input following parameter comment, it indicates deleting this comment.
<*string3*>	This s one string, indicating the content of comment.

Example:

To modify the privilege of accessing area_eth0 to off, execute the following command:

```
ZXR10_FW.define# area modify name area_gei_1/1 access off
```

3. Deleting one area.

Command	Function
ZXR10_FW.define#**area delete name** <*string*>	This deletes one area.

Parameter Description:

Parameter	Description
delete	This deletes one area.
name	This specifies the name of area to be deleted.
<*string*>	This is a string, indicating the name of area.

Example:

To delete area_gei_1/1, execute the following command:

```
ZXR10_FW.define#  area delete name area_gei_1/1
```

4. Renaming one area.

Command	Function
ZXR10_FW.define#**area rename oldname** <*string1*> **newname** <*string2*>	This renames one area.

Parameter Description:

Parameter	Description
rename	This renames one area.
oldname	This specifies the name of area to be renamed.
<*string1*>	This is a string, indicating the name of area.
newname	This specifies new area name.
<*string2*>	This is a string, indicating the name of area.

Example:

To rename area_gei_1/1 to firstarea, execute the following command:

```
ZXR10_FW.define# area rename oldame
area_gei_1/1 newname firstarea
```

5. Showing all areas in FW system.

Command	Function
ZXR10_FW.define#**area show**	This shows all areas in FW system.

6. Cleaning all unquoted areas in FW system.

Command	Function
ZXR10_FW.define#**area clean**	This cleans all unquoted areas in FW system.

Configuring Time Resource

Time Resource Configuration Overview

User can set time resource for using in access control rule, which can provides control with finer granularity. For example, user hopes to set different access control rules for working time and non-working time. With time resource, this problem can be solved easily.

According to using times, time resource can be classified into multi-time time resource and one-time time resource, where multi-time time indicates cycle time, such as a specific day in a week or a period in one day and one-time time indicates a certain period.

Configuring Week Cycle

1. Adding one week cycle.

Command	Function
ZXR10_FW.define#**schedule add name** <*string1*>[**cyctype** <*weekcyc* >][**week** <*string2*>][**start** <*string3*>][**end** <*string4*>]	This adds one week cycle. Week cycle indicates this object contains multiple uncontinuous regular period such as 9am to 5pm each Tuesday.

Parameter Description:

Parameter	Description
add	This adds one cycle.
name	This sets name of cycle.
<*string1*>	This is one string, indicating name of cycle.
cyctype	This sets type of cycle: weekcyc or yearcyc. The former indicates week cycle and the latter indicates year cycle.
<*weekcyc* >	This indicates week cycle, such as 8am to 8pm each Monday.
week	This sets which days are included in one week.
<*string2*>	This is one string , indicating one day in the week, in format of 1234567 (indicating from Monday to Sunday).
week-start	This sets the start time in one day.
<*string3*>	This is one string, indicating the start time, in format of HH:MM (hour:minute).
week-end	This sets the end time in each day. End time must be larger than start time.
<*string4*>	This is one string, indicating the end time, in format of HH:MM (hour:minute).

Command Illustration:

When specifying that multiple periods are contained in the week, with no separator among periods. For example, 12 indicates Monday and Tuesday.

Note:

24-hour time format is used for start time and end time in each day period. For example 10pm is expressed as 22:00.

Example:

To add week1 with period to be 10am to 18pm each Wednesday, execute the following command:

```
ZXR10_FW.define# schedule add name week1
cyctype weekcyc week 3 start 10:00 end 18:00
```

2. Modifying one week cycle.

Command	Function
ZXR10_FW.define#**schedule modify name** <*string1*>[**type** <*weekcyc* >][**week** <*string2*>][**start** <*string3*>][**end** <*string4*>]	This modifies one week cycle. Week cycle indicates this object contains multiple uncontinuous regular period such as 9am to 5pm each Tuesday.

Parameter Description:

Parameter	Description
modify	This modifies one cycle.
name	This specifies the name of cycle to be modified.
<*string1*>	This is one string, indicating name of cycle.
type	This specifies type of cycle to be modified: weekcyc or yearcyc. The former indicates week cycle and the latter indicates year cycle.
<*weekcyc* >	This indicates week cycle, such as 8am to 8pm each Monday.
week	This modifies which days of a week are included.
<*string2*>	This is one string , indicating one day in the week, in format of 1234567 (indicating from Monday to Sunday).
week-start	This modifies the start time in one day.
<*string3*>	This is one string, indicating the start time, in format of HH:MM (hour:minute).
week-end	This modifies the end time in each day. End time must be larger than start time.
<*string4*>	This is one string, indicating the end time, in format of HH:MM (hour:minute).

Example:

To modify the period of week1 to 9am to 16pm each Monday, execute the following command:

```
ZXR10_FW.define# schedule modify name week1
type weekcyc week 1 start 09:00 end 14:00
```

3. Renaming one cycle.

Command	Function
ZXR10_FW.define#**schedule rename oldname** <*string1*> **newname** <*string2*>	This renames one cycle.

Parameter Description:

Parameter	Description
rename	This renames one cycle.
oldname	This specifies the name of cycle to be renamed.
<*string1*>	This is one string, indicating the name of cycle (the cycle name has been defined).
newname	This specifies new name for the cycle.
<*string2*>	This is one string, indicating new name of cycle.

Example:

To rename week1 to week2, execute the following command:

```
ZXR10_FW.define# schedule rename oldname week1
newname week2
```

4. Deleting one cycle.

Command	Function
ZXR10_FW.define#**schedule delete** [**id** <*number1*>][**name** <*string*>]	This deletes one cycle.

Parameter Description:

Parameter	Description
delete	This deletes one cycle.
id	This specifies ID of the cycle to be deleted.
<*number1*>	This is one number, indicating ID of cycle.
name	This specifies the name of cycle to be deleted.
<*string*>	This is one string, indicating name of cycle.

Example:

To delete week1, execute the following command:

```
ZXR10_FW.define# schedule delete name week1
```

Configuring Year Cycle

1. Adding one year cycle

Command	Function
ZXR10_FW.define#**schedule add name** <*string1*>[**cyctype** <*yearcyc*>][**sdate** <*string2*>][**stime** <*string3*>][**edate** < *string4*>][**etime** < *string5*>]	This adds one year cycle, which indicates it only contains one period, such as from am 0 on January 1, 2007 to pm 23 on December 12, 2007.

Parameter Description:

Parameter	Description
add	This adds one cycle.
name	This sets name of cycle.
<*string1*>	This is one string, indicating name of cycle.
cyctype	This sets type of cycle: weekcyc or yearcyc. The former indicates week cycle and the latter indicates year cycle.
<*yearcyc* >	This indicates year cycle, such as from 0am in January 1, 2007 to 23pm in December 12, 2007.
sdate	This sets start date.
<*string2*>	This indicates start date , in format of YYYY-MM-DD (Year-Month-Day).
stime	This sets start time.
<*string3*>	This indicates the start time, in format of HH:MM:SS (hour:minute:second).
edate	This sets end date.
<*string4*>	This indicates end date , in format of YYYY-MM-DD (Year-Month-Day).
etime	This sets end time.
<*string5*>	This indicates the start time, in format of HH:MM:SS (hour:minute:second).

Example:

To add week1 with period to be 10am to 18pm each Wednesday, execute the following command:

```
ZXR10_FW.define# schedule add name week1
cyctype weekcyc week 3 start 10:00 end 18:00
```

2. Modifying one year cycle.

Command	Function
ZXR10_FW.define#**schedule modify name** <*string1*>[**type** <*yearcyc*>][**sdate** <*string2*>][**stime** <*string3*>][**edate**< *string4*>][**etime** < *string5*>]	This modifies one year cycle, which indicates it only contains one period, such as from am 0 on January 1, 2007 to pm 23 on December 12, 2007.

Parameter Description:

Parameter	Description
modify	This modifies one cycle.
name	This specifies the name of cycle to be modified.
<*string1*>	This is one string, indicating name of cycle.
type	This specifies type of cycle: weekcyc or yearcyc. The former indicates week cycle and the latter indicates year cycle.

Parameter	Description
<*yearcyc* >	This indicates year cycle, such as from 0am in January 1, 2007 to 23pm in December 12, 2007.
sdate	This sets new start date.
<*string2*>	This indicates start date , in format of YYYY-MM-DD (Year-Month-Day).
stime	This sets new start time.
<*string3*>	This indicates the start time, in format of HH:MM:SS (hour:minute:second).
edate	This sets new end date.
<*string4*>	This indicates end date , in format of YYYY-MM-DD (Year-Month-Day).
etime	This sets new end time.
<*string5*>	This indicates the start time, in format of HH:MM:SS (hour:minute:second).

Example:

To modify period of year1 to 0am in January 5, 2007 to 23pm in February 20, 2007, execute the following command:

```
ZXR10_FW.define# schedule modify name year1
type yearcyc sdate 2007-01-05 stime 00:00:00 edate2007-02-20
etime 23:00:00
```

3. Renaming one cycle.

Command	Function
ZXR10_FW.define#**schedule rename oldname** <*string1*> **newname** <*string2*>	This renames one cycle.

Parameter Description:

Parameter	Description
rename	This renames one cycle.
oldname	This specifies the name of cycle to be renamed.
<*string1*>	This is one string, indicating the name of cycle (the cycle name has been defined).
newname	This specifies new name for the cycle.
<*string2*>	This is one string, indicating new name of cycle.

Example:

To rename year1 to year2, execute the following command:

```
ZXR10_FW.define# schedule rename oldname
year1newname year2
```

4. Deleting one cycle.

Command	Function
ZXR10_FW.define#**schedule delete** [**id** <*number1*>][**name** <*string*>]	This deletes one cycle.

Parameter Description:

Parameter	Description
delete	This deletes one cycle.
id	This specifies ID of the cycle to be deleted.
<*number1*>	This is one number, indicating ID of cycle.
name	This specifies the name of cycle to be deleted.
<*string*>	This is one string, indicating name of cycle.

Example:

To delete year1, execute the following command:

```
ZXR10_FW.define# schedule delete name year1
```

Configuring Service Resource

Service Resource Configuration Overview

With setting of service resource, user can define access control rules according to different services. System defines some common services and user can customize services and port ids according to its demands or define various service combinations to service groups.

Showing System Defined Services

System has preset some common services for using when user sets access control rules. User can only view these preset services instead of modifying or deleting them.

1. Showing all services.

Command	Function
ZXR10_FW.define#**service show** [**custom**|**default**]	This shows all services.

Parameter Description:

Parameter	Description
custom|**default**	This specifies the type of service to be viewed, where keyword **custom** indicates the service is customized by user and **default** indicates the service is the default one in FW system (user doesn't need to customize it).

Configuring Customized Services

1. Adding one service.

Command	Function
ZXR10_FW.define#**service add name** <*string1*> **protocol** <*number1*> **port** <*number2*>[**port2** <*number3*>][**comment** <*string2*>]	This adds one service.

Parameter Description:

Parameter	Description
add	This adds one service.
name	This sets name for the service.
<*string1*>	This is one string, indicating name of customized service.
protocol	This sets L3 or L4 protocol number.
<*number1*>	This is one number, indicating protocol number.
port	This sets the start port, from which service is enabled. In case only one port is available, it only needs to set start port and doesn't need to set end port.
<*number2*>	This is one number, indicating id of start port.
port2	This sets end port of service.
<*number3*>	This is one number, indicating id of end port.
comment	This sets comment.
<*string1*>	This s one string, indicating the contents of comment.

Command Illustration:

Services are classified into default services provided by system and user customized services. As for default services, user cannot perform add, delete, modify and some other operations.

Example:

To add service http8080, set protocol number to 6 and port id to 8080, and set httpservice to be the content of comment, execute the following command:

```
ZXR10_FW.define# service add name http8080
protocol 6 port 8080 comment httpservice
```

2. Modifying one customized service.

Command	Function
ZXR10_FW.define#**service modify name** <*string1*>[**protocol** <*number1*>][**port** <*number2*>][**port2** <*number3*>][**comment** <*string2*>]	This modifies one customized service.

Parameter Description:

Parameter	Description
modify	This modifies one service.
name	This specifies the name of service to be modified.
<*string1*>	This is one string, indicating name of customized service.
protocol	This sets L3 or L4 protocol number.
<*number1*>	This is one number, indicating protocol number.
port	This modifies the start port, from which service is enabled. In case only one port is available, it only needs to set start port and doesn't need to set end port.
<*number2*>	This is one number, indicating id of start port.
port2	This modifies end port of service.
<*number3*>	This is one number, indicating id of end port.
comment	This modifies content of comment.
<*string1*>	This s one string, indicating the contents of comment.

Command Illustration:

Services are classified into default services provided by system and user customized services. As for default services, user cannot perform add, delete, modify and some other operations.

Example:

To modify port id of service http8080 from 8000 to 8080 and modify protocol number to 4, execute the following command:

```
ZXR10_FW.define# service modify name http8080
port 8000 port2 8008 protocol 4
```

3. Renaming customized service.

Command	Function
ZXR10_FW.define#**service rename oldname** <*string1*> **newname** <*string2*>	This renames one customized service.

Parameter Description:

Parameter	Description
rename	This renames one service.
oldname	This specifies the name of service to be renamed.
<*string1*>	This is one string, indicating the name of service (the service name has been defined).
newname	This specifies new name for one service.
<*string2*>	This is one string, indicating new name of the service.

Command Illustration:

Services are classified into default services provided by system and user customized services. As for default services, user cannot perform add, delete, modify and some other operations.

Example:

To rename service http8080 to http8000, execute the following command:

```
ZXR10_FW.define# service rename oldname
http8080 newname http8000
```

4. Deleting one customized service.

Command	Function
ZXR10_FW.define#**service delete** [**id** <*number1*>][**name** <*string*>]	This deletes one customized service.

Parameter Description:

Parameter	Description
delete	This deletes one service.
id	This specifies ID of the service to be deleted.
<*number1*>	This is one number, indicating ID of service.
name	This specifies the name of service to be deleted.
<*string*>	This is one string, indicating name of the service.

Command Illustration:

To delete one service, it is available to delete it according to service name, id or both. In case the two are inconsistent, service name shall apply.

When no parameter is given, the service not quoted by policy is deleted.

Example:

To delete service http8000, execute the following command:

```
ZXR10_FW.define# service delete name http8000
```

5. Deleting all customized services not quoted by policy.

Command	Function
ZXR10_FW.define#**service clean**	This deletes all customized services not quoted by policy.

6. Showing all services.

Command	Function
ZXR10_FW.define#**service show** [**custom**|**default**]	This shows all services.

Parameter Description:

Parameter	Description
custom|**default**	This specifies the type of service to be viewed, where keyword **custom** indicates the service is customized by user and **default** indicates the service is the default one in FW system (user doesn't need to customize it).

Configuring Server Group

User can combine a few services into one group, which can be used when setting access control.

1. Adding one service group.

Command	Function
ZXR10_FW.define#**group_service add name** <*string1*>[**member** <*string2*>]	This adds one service group.

Parameter Description:

Parameter	Description
add	This adds one service group.
name	This sets name for the service group.
<*string1*>[	This is one string, indicating the name of service group.

Parameter	Description
member	This sets members (services) in service group. The services can be customized by user or default ones in system.
<*string2*>[	This is a string, indicating the name of service.

Command Illustration:

Before defining service group, define services. For details, please refer to section Showing System Defined Services.

Note:

Both system default services and user customized services can be included in one service group.

2. Adding service to existing service group.

Command	Function
ZXR10_FW.define#**group_service add name** <*string1*>[**member** <*string2*>]	This adds service to existing service group.

Parameter Description:

Parameter	Description
addmember	This adds service to service group.
groupname	This sets name of service group, to which service will be added.
<*string1*>	This is a string, indicating the name of service group.
member	This sets services to be added. The services can be customized by user or default one in system.
<*string2*>	This is a string, indicating the name of service.

Command Illustration:

Before defining service group, define services. For details, please refer to section Configuring Customized Services.

3. Renaming service group.

Command	Function
ZXR10_FW.define#**group_service rename oldname** <*string1*> **newname** <*string2*>	This renames service group.

Parameter Description:

Parameter	Description
rename	This renames service group.

Parameter	Description
oldname	This specifies the name of service group to be renamed.
<*string1*>	This is one string, indicating the name of service group (the service group name has been defined).
newname	This specifies new name for one service group.
<*string1*>	This is one string, indicating new name of service group.

4. Deleting service member in service group.

Command	Function
`ZXR10_FW.define#`**group_service delmember groupname** <*string1*> **member** <*string2*>	This deletes service member in service group.

Parameter Description:

Parameter	Description
delmember	This deletes service member in service group.
groupname	This specifies the name of service group, whose service member is to be deleted.
<*string1*>	This is one string, indicating the name of service group.
member	This specifies the service member to be deleted in the service group.
<*string2*>	This is a string, indicating the name of service.

Chapter 4

ZXR10 FW Function Management

Table of Contents:

ZXR10 FW Function Management Overview

Management to most functions of ZXR10 8900 Series Switch FW service card is based on VLAN and implemented on main board with command lines. The following FW-related confutations are available on main board:

- Entering and exiting from FW configuration mode
- Creating and deleting fw-template
- Binding management IP
- Configuring flow recovery
- Binding slot number
- Configuring nat ip
- Configuring session
- Binding FW template for specific vlan
- Viewing management configuration
- Configuring vlan

Configuring ZXR10 FW

Accessing and Exiting FW Configuration Mode

This topic describes how to access and exit FW configuration node.

1. Entering FW configuration mode (used in configure terminal mode)

Command	Function
ZXR10(config)#**fw**	This enters FW configuration mode.

2. Exiting FW configuration mode (used in FW configuration mode)

Command	Function
ZXR10(config-fw)#**exit**	This exit FW configuration mode.

Creating and Deleting FW-Template Mode

This topic describes how to create and delete fw-template on main board.

1. This creates fw-template (used in fw configuration mode).

Command	Function
ZXR10(config-fw)#**fw-template**< *template-id* >	This creates fw-template.

Parameter Description:

Parameter	Description
<*template-id* >	This is id of fw-template, in range of 1-127.

2. Showing configuration state of fw-template (used in any mode of main board).

Command	Function
ZXR10(config-fw)#**show fw-template** <*template-id*>	This shows if the configuration is successful.

3. Deleting fw-template configuration node (used in fw configuration mode).

Command	Function
ZXR10(config-fw)#**no fw-template**< *template-id* >	This deletes the specific fw-template.

Note:

FW-template ranges from 1 to 127.

Example

1. To add one template whose id is 100 (or to enter the configuration mode of existing fw-template whose id is 100), execute the following command.

   ```
   ZXR10(config-fw)#fw-template 100
   ```

2. To delete one template whose id is 100 (or to enter the configuration mode of existing fw-template whose id is 100), execute the following command.

   ```
   ZXR10(config-fw)#no fw-template 100
   ```

Binding Management IP

This topic describes how to bind and delete management IP on FW.

Step	Command	Function
1	ZXR10(config)#**interface vlan** <*vlanid*>	This enters L3 interface configuration mode.
2	ZXR10(config-if-vlan10)#**bind fw-template 127**	This binds fw for L3 interface (take L3 interface in vlan10 for example here)
3	ZXR10(config-if-vlan10)#**ip address** <*ipaddress*><*ipmask*>	This configures IP address for L3 interface (take L3 interface in vlan10 for example here)
4	ZXR10(config)#**fw**	This enters FW configuration mode.
5	ZXR10(config-fw)#**bind mng-ip**< *ipaddress* >	This binds management IP with FW.
6	ZXR10(config-fw)#**show mng-ip**	This shows management IP of FW.
7	ZXR10(config-fw)#**no bind mng-ip**	This deletes management IP of FW.

Example To bind one management IP with FW, execute the following commands:

```
ZXR10(config)#fw ZXR10(config-fw)#fw-template 7
ZXR10(config-fw-templete-7)#bind slot 8
ZXR10(config)#interface vlan 10
ZXR10(config-if-vlan10)#bind fw-template  7
ZXR10(config-if-vlan10)#ip addr 1.2.3.4 255.255.255.0
ZXR10(config-if-vlan10)#exit ZXR10(config)#inter gei_1/2
ZXR10(config-gei_1/2)#switchport mode access
ZXR10(config-gei_1/2)#switchport access vlan 10
ZXR10(config- gei_1/2)#exit ZXR10(config)#fw
ZXR10(config-fw)#bind mng-ip 8 1.2.3.4
(assume that FW is in slot 8 currently)
```

Where joining one port (gei_1/2) in link state to vlan10 is to enable protocol of vlan 10 to up state. Only in this case, can this L3 interface IP be bound to FW.

Configuring Flow Recovery

This topic how to configure FW flow recovery.

Steps	Command	Function
1	`ZXR10(config)#`**fw**	This enters FW configuration mode (used in configure terminal mode)
2	`ZXR10(config-fw)#`**fw-template**< *template-id* >	This accesses fw-template node (used in fw configuration mode).
3	`ZXR10(config-fw-template-1)#`**flow-recovery enable**	This enables flow recovery function (take fw-template 1 for example to configure flow recovery).
4	`ZXR10(config-fw-template-1)#`**flow-recovery disable**	This disables flow recovery function (used in fw-template configuration mode).

Binding Slot Number

This topic describes how to bind FW template with slot number.

Steps	Command	Function
1	ZXR10(config)#**fw**	This enters FW configuration mode (used in configure terminal mode)
2	ZXR10(config-fw)#**fw-template**< *template-id* >	This accesses fw-template node (used in fw configuration mode).
3	ZXR10(config-fw-template-1)#**bind**< *slot-number* >	This binds slot number (used in fw-template configuration mode).
4	ZXR10(config-fw)#**show fw-template** <*template-id*>	This shows if fw-template is successfully configured (used in any mode of main board).

Example To bind slot 8 with fw-template10 (fw is inserted in slot 8), execute the following command:

```
ZXR10(config)#fw ZXR10(config-fw)#fw-template 10
ZXR10(config-fw-templete-10)#bind slot 8
```

Configuring NAT IP

This topic describes how to specify IP POOL for NAT function.

Step	Command	Function
1	ZXR10(config)#**fw**	This enters FW configuration mode (used in configure terminal mode)
2	ZXR10(config-fw)#**fw-template**< *template-id* >	This accesses fw-template node (used in fw configuration mode).
3	ZXR10(config-fw-template-1)#**nat dip**< *ipaddr* >< *ipmask* >	This specifies one destination nat address (used in fw-template configuration mode).
4	ZXR10(config-fw-template-1)#**nat sip**< *ipaddr* >< *ipmask* >	This specifies one source nat address (used in fw-template configuration mode).

Note:

Masks used in steps 3 and 4 are inverse masks.

Example To specify source mac address of fw-template 7 to 10.1.1.1 255.255.0.0, execute the following commands:

```
ZXR10(config)#fw ZXR10(config-fw)#fw-template 7
ZXR10(config-fw-templete-7)#bind slot 8
ZXR10(config-fw -templete-7)#nat sip  10.1.1.1   0.0.255.255
```

Configuring Session

This topic describes how to log in FW with command **session**.

Step	Command	Function
1	`ZXR10(config)#`**fw**	This enters FW configuration mode (used under configure terminal node)
2	`ZXR10(config-fw)#`**fw-template**< *template-id* >	This accesses fw-template node (used in fw configuration mode).
3	`ZXR10(config-fw-template-1)#`**session**	This executes session (used in fw-template configuration mode).

Example To configuring session, execute the following commands:

```
ZXR10(config)#fw ZXR10(config-fw)#fw-template 7
ZXR10(config-fw-templete-7)#bind slot 8
ZXR10(config-fw -templete-7)#session
```

Binding FW Template for Specific VLAN

This topic describes how to bind FW template for specific VLAN.

Step	Command	Function
1	`ZXR10(config)#`**fw**	This enters FW configuration mode (used in configure terminal mode)

Step	Command	Function
2	ZXR10(config-fw)#**fw-template**< *template-id* >	This accesses fw-template node (used in fw configuration mode).
3	ZXR10(config)#**interface vlan**< *vlan-id* >	This enters L3 interface configuration mode (used under configure terminal node)
4	ZXR10(config-if-vlan10)#**bind fw-template** < *template-id* >	This binds fw-template (used under if-vlan node).
5	ZXR10(config)#**show fw-vlan-binding**	This shows if vlan succeeds in binding with FW (it can be used in any mode of main board.)

Example To bind fw-template1 with vlan10 and bind slot 7 with fw-template1, execute the following commands:

```
ZXR10(config)#fw ZXR10(config-fw)#fw-template 1
ZXR10(config-fw- template 1)#bind slot 7
ZXR10(config)#interface vlan 10
ZXR10(config-if-vlan10)#bind fw-template 1
```

Viewing Management Configuration

This topic shows how to view binding among management IP, fw-template and vlan.

Step	Command	Function
1	ZXR10(config)#**show mng-ip**	This shows FW management IP (used in any mode of main board).
2	ZXR10(config)#**show fw-template**<*template-id*>	This shows fw-template information (used in any mode of main board).
3	ZXR10(config)#**show fw-vlan-binding**	This shows binding between vlan and fw-template (used in any mode of main board).

Configuring VLAN

This topic describes how to configure VLAN.

1. Creating specific VLAN and entering VLAN configuration mode (under configure terminal node)

Command	Function
ZXR10(config)#**vlan** {<*vlan-id*>}	When this vlan doesn't exist, a vlan whose id is < vlan-id > is generated and this enters corresponding vlan configuration mode. When this vlan exist, this enters corresponding vlan configuration mode.

2. Deleting specific VLAN (under configure terminal node)

Command	Function
ZXR10(config)#**no vlan** {<*vlan-id*>}	This deletes vlan whose id is < vlan-id >.

3. Creating VLANs in batch (under configure terminal node)

Command	Function
ZXR10(config)#**vlan-list** <*vlanid-range*>	This creates vlan in batch.

Parameter Description:

Parameter	Description
<*vlanid-range*>	Vlanid-range is from id1 to idN and the max value of vlan id is 4094.

4. Deleting VLANs in batch (under configure terminal node)

Command	Function
ZXR10(config)#**no vlan-list** <*vlanid-range*>	This deletes all VLANs in batch with VLAN ids belong to range < vlanid-range >.

5. Setting VLAN link type on Ethernet interface (under L2 interface configuration mode)

Command	Function
ZXR10(config-gei_1/2)#**switchport mode** {**access**|**trunk**|**hybrid**}	This sets VLAN link type on Ethernet interface.

Command Illustration:

ZXR10 8900 Series SwitchThere are three VLAN link types for Ethernet interface: Access mode, Trunk mode and Hybrid mode. Access mode is used by default.

- The port connecting with access link can only belong to one VLAN. It shall be untagged and is used to connect PC in usual cases.
- The port connecting with trunk link can belong to multiple vlans. It must be tagged, can receive and send packets of multiple vlans, and is used to connect two switches in usual cases.
- The port connecting with hybrid link can belong to multiple vlans. User can customize whether to attach tag to the packet on the port. It can receive and send packets of multiple vlans and can be used to connect two switches or to connect pc.

6. Adding an Ethernet interface into a designated vlan (under L2 interface configuration mode)

Command	Function
ZXR10(config-gei_1/2)#**switchport access**|**trunk**|**hybrid vlan** {<*vlan-id*>|<*vlan-range* >}	This adds one Ethernet interface into the designated VLAN;

Command Illustration:

- The port connecting with access link can only belong to one VLAN. It shall be untagged.
- The port connecting with trunk link can belong to multiple vlans.
- The port connecting with hybrid link can belong to multiple vlans.

7. Setting native VLAN on Ethernet interface (under L2 interface configuration mode)

Command	Function
ZXR10(config-gei_1/2)#**switchport trunk**|**hybrid native vlan** {<*vlan-id*>|<*vlan-name*>}	This sets native vlan on Ethernet interface.

Command Illustration:

Trunk port and hybrid port belong to multiple vlans and they need to set native vlan. If native vlan is set on port, when one frame with no vlan tag is received on port, it will be forwarded to the port belonging to this native vlan. Native vlan of trunk port and hybrid port is vlan 1 by default.

8. Creating VLAN L3 interface

Command	Function
ZXR10(config)#**interface** {**vlan** <*vlan-id*>|<*vlan-if*>}	This creates VLAN L3 interface.

Command Illustration:

To create VLAN L3 interface, it is necessary to create this VLAN firstly.

9. Enabling/Disabling VLAN L3 interface (used under if-vlan node)

Command	Function
ZXR10(config-if-vlan10)#**shutdown/ no shutdown**	This enables/disables VLAN L3 interface.

Command Illustration:

To enable/disable VLAN L3 interface is just to enable/disable VLAN L3 forwarding function and it doesn't influence member ports of this vlan. When all Ethernet interfaces under VLAN interface are down, vlan interface is down by default; when one or more Ethernet interfaces under one VLAN interface are up, the vlan interface is up. VLAN interface in up state can be disabled by force.

Chapter 5

Packet Filtering and Access Control Rule Configuration

Table of Contents:

Configuring Packet Filtering Policy

Packet Filtering Overview

By reading this chapter, user can learn how to control data flow by setting packet block policy and access control rule.

This chapter has the following content:

- Packet block policy: This part describes how to control L2/3 access by setting packet block policy.
- Access control rule: This part describes how to control L3-L7 access by setting access control rule.

Configuring Packet Filtering Policy

This topic describes basic configuration commands and configuration examples of packet filtering policy.

Commands introduced in this topic are used to set packet filtering rule, control access of received IP packets, filter illegal packets or those denied by rules, and provide protection in the case that GW system doesn't join GW module.

To access this command module, execute the following command:

#**pf**

To exit from this command module, execute the following command:

#**exit**

1. Setting default packet filtering rule.

Command	Function
ZXR10_FW.pf #**rule set default action** <*accept*|*reject*> **log**<*yes*|*no*>	This sets default packet filtering rule.

Parameter Description:

Parameter	Description
accept|*reject*	permit|deny
log	This specifies whether to record it in the log.
yes|*no*	Yes|No

Example:

To set default packet filtering rule to permit and not record it into log, execute the following command:

```
ZXR10_FW.pf # rule set default action accept log no
```

2. Adding one IP packet filtering rule.

Command	Function
ZXR10_FW.pf #**rule add action** <*accept*|*reject*> **l2protocol** <*ip*|*0800*>[**area** <*string1*>][**log** <*yes*|*no*>][**smac** <*string2*>][**dmac** <*string3*>][**l3protocol** <*all*|*0*|*tcp*|*6*|*udp*|*17*|*icmp*|*1*|*igmp*|*2*|*number*>][**sip** <*string4*>][**dip** <*string5*>][**sport** <*number1*>][**dport** <*number2*>][**sport_end** <*number3*>][**dport_end** <*number4*>]	This adds one IP packet filtering rule.

Parameter Description:

Parameter	Description
add	This adds one packet filtering rule.
action	This is the action to packet meeting rules: permit or deny
accept|*reject*	permit|deny
l2protocol	This is the L2 protocol type used by packet.
ip|*0800*	IP protocol|IP protocol number
area	This specifies area resource.

Parameter	Description
<*string1*>	This is one string, which must be one predefined area resource.
log	This specifies whether to record it into log.
yes|*no*	Yes|No
smac	This sets source mac address.
<*string2*>	This is one standard mac address string.
dmac	This sets destination mac address.
<*string3*>	This is one standard mac address string.
l3protocol	This is the L3 protocol type used by packet.
all|*0*|*tcp*|*6*|*udp*|*17*|*icmp*|*1*|*igmp*|*2*|*number*	All protocols/all protocols/TCP protocol/TCP protocol/UDP protocol/UDP protocol/ICMP protocol/ICMP protocol/IGMP protocol/User inputs specified protocol number
sip	This specifies source address, which must be one predefined address.
<*string4*>	This is one string.
dip	This specifies destination address, which must be one predefined address.
<*string5*>	This is one string.
sport	This specifies source start port id.
<*number1*>	This is one number.
sport_end	This specifies source end port id.
<*number2*>	This is one number.
dport	This specifies destination start port id.
<*number3*>	This is one number.
dport_end	This specifies destination end port id.
<*number4*>	This is one number.

Example:

To permit the device whose source MAC address is 00:50:04:C3:B0:31 to access the device whose destination ip is doc_server and destination port id is 8000, execute the following command, where doc_server is predefined address.

```
ZXR10_FW.pf # rule add action accept smac
00:50:04:C3:B0:31 dip doc_server dport 8000
```

3. Adding one ARP/RARP/IPX packet filtering rule.

Command	Function
ZXR10_FW.pf #**rule add action** <*accept*|*reject*> **l2protocol** <*arp*|*0806*|*rarp*|*8035*|*ipx*|*8137*>[**log**<*yes*|*no*>][**area** <*string*>][**smac** < *string 2*>][**dmac** < *string 3*>]	This adds one ARP/RARP/IPX packet filtering rule.

Parameter Description:

Parameter	Description
add	This adds one packet filtering rule.
action	This is the action to packet meeting rules: permit or deny
accept|*reject*	permit|deny
l2protocol	This is the L2 protocol type used by packet.
arp|*0806*|*rarp*|*8035*|*ipx*|*8137*	ARP| ARP protocol number|RARP| RARP protocol number|IPX|IPX protocol number
log	This specifies whether to record it into log.
yes|*no*	Yes|No
area	This specifies area resource.
<*string1*>	This is one string, which must be one predefined area resource.
smac	This sets source mac address.
<*string2*>	This is one standard mac address string.
dmac	This sets destination mac address.
<*string3*>	This is one standard mac address string.

Example:

To add arp accept rule to area_eth1 and record it into log, execute the following command:

```
ZXR10_FW.pf # rule add action accept l2protocol
arp  smac 00:50:04:C3:B0:31
```

4. Adding one L3 protocol packet filtering rule.

Command	Function
ZXR10_FW.pf #**rule add action** <*accept*|*reject*> **l3protocol** <*number*>[**log**<*yes*|*no*>][**area** <*string1*>][**smac** < *string2* >][**dmac** < *string3*>]	This adds one L3 protocol packet filtering rule.

Parameter Description:

Parameter	Description
add	This adds one packet filtering rule.
action	This is the action to packet meeting rules: permit or deny
accept|*reject*	permit|deny
l3protocol	This is the L3 protocol type used by packet.
<*number*>	This is the protocol number.
log	This specifies whether to record it into log.
yes|*no*	Yes|No

Parameter	Description
area	This specifies area resource.
<*string1*>	This is one string, which must be one predefined area resource.
smac	This sets source mac address.
<*string2*>	This is one standard mac address string.
dmac	This sets destination mac address.
<*string3*>	This is one standard mac address string.

Command Illustration:

To reject TCP packets passing through and not record it into log, execute the following command:

```
ZXR10_FW.pf # rule add action reject l3protocol 6 log no
```

5. Modifying one IP packet filtering rule.

Command	Function
ZXR10_FW.pf #**rule modify id** < *numbe1*>[**action** <*accept*| *reject* >][**l2protocol** <*ip*|*0800*>][**area** <*string1*>][**log** <*yes*|*no*>][**smac** <*string2*>][**dmac** <*string3*>][**l3protocol** <*all*|*0*|*tcp*|*6*|**udp**|*17*|*icmp*|*1*|*igmp*|*2*| *number*>][**sip** <*string4*>][**dip** <*string5*>][**sport** <*number2*>][**dport** <*number3*>][**sport_end** <*numbe4*>][**dport_end** <*number5*>]	This modifies one IP packet filtering rule.

Parameter Description:

Parameter	Description
modify	This modifies one packet filtering rule.
id	This is rule id.
< *numbe1*>	This is one number.
action	This is the action to packet meeting rules: permit or deny.
accept|*reject*	permit|deny
l2protocol	This is the L2 protocol type used by packet.
ip|*0800*	IP protocol|IP protocol number
area	This specifies area resource.
<*string1*>	This is one string, which must be one predefined area resource.
log	This specifies whether to record it into log.
yes|*no*	Yes|No

Parameter	Description
smac	This sets source mac address.
<string2>	This is one standard mac address string.
dmac	This sets destination mac address.
<string3>	This is one standard mac address string.
l3protocol	This is the L3 protocol type used by packet.
all|0|tcp|6|udp|17|icmp|1|igmp|2|number	All protocols/all protocols/TCP protocol/TCP protocol number/UDP protocol/UDP protocol number/ICMP protocol/ICMP protocol number/IGMP protocol/IGMP protocol number/User inputs specified protocol number
sip	This specifies source address, which must be one predefined address.
<string4>	This is one string.
dip	This specifies destination address, which must be one predefined address.
<string5>	This is one string.
sport	This specifies source start port id.
<number2>	This is one number.
sport_end	This specifies source end port id.
<number3>	This is one number.
dport	This specifies destination start port id.
<number4>	This is one number.
dport_end	This specifies destination end port id.
<number5>	This is one number.

Command Illustration:

As for execution of policy, "First match" principle is adopted, where policy sequence is related to policy logic. After adding one policy, by moving the position of packet filtering rule, policy execution sequence can be changed.

Example:

To modify the rule whose id is 8054 and permit the device whose source MAC address is 00:50:04:C3:B0:31 to access the device whose destination ip is doc_server and destination port id is 8080, execute the following command,

```
ZXR10_FW.pf # rule modify id 8054 action reject
smac 00:50:04:C3:B0:31 dip doc_server dport 8080
```

6. Clearing all packet filtering rules.

Command	Function
`ZXR10_FW.pf #`**rule clean**	This clears all packet filtering rules.

7. Deleting one packet filtering rule.

Command	Function
ZXR10_FW.pf #**rule delete id** <*number*>	This deletes one packet filtering rule.

Parameter Description:

Parameter	Description
<*number*>	This is rule id.

8. Viewing packet filtering rule.

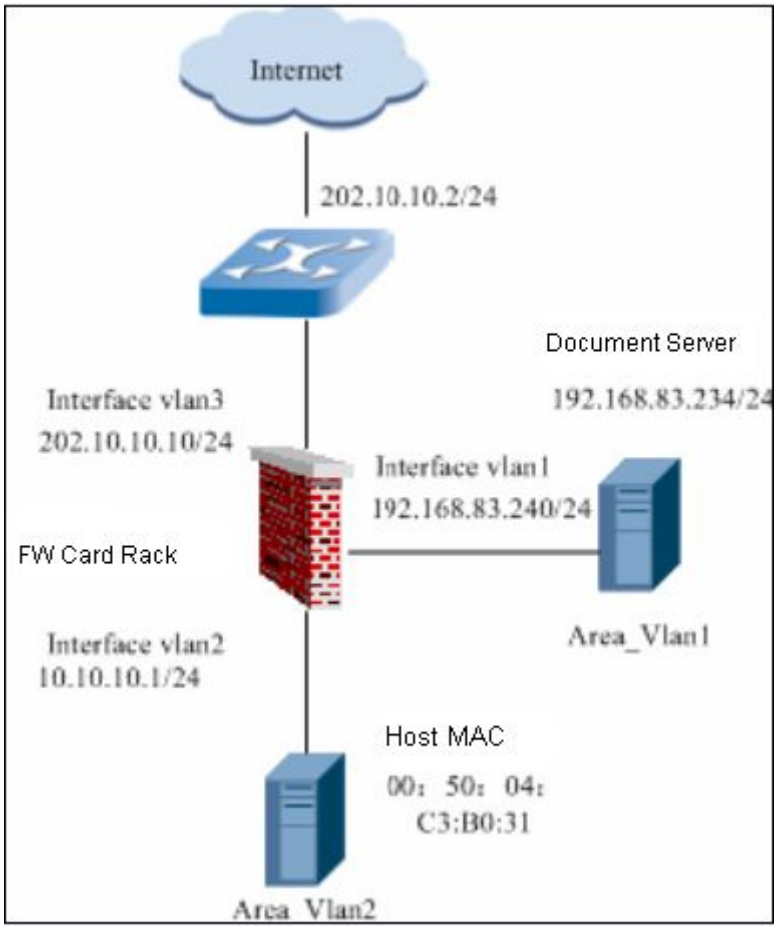
Command	Function
ZXR10_FW.pf #**rule show**	This shows packet filtering rule.

Packet Filtering Policy Configuration Example

Packet Filtering Policy Configuration Example One

The most basic function of FW is to control communication between intranet and extranet. MAC address can be used to directly identify one network device. FW card supports to filter packets based on MAC address. Only packets whose MAC address meets packet filtering rule can pass through FW and access destination area. With MAC address filtering technology, only authorized MAC address can access network resources. In the Figure 1, only forbid the host in Area_Vlan2 and whose MAC address is 00:50:04:C3:B0:31 to access document server (port 8000 in 192.168.83.234/24) in Area_Vlan1.

FIGURE 1 PACKET FILTERING CONFIGURATION EXAMPLE



Configuration Points:

- Specifying server host address;
- Configuring default packet block policy
- Configuring packet block policy;

1. To configure default packet block policy——permit any packets to pass through FW, execute the following command:

```
ZXR10_FW #pf rule set default action accept
log yes
```

2. To add host doc_server, execute the following command:

```
ZXR10_FW #define host add name doc_server
ipaddr 192.168.83.234
```

3. To add area_vlan3 (setting gei_2/3 to be in vlan 3; it is the interface connecting to router), execute the following command:

```
ZXR10_FW #define area add name area_vlan3
access off attribute gei_2/3
```

4. To configure packet filtering rule: forbid the host whose MAC address is 00:50:04:C3:B0:31 to access document server, execute the following command:

```
ZXR10_FW #pf rule add smac 00:50:04:C3:B0:31
area area_vlan2 dip doc_server dport 8000 action reject
```

Notes:

- Make sure that area attribute and access control rule specify to forbid this host to access document sever.
- Hexadecimal digit letter in MAC address shall be upper case. For example, if MAC address 00:50:04:C3:B0:31 is wrongly input to 00:50:04:c3:b0:31, system will prompt error message.
- If source MAC address and source IP address are input when defining packet filtering policy, this rule takes effect only when both MAC address and IP address of host match this condition.
- To forbid accessing some ports of destination host, set range on destination port if port ids are continuous, or it needs to set corresponding packet block policy to each port. If there is only one port, it is ok to only input start port id or set the same value for start and end port ids. In this example, it is set to 8000-8000.
- It is not recommended to input MAC address. If destination MAC address needs to be input, it must be MAC address of FW (corresponding to this area) physical interface and cannot be other values.

Packet Filtering Policy Configuration Example Two

Disable port 8000 in 192.168.83.234 to 10.10.10.0/24. That's to say, users of all network segments except for 10.10.10.0/24 can access port 800 in 192.168.83.234.

Configuration Points:

- Defining server host address resource and subnet address resource;
- Configuring default packet block policy;
- Configuring packet block policy;

1. Configuring default packet block policy——permit any packets to pass through FW.

```
ZXR10_FW #pf rule set default action
accept log yes
```

2. Adding host address resource doc_server.

```
ZXR10_FW #define host add name doc_server
ipaddr 192.168.83.234
```

3. Adding subnet address resource market department.

```
ZXR10_FW #define subnet add name market
ipaddr 10.10.10.0 mask 255.255.255.0
```

4. Adding packet block policy and forbidding market accessing port 8000 of document server.

```
 ZXR10_FW #pf rule add sip market dip
doc_server dport 8000 action reject
```

Configuring Access Control Rules

Access Control Rule Overview

As for access control rule, FW card permits or denies the packets matching access control rule to pass through.

After receiving one packet, FW will match it with all rules in ACL sequentially. Once matched rule is found, FW processes this packet according to operation (permit or drop) specified by this policy and not check default area attribute. In case matched access rule is unavailable, FW card will process this packet according to default attribute (permit or deny) of the area where destination interface locates.

Before querying access control rule, FW card will query if the packet matches destination address translation rule. If the packet matches destination address translation rule, FW card will translate destination IP address of received packet to preset IP address (actual IP address in usual cases). Therefore, when setting access control rule, system uses actual source and destination addresses (destination address after translation) to set access rule; meanwhile, system supports to set access rule according to destination address before translation. In this case, packet will match access control rule according to destination address before translation.

By defining access control rule, that is defining match rule of packet, FW card can identify and match packet from various aspects such as area, VLAN, address, user, connection and time. Source and destination of access control rule can be preset VLAN/area or one or more address resources and user group resources.

Configuring Access Control Rule

This topic describes configuration commands and configuration examples of access control rule.

User can control L3-L7 access flexibly and powerfully by setting access control rule. FW card can identify and match packet from various aspects such as area, VLAN, address, user, connection and time. What's more, FW card can perform deep data detection and filtering for various application layer protocols. Similar to packet filtering policy, packet matches access control rules sequentially. However, there is no default rule for access control rule. That's to say, if no Deny All rule is attached to the end of ACL, system will process this packet according to default attribute (permit or deny) of area where destination interface locates.

To access this command module, execute the following command:

#**firewall**

To exit from this command module, execute the following command:

#**end**

1. Adding one access control rule.

Command	Function
ZXR10_FW.firewall #**policy add action** <*accept*|*deny*>[**srcarea** <*string1*>][**dstarea** <*string2*>][**srcvlan** <*string3*>][**dstvlan** <*string4*>][**src** <*string5*>][**dst** <*string6*>][**service** <*string7*>][**schedule** <*string8*>][**sport** <*string9*>][**orig_dst** <*string10*>][**dpi** <*string11*>][**ar**<*string12*>][**av**<*on*|*off*>][**permanent** <*on*|*off*>][**log** <*on*|*off*|*alarm*>][**enable**<*yes*|*no*>][**before** <*number1*>]	This adds one access control rule.

Parameter Description:

Parameter	Description
add	This adds one FW access control rule.
action	This sets access privilege, that is to permit or deny packets matching this rule to pass through FW.
accept|*deny*	permit/deny
srcarea	This sets source area.
<*string1*>	This is one string. It must be one or more preset area name(s). As for multiple area names, space is used between each two area names and all addresses are quoted with single quotes, such as 'area_gei_5/1'.
dstarea	This sets destination area.
<*string2*>	This is one string. It must be one or more preset area name(s). As for multiple area names, space is used between each two area names and all addresses are quoted with single quotes, such as 'area_gei_5/1'.
srcvlan	This sets source VLAN.
<*string3*>	This is one string, indicating preset vlan number.
dstvlan	This sets destination VLAN.
<*string4*>	This is one string, indicating preset vlan number.
src	This is source address.
<*string5*>	This is one string, indicating preset address name. Multiple address names can be input and space is used between each two address names and all address names are quoted with single quotes, such as 'aa ll'.
dst	This is destination address.

Parameter	Description
<*string6*>	This is one string, indicating preset address name. Multiple address names can be input and space is used between each two address names and all address names are quoted with single quotes, such as 'aa ll'.
service	This sets service resource.
<*string7*>	This is one string. It must be one or more names of system default services or customized services. As for multiple service names, space is used between each two service names and all service names are quoted with single quotes, such as 'IP ICMP'. The case of names must be identical with that defined by system, such as 'IP'. To view service resources, execute command `ZXR10#`**define service show default**.
schedule	This selects time resource, which must be defined in previous define module.
<*string8*>	This is the object name.
sport	This specifies service resource on source port.
<*string9*>	This is one string, which must be system-predefined service resource name.
orig_dst	This specifies destination address before NAT.
<*string10*>	This is one string, which must be system-predefined address name.
permanent	This is optional. switch of long connection. It is disabled by default, which means the connection is an common connection. In usual cases, FW disconnects one connection if communication on it is idle for a period for improving security and releasing communication resources. However connection for some applications requires long-time holding, even if the connection is in idle state. For example, ATM must hold connection with server at processing center, so this connection must be set to long connection.
on|*off*	On/off, indicating long connection and common connection.
log	This is optional. It sets whether to record the event in log or prompt alarm message when a packet matches rule. It doesn't record event into log by default.
on|*off*| *alarm*	This records the event into log/doesn't record the event into log/generates alarm.
enable	This is optional, indicating whether to enable this rule. The rule is enabled by default.

Parameter	Description
yes|*no*	Enable/not enable
before	This is optional. When adding one new access control rule, it is available to select before which rule to place this new rule. The new rule is placed at end by default.
<*number*>	This is one number, indicating ID of added access control rule.

Example:

To add one access control rule. execute the following command, where 'area_eth0', 'any', 'http_policy' and 'msn' are defined objects.

```
    ZXR10_FW.firewall #policy add action accept
srcarea 'area_eth0' src 'any' service IP dpi 'http_policy'
 ar 'msn' av on log on enable yes
```

2. Modifying one added access control rule.

Command	Function
ZXR10_FW.firewall #**policy modify id** <*number1*>[**action** <*accept*|**deny**>][**srcarea** <*string 1*>][**dstarea** <*string2*>][**srcvlan** <*string3*>][**dstvlan** <*string4*>][**src** <*string5*>][**dst** <*string6*>][**service** <*string7*>][**schedule** <*string8*>][**sport** <*string9*>][**orig_dst** <*string10*>][**dpi** <*string11*>][**ar**<*string12*>][**av**<*on*|*off*>][**permanent** <*on*|*off*>][**log** <*on*|*off*>][**enable** < *yes*|*no*>]	This modifies one added access control rule.

Parameter Description:

Parameter	Description
modify	This modifies one FW access rule.
id	This is ID of defined access control rule.
<*number1*>	This is one number.
action	This sets access right.
accept|*deny*	permit/deny
srcarea	This sets source area.
<*string1*>	This is one string. It must be one or more preset area name(s). As for multiple area names, space is used between each two area names and all addresses are quoted with single quotes, such as 'area_gei_5/1'.
dstarea	This sets destination area.
<*string2*>	This is one string. It must be one or more preset area name(s). As for multiple area names, space is used between each two area names and all addresses are quoted with single quotes, such as 'area_gei_5/1'.

Parameter	Description
srcvlan	This sets source VLAN.
<*string3*>	This is one string, indicating preset vlan number.
dstvlan	This sets destination VLAN.
<*string4*>	This is one string, indicating preset vlan number.
src	This is source address.
<*string5*>	This is one string, indicating preset address name. Multiple address names can be input and space is used between each two address names and all address names are quoted with single quotes, such as 'aa ll'.
dst	This is destination address.
<*string6*>	This is one string, indicating preset address name. Multiple address names can be input and space is used between each two address names and all address names are quoted with single quotes, such as 'aa ll'.
service	This sets service resource.
<*string7*>	This is one string. It must be one or more names of system default services or customized services. As for multiple service names, space is used between each two service names and all service names are quoted with single quotes, such as 'IP ICMP'. The case of names must be identical with that defined by system, such as 'IP'. To view service resources, execute command `ZXR10#`**define service show default**.
schedule	This selects time resource, which must be defined in previous define module.
<*string8*>	This is the object name.
sport	This specifies service resource on source port.
<*string9*>	This is one string, which must be system-predefined service resource name.
orig_dst	This specifies destination address before NAT.
<*string10*>	This is one string, which must be system-predefined address name.
dpi	This sets DPI object.
<*string11*>	This specifies object name, which must be defined in previous DPI module and only one name can be selected.
ar	This sets application identification policy.

Parameter	Description
<string12>	This specifies object name, which must be defined in content filtering module and only one name can be selected.
av	This sets whether to enable anti-virus module.
on|off	Enable/disable, disable by default.
permanent	This is the switch of long connection.
on|off	on/off
log	It sets whether to record the event in log or prompt alarm message when a packet matches rule. It doesn't record event into log by default.
on|off	This records the event into log/doesn't record the event into log/generates alarm.
enable	This specifies whether to enable this rule.
yes|no	Enable/Disable

Example:

To modify one access control rule. execute the following command, where 'area_eth0', 'any', 'http_policy' and 'msn' are defined objects.

```
ZXR10_FW . firewall #policy modify id 8048 action
accept srcarea 'area_eth0' src 'any' service IP dpi
'http_policy' ar 'msn' av on log on enable yes
```

3. Deleting one access control rule. execute the following command,

Command	Function
`ZXR10_FW.firewall #`**policy delete id** *<number1>*	This deletes one access control rule. execute the following command.

Parameter Description:

Parameter	Description
<number1>	This is one string, which must be ID of predefined rule.

Example:

To delete one access control rule whose id is 8503, execute the following command:

```
ZXR10_FW.firewall #policy delete id 8503
```

Access Control Rule Configuration Example

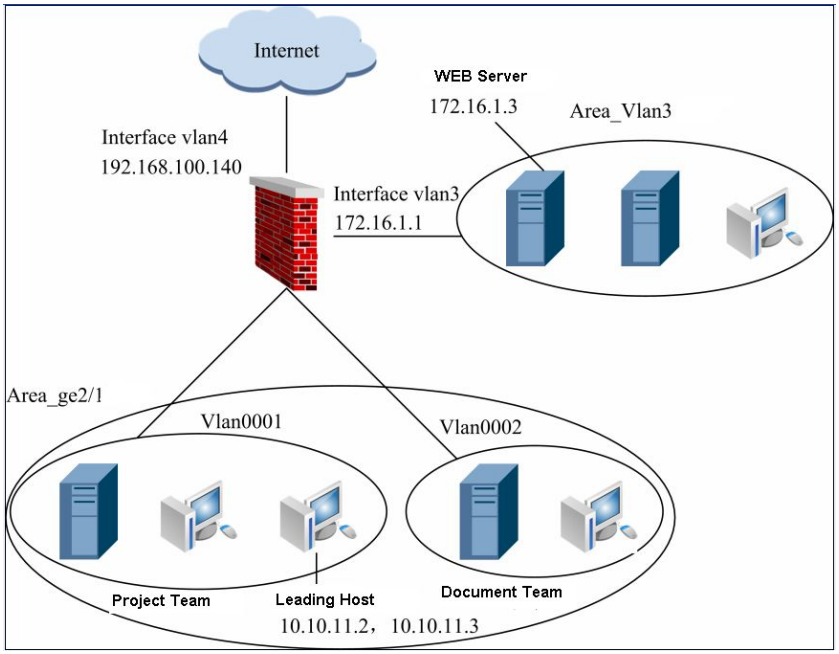
Access Control Rule Configuration Example One

As shown in network structure diagram of an enterprise, FW cards works in hybrid mode. Interface gei_2/1 belongs to intranet area_gei_2/1, is a switch trunk port, belongs to both VLAN.0001 and VLAN 0002, where IP address of vlan 0001 is 10.10.10.1 and connected to intranet 10.10.10.0/24 where document team of R&D department locates; IP, and IP address of vlan 0002 is 10.10.11.1 and connected to intranet 10.10.11.0/24 where project team of R&D department locates. IP address of Interface Vlan4 is 192.168.100.140 and belongs to extranet area_vlan4. Intranet is connected with extranet through the router connected with FW Interface Vlan4. Interface vlan3 belongs to area_vlan3 and it is route interface with interface IP to be 172.16.1.1. Data management department locates in area_vlan3 and multiple servers are in this area, where IP address of web server is 172.16.1.3.

User has the following requirements:

- PCs of intranet document team can access internet, leaders of project team can access internet, and common members of project team cannot access internet;
- Extranet and PCs in area_vlan3 cannot access intranet of R&D department;
- All pcs in intranet can access web server in area_vlan3.

FIGURE 2 ACCESS CONTROL RULE CONFIGURATION EXAMPLE ONE



1. To set IP addresses for interface vlan.0004 and interface vlan.0003, execute the following commands:

```
ZXR10(config)#vlan 3 ZXR10(config-vlan3)#exit
ZXR10(config)#interface vlan 3
ZXR10(config-if-vlan3)# ip addr 172.16.1.1  255.255.255.0
ZXR10(config-if-vlan3)#exit ZXR10(config)#vlan 4
ZXR10(config-vlan4)#exit ZXR10(config)#interface vlan 4
ZXR10(config-if-vlan4)# ip addr 192.168.100.140  255.255.255.0
```

2. To set IP addresses for interface vlan.0001 and interface vlan.0002 and join interface gei_2/1 to two vlans in trunk mode, execute the following commands:

```
ZXR10(config)#vlan 1 ZXR10(config-vlan1)#exit
ZXR10(config)#interface vlan 1
ZXR10(config-if-vlan1)# ip addr 10.10.10.1   255.255.255.0
ZXR10(config-if-vlan1)#exit ZXR10(config)#vlan 2
ZXR10(config-vlan2)#exit ZXR10(config)#interface vlan 2
ZXR10(config-if-vlan2)# ip addr 110.10.11.1  255.255.255.0
ZXR10(config-if-vlan2)#exit ZXR10(config)#int gei_2/1
ZXR10(config- gei_2/1)# switchport mode trunk
ZXR10(config- gei_2/1)# switchport  trunk vlan 1-2
ZXR10(config- gei_2/1)# switchport  trunk  native vlan 1
```

3. Defining host and subnet address resources (gei_2/8 is the interface for switch connecting with extranet and it belongs to vlan4).

```
ZXR10_FW #define host add name 172.16.1.3 ipaddr 172.16.1.3
ZXR10_FW #define host add name 192.168.100.143 ipaddr 192.168.10
ZXR10_FW #define host add name doc_server ipaddr 10.10.10.3
ZXR10_FW #define subnet add name rd_group ipaddr 10.10.11.0 mask
255.255.255.0 except  '10.10.11.2 10.10.11.3'
ZXR10_FW #define area add name area_vlan4 access on attribute ge
```

4. Setting default privilege for accessing area resource: conduct the following configurations on main board to join interface gei_2/3 to vlan3 in access mode.

```
ZXR10(config- gei_2/3)# switchport mode acesss
ZXR10(config- gei_2/3)# switchport  access vlan 3
ZXR10_FW #define area add name gei_2/1 access off attribute
gei_2/1 (deny accessing intranet)
ZXR10_FW #define area add name area_vlan3 access off attribute
gei_2/3 (deny accessing intranet)
```

5. Defining NAT rule.

   Execute the following command to define source address translation rule, so that intranet users can access extranet.

```
ZXR10_FW #nat policy add dstarea area_vlan4
trans_src 192.168.100.140
```

   Execute the following command to define destination address translation rule, so that both intranet document team and extra users can access web server of area_vlan3 (192.168.100.143 is a bogus extranet address, used to access web server).

```
ZXR10_FW #nat policy add orig_dst 192.168.100.143
orig_service HTTP trans_dst 172.16.1.3
```

6. Defining access control rule.

To permit intranet users to access web server, execute the following command:

```
ZXR10_FW #firewall policy add action accept
srcarea area_gei_2/1 dst 172.16.1.3 service HTTP
```

To permit leaders of project team to access extranet and deny common members of project team accessing extranet, execute the following command:

```
ZXR10_FW #firewall policy add action deny ssrcvlan
vlan.0002 src rd_group dstarea area_vlan4 service HTTP
```

Notes:

- It needs to select the actual IP address of web server as destination address, since FW needs translating destination address of the packet firstly. When an intranet user access web server of SSN area through http://192.168.100.143, destination address of the packet will be translated to 172.16.1.3 since it meets NAT destination address translation rule. The next step is to proceed access rule query. Only when destination address is set to actual ip address of web server, can intranet user access web server of SSN area.
- When defining destination address translation rule, don't select destination area and destination vlan.

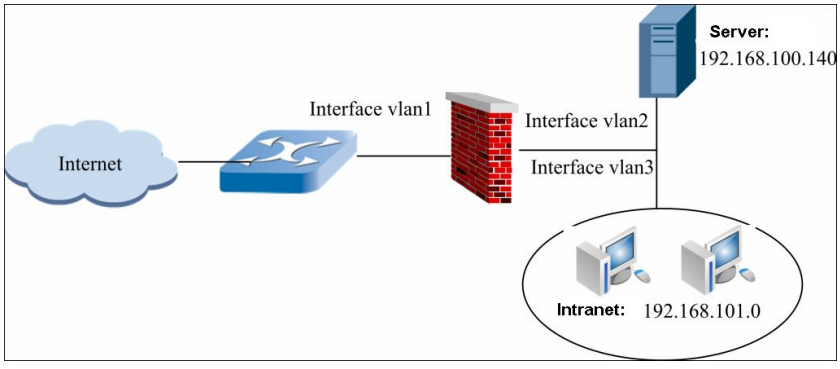
Access Control Rule Configuration Example Two

An enterprise network is divided into three areas: area_vlan1, area_vlan2 and area_vlan3. The three areas are bound with interface vlan1, interface vlan2 and interface vlan3 respectively. Area_vlan1 is connected with extranet and permits user access. Area_vlan2 and area_vlan3 forbid user to access. Server locates in area_vlan2 and IP address is 192.168.100.140. Intranet locates in area_vlan3 and network address is 192.168.101.0. Enterprise network structure is shown in Figure 3.

User has the following requirements:

Intranet user can access TELNET, SSH, FTP and Web_port services on server, where Web_port service is customized, and port id is 8080; intranet user cannot access other servers and services on Interface vlan 2. Extranet user can access TCP service on Interface vlan 2 server and the port id is 8080.

FIGURE 3 ACCESS CONTROL RULE CONFIGURATION EXAMPLE TWO



Configuration Points:

- Defining area and address resources
- Defining service resource
- Defining service group resource

1. Defining area resource: To join gei_2/1, gei_2/2 and gei_2/3 to vlan1, vlan2 and vlan3 respectively in access mode, execute the following commands:

```
ZXR10_FW #define area add name area_vlan1 access
on attribute gei_2/1 ZXR10_FW #define area add name area_vlan2
 access on attribute gei_2/2 ZXR10_FW #define area add name
area_vlan3 access on attribute gei_2/3
```

2. Defining host and subnet address resources: To define host address resource "192.168.100.140" and subnet address source "intranet resource" inner_web, execute the following commands:

```
ZXR10_FW #define host add name 192.168.100.140
ipaddr 192.168.100.140 ZXR10_FW #define subnet add name
inner_web ipaddr 192.168.101.0 mask 255.255.255.0
```

3. Defining customized service with service name to be Web_port and port id to be 8080, execute the following command:

```
ZXR10_FW #define service add name Web_port
protocol tcp port 8080
```

4. Setting service group resource: To name service group to inner_web_srv (intranet access service) and include services Web_port, FTP, Telnet and SSH into this group, execute the following command:

```
ZXR10_FW #difine group_service add name
inner_web_srv member Web_port,FTP,TELNET,SSH
```

5. Setting access control rule:

   Permit subnet object (intranet) inner_web (192.168.101.0/24) in area_vlan3 to access Web_port, FTP, TELNET and SSH services (bound with customized service group inner_web_srv) on server of area_vlan2 with server IP address to be 192.168.100.140.

```
ZXR10_FW #firewall policy add action accept
srcarea area_vlan3 dstarea area_vlan2 src inner_web dst
192.168.100.140 service inner_web_srv enable yes
```

   Setting service access control rule to only permit extranet user (area_vlan1) to access services on port 8080 of server 192.168.100.140.

```
ZXR10_FW #firewall policy add action accept
srcarea area_vlan1  dstarea area_vlan2 dst 192.168.100.140
service Web_port enable yes
```

Notes:

To permit only partial services to be accessed and deny others, set the default access privilege of destination area to "deny". System will match default access privilege of area automatically after matching access control rule.

Configuring IDS Interaction

IDS Interaction Overview

It is hard for one security system to integrate all security technologies. It is convenient for management and maintenance to include IDS, anti-virus, content auditing and other functions into FW and it can also degrade performance of FW, so it is inappropriate for FW which acts as GW to integrate all security technologies. Firstly, IDS needs to update attack pattern database periodically. However it is obviously inappropriate to upgrade critical devices, such as FW. Secondly, if FW contains too many additional functions, its running speed will be slower, which brings bottleneck for communication between intranet and extranet.

Taking convenience and maintainability of user operation and security system construction into account, FW designs core platform in security system and provides sound assistant system for IDS, anti-virus and other security products to interact with products of other main IDS and anti-virus manufacturers.

Configuring IDS Interaction

This topic describes how to configure IDS interaction.

1. Adding one IDS interaction rule.

Command	Function
`ZXR10_FW.pf #`**idsserver add ip** *<ipaddress>* **key** *<string>*	This adds one IDS interaction rule.

Parameter Description:

Parameter	Description
add	This adds one IDS interaction rule.
ip	This sets IP address for interacted IDS.
<ipaddress>	This is IP address string, in format of A.B.C.D.
key	This sets shared key with IDS device.
<string>	This is one string.

Command Illustration:

- Shared key of FW and interacted IDS device is set manually. If the key of interacted IDS device is generated automatically by system , user needs to give the configuration on WEBUI interface. For details, please refer to Logging into FW through Browser.

- To realize interaction between FW and IDS device, it needs to enable IDS servie in corresponding area. For details, please refer to section Setting Open Services.

2. Modifying one IDS interaction rule.

Command	Function
ZXR10_FW.pf #**idsserver modify id**<*number*><[**ip** <*ipaddress*>][**key** <*string*>]>	This modifies one IDS interaction rule.

Parameter Description:

Parameter	Description
modify	This modifies one IDS interaction rule.
id	This modifies ID for the rule to be modified.
<*number*>	This is one number.
ip	It is an option, modifying IP address of IDS interaction device.
<*ipaddress*>	This is a string, in format of A.B.C.D.
key	This modifies shared key of FW and IDS device.
<*string*>	This is one string.

3. Deleting one IDS interaction rule.

Command	Function
ZXR10_FW.pf #**idsserver delete id** <*number*>	This deletes one IDS interaction rule.

4. Clearing all IDS interaction servers.

Command	Function
ZXR10_FW.pf #**idsserver clean**	This clears all IDS interaction servers.

5. Showing all IDS interaction rules.

Command	Function
ZXR10_FW.pf #**idsserver show**	This shows all IDS interaction rules.

This page is intentionally blank.

Chapter 6

NAT Configuration

Table of Contents:

NAT Overview

Rapid development of Internet speeds lack of IP addresses. To alleviate this problem, RFC1631 and related RFC define Network Address Translation (NAT), which is used widely. NAT is to map an IP address from one address domain to another address domain. One of this typical application is to map private IP address defined in RFC1918 to available public IP address in Internet.

RFC 1918 gives the following definitions to private IP address:

Internet Assigned Numbers Authority (IANA) reserves three IP addresses for private network.

10.0.0.0 - 10.255.255.255 (Class A address segment)

172.16.0.0 - 172.31.255.255Class B address segment)

192.168.0.0 - 192.168.255.255 (Class C address segment)

When one user of private IP address needs to access public network or one user in public network needs to access one server with private IP address, administrator needs to set corresponding address translation rule.

Address Translation Advantages

With network address translation, enterprises can use quite a few Internet public IP addresses to access Internet, which relieves lack of IPv4 addresses and meanwhile provides certain security. NAT has the following advantages:

- Guarantee that intra-users of enterprises using private IP addresses can access Internet normally;
- Protect intranet, hide intranet topology and actual IP, and reduce direct attacks.
- Protect internal server that provides service externally and provide the function of load balancing.

Address Translation Rule

FW card can configure NAT rules flexibly according to user network planning and function demands. When user defines address translation rules on FW card, firstly he needs to define source and destination of this rule, that is source address range and destina-

tion address range of packet of applicable to address translation rule, then define corresponding services, and the last one is translation control mode. FW card provides the following translation control modes:

- SNAT: Users with private addresses can access public network.
- DNAT: Users in public network can access intranet server with private address.
- NoNAT: It can be used to define special cases of SNAT, DNAT, bi-directional NAT rules. In this case, it shall be placed in the front of NAT rule list.

All address translation rules defined by FW card are stored in rule table in certain sequence. When one packet passes through FW card, FW card will retrieve address translation rule table according to sequence of address translation rules and match them with the packet one by one. Once the packet is found to match one address translation rule, FW card will stop retrieving and process the packet according to defined rule.

Configuring NAT

This topic describes configuration commands and configuration examples of NAT.

Commands in this module are used for address translation policy-related configurations.

To access this command module, execute the following command:

`ZXR10_FW # nat`

To exit from this command module, execute the following command:

`ZXR10_FW # end`

1. Adding NAT policy.

Command	Function
`ZXR10_FW.nat#`**policy add** [**srcarea** <*srcarea_nam*>][**dstarea** <*dstarea_nam*>][**srcvlan** <*srcvlan_no*>][**dstvlan** <*dstvlan_no*>][**orig_src** <*src_addr1*>][**orig_dst** <*dst_addr1*>][**orig_sport** <*sport_id*>][**orig_service** <*ser_id*>][**trans_src** < *src_addr2*>][**trans_dst** <*dst_addr2*>][**trans_service** <*ser_obj*>][**pat** <*yes*|*no*>][**enable** <*yes*|*no*>][**before** <*number2*>]	This adds NAT policy.

Parameter Description:

Parameter	Description
add	This adds NAT policy.
srcarea	This sets source area.

Parameter	Description
<*srcarea_nam*>	This is one string and source area resource name is input here. Tips: This parameter value must be predefined area name. One ore more area names can be input here. As for multiple area names, space is used between each two area names and all addresses are quoted with single quotes, such as 'area1 area2'
dstarea	This sets destination area.
<*dstarea_nam*>	This is one string and destination area resource name is input here. Tips: a) This parameter value must be predefined area name. One ore more area names can be input here. As for multiple area names, space is used between each two area names and all area names are quoted with single quotes, such as 'area1 area2'. b) When adding destination address translation policy, this parameter mustn't be set.
srcvlan	This sets source VLAN.
<*srcvlan_no*>	This is one string and source VLAN name is input here. Tips: This parameter value must be predefined VLAN name. One ore more VLAN names can be input here. As for multiple VLAN names, space is used between each two VLAN names and all VLAN names are quoted with single quotes, such as '1 2'.
dstvlan	This sets destination VLAN.
<*dstvlan_no*>	This is one string and destination VLAN name is input here. Tips: a) This parameter value must be predefined VLAN name. One ore more VLAN names can be input here. As for multiple VLAN names, space is used between each two VLAN names and all VLAN names are quoted with single quotes, such as '1 2'. b) When adding destination address translation policy, this parameter mustn't be set.
orig_src	This sets source object of original packet.

Parameter	Description
<*src_addr1*>	This is one string and source object name of original packet is input here. Tips: a) This parameter must be one predef ined address object name. b) Multiple address objects can be input at the same time, in format of 'test1 test2'. As for multiple address objects, space is used between each two and all address object names are quoted with single quotes.
orig_dst	This sets destination object of original packet.
<*dst_addr1*>	This is one string and source object name of original packet is input here. Tips: a) This parameter must be one predef ined address object name. b) Multiple address objects can be input at the same time, in format of 'test1 test2'. As for multiple address objects, space is used between each two and all address object names are quoted with single quotes.
orig_sport	This sets source port of original packet.
<*sport_id*>	This is one string and source port name of original packet is input here. Tips: a) This parameter must be one predef ined port name. b) Multiple port objects can be input at the same time, in format of 'server1 server2'. As for multiple port objects, space is used between each two and all port names are quoted with single quotes.
orig_service	This sets service resource of original packet.
<*ser_id*>	This is one string and service resource name of original packet is input here. Tips: a) This parameter must be one predef ined service resource. b) Multiple service resources can be input at the same time, in format of 'ser ver1 server2'. As for multiple service resources, space is used between each two and all address object names are quoted with single quotes.
trans_src	This sets source object after translation.

Parameter	Description
<*src_addr2*>	This is one string and source object name after translation is input here. Tips: a) This parameter must be one prede fined address object name or attribute name. b) Only one object can be input here. c) This parameter is necessary when adding source address translation poli cy.
trans_dst	This sets destination object after translation.
<*dst_addr2*>	This is one string and destination object name after translation is input here. Tips: a) This parameter must be one prede fined address object name or attribute name. b) Only one object can be input here. c) This parameter is necessary when adding destination address translation policy.
trans_service	This sets service resource after translation.
<*ser_obj*>	This is the number and service resource name after translation is input here. Tips: This parameter must be one name predefined in define module.
pat	This sets source port translation switch.
yes|*no*	YES means conducting port address translation to source port and no means not conducting port address translation to source port. Tips: Yes is the default value.
enable	This sets address translation policy switch.
yes|*no*	Yes means enabling this address translation policy and no means forbidding this address translation policy temporarily. Tips: Yes is the default value.

Parameter	Description
before	This places this address translation policy before one policy.
<*number*>	This is one number, which shall be ID of the next address translation policy after inputting this address translation policy. Tips: a) This parameter must be id of an added address translation policy. b) To view id of system default service resource, execute the following command: **define service show default**

Command Illustration:

When defining destination address translation policy, don't specify destination area and destination vlan.

System also translates source port address by default when translating source address.

Attribute resource with no interface binding cannot be used as address after translation of address translation policy.

2. Modifying NAT policy.

Command	Function
ZXR10_FW.nat#**policy modify id** <*number1*>[**srcarea** <*srcarea_nam*>][**dstarea** <*dstarea_nam*>][**srcvlan** <*srcvlan_no*>][**dstvlan** <*dstvlan_no*>][**orig_src** <*src_addr1*>][**orig_dst** <*dst_addr1*>][**orig_sport** <*sport_id*>][**orig_service** <*ser_id*>][**trans_src** < *src_addr2*>][**trans_dst** <*dst_addr2*>][**trans_service** <*ser_obj*>][**pat** <*yes*|*no*>][**enable** <*yes*|*no*>]	Modifying NAT policy.

Parameter Description:

Parameter	Description
modify	This modifies NAT policy.
id	This sets ID of policy to be modified.
<*number*>	This is one number, which is the ID of policy to be modified.

3. Deleting one NAT policy.

Command	Function
ZXR10_FW.nat#**policy delete id** <*number1*>	This deletes one NAT policy.

Parameter Description:

Parameter	Description
<*number1*>	This is one number, which is the ID of NAT policy to be deleted.

4. Showing NAT policy.

Command	Function
ZXR10_FW.nat#**policy show**	This shows NAT policy.

5. Clearing NAT policy.

Command	Function
ZXR10_FW.nat#**policy clean**	This clears NAT policy.

6. Moving NAT policy.

Command	Function
ZXR10_FW.nat#**policy move** <*number1*>[<**before** <*number2*>|**after** <*number3*>]	This moves NAT policy. NAT policy conforms to sequential matching principle. By moving NAT policy, matching priority of policy can be changed.

Parameter Description:

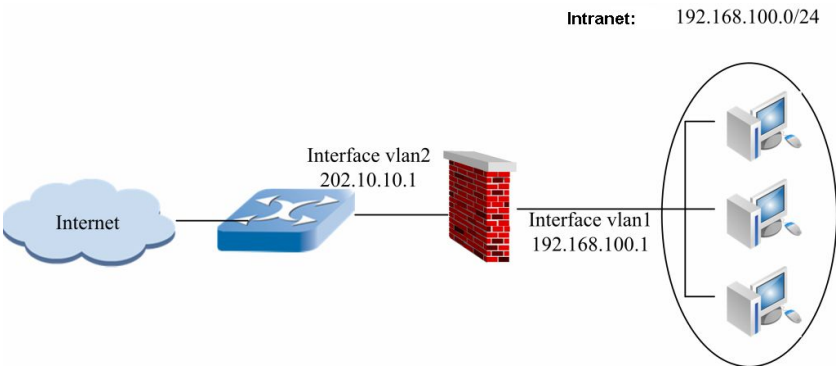
Parameter	Description
<*number1*>	This is one number, which is the ID of NAT policy to be moved.
<*number2*>	This is one number, which is the ID of policy. Tips: Values of parameters before and after cannot be set at the same time.
<*numbe3*>	This is one number, which is the ID of policy. Tips: Values of parameters before and after cannot be set at the same time.

NAT Configuration Example

Address-Based Source Address Translation Configuration Example

Source address translation policy of FW card supports address resource-based source address translation. Address resources that can be translated include single host, host address range and subnet. Source address can be translated in the following modes: fixedly mapping source address to a legal IP address and dynamically mapping source address to a network segment or the address within an address range. Gei_3/1 in interface vlan1 on FW card is connected to intranet, intranet address is 192.168.100.0/24 and ip address of interface vlan1 is 192.168.100.1; Gei_3/2 in interface vlan2 is connected to extranet and ip address of interface vlan2 is 202.10.10.1. The available range of public network IP address for enterprise is 202.10.10.1-202.10.10.10. Network topology diagram is shown in Figure 4.

FIGURE 4 ADDRESS-BASED SOURCE ADDRESS TRANSLATION CONFIGURATION EXAMPLE



1. To define area resource, execute the following command:

```
ZXR10_FW #define area add name area_vlan2 access on
attribute gei_3/2 ZXR10_FW #define area add name area_vlan1
access off  attribute gei_3/1
```

2. To define intranet address resource, execute the following command:

```
ZXR10_FW #define subnet add name subnet1 ipaddr
192.168.100.0 mask 255.255.255.0
```

3. To define NAT address pool resource, execute the following command:

```
ZXR10_FW #define range add name nat-pool ip1
202.10.10.1 ip2 202.10.10.10
```

4. To define NAT rule and dynamically select one IP address after being translated in address pool, execute the following command:

```
ZXR10_FW #nat policy add srcarea area_vlan1
```

```
orig_src sbunet1 dstarea area-vlan2 trans_src  nat-pool
enable yes
```

Notes:

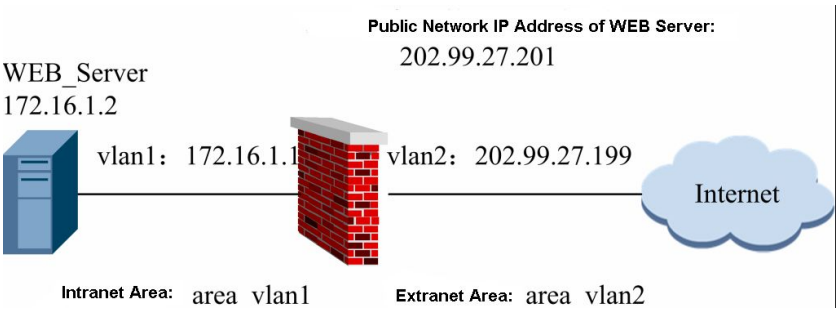
System also translates source port address by default when translating source address.

IP Address-Based Destination Address Translation Configuration Example

Due to frequent Internet attacks to government and enterprise networks, it is necessary to provide protection to the intranet critical device which provides access service to extranet. With destination address NAT, intranet addresses can be hidden.

Internet users need to access WEB server through FW. To hide the actual address 172.16.1.2 of server in intranet, public network address 202.99.27.201 is used as user access address. Network topology diagram is shown in Figure 5.

FIGURE 5 IP ADDRESS-BASED DESTINATION ADDRESS TRANSLATION CONFIGURATION EXAMPLE



Configuration Points:

- Defining area resource: area_vlan2.
- Defining address resource corresponding to actual address of WEB server.
- Defining public network virtual IP address resource of WEB server.
- Defining NAT policy.

1. To set area_vlan2 and define default attribute to permit to access, execute the following command:

   ```
   ZXR10_FW #define area add name area_vlan2 access
   on attribute interface vlan2
   ```

   To set area_vlan1 and define default attribute to deny accessing, execute the following command:

   ```
   ZXR10_FW #define area add name area_vlan1
   access off attribute interface vlan1
   ```

2. To specify actual address of WEB server, execute the following command:

```
ZXR10_FW #define host add name
WEB_server ipaddr 172.16.1.2
```

3. To specify public network address of WEB server, execute the following command:

```
ZXR10_FW #define host add name MAP_IP ipaddr
202.99.27.201
```

4. To set NAT rule, execute the following command:

```
ZXR10_FW #nat policy add srcarea area_vlan2 orig_dst
MAP_IP orig_service http trans_dst WEB_server
```

Notes:

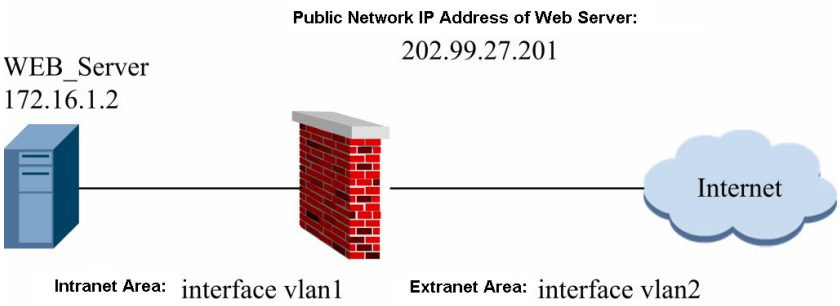
- When defining destination address translation policy, don't specify destination area and destination vlan.
- If web server uses customized port id rather than standard port 80 to provide web service, the actual applied server port shall be filled in "destination port is translated to" when defining address translation policy. For detailed configuration method, please refer to "Port-Based Destination NAT" configuration example.

Port-Based Destination Address Translation Configuration Example

Basic requirement: With destination address NAT, intranet addresses can be hidden. However, sometimes server open application port is different from the port used for user access (default port in usual cases). In this case, NAT is necessary.

Public Internet user accesses web server through public address 202.99.27.199:80. Actual address of web server is 172.16.1.2 and the port providing HTTP service is 8080. The network topology diagram is as shown in Figure 6.

FIGURE 6 PORT-BASED DESTINATION ADDRESS TRANSLATION CONFIGURATION EXAMPLE



Configuration Points:

- Defining area resource.

- Specifying actual address of WEB server.
- Specifying WEB server access address.
- Specifying actual port of WEB server.
- Defining NAT policy.

1. To set E1 and E0 areas, execute the following commands:

```
ZXR10_FW #define area add name E1 access on attribute
interface vlan2 ZXR10_FW #define area add name E0 access off
attribute interface vlan1
```

2. To specify actual address of WEB server, execute the following command:

```
ZXR10_FW #define host add name WEB_server ipaddr 172.16.1.2
```

3. To specify WEB server access address, execute the following command:

```
ZXR10_FW #define host add name MAP_IP ipaddr 202.99.27.199
```

4. To define service port, execute the following command:

```
ZXR10_FW #define service add name Web_port protocol 6 port 8080
```

   Tips: 6 here is the TCP protocol number.

5. To define NAT rule, execute the following command:

```
ZXR10_FW #nat policy add srcarea E1 orig_dst MAP_IP
orig_service http trans_dst Web_server trans_service Web_port
```

Notes:

- When public user accesses web server, the used default port is port 80 and the port for web server providing services is port 8080. Therefore, destination address NAT is necessary.
- When defining destination address NAT, note not to define destination AREA and destination VLAN.

This page is intentionally blank.

Chapter 7

Protocol Filtering Configuration

Table of Contents:

Protocol Filtering Overview

FW card can provide control over application layer with finer granularity through content filtering. At present, FW card permits user to bind application layer protocol with port.

Configuring Application Port Binding

Application Port Binding Overview

Application port binding on FW card is used to bind application layer protocol and port. When FW card performs deep content detection to application layer protocol, it will check these packets and perform corresponding processing.

System defines parting bindings between application layer protocols and standard ports by default. In case some application layer protocols use non-standard ports or sub-connection (such as FTP) is contained, user must bind these ports and application layer protocols, or FW card will fail to detect and process these packets. For example, there is one FTP server in intranet SSN area and non-standard port 2200 is used to provide FTP service for intranet users. In case application port binding is unavailable, FW will not process packets of this connection. Customize application protocol port binding policy.

System has the following default standard ports:

Application Protocol Name	Default Port ID	Protocol Type	Used Protocol Number
FTP	21	TCP	6
SMTP	25	TCP	6
TFTP	69	UDP	17
HTTP:	80	TCP	6
IMAP	143	TCP	6
Telnet	23	TCP	6
POP3	110	TCP	6

Configuring Application Port Binding

This topic describes configuration commands of application protocol filtering.

Commands in this module are used for application protocol filtering-related configurations.

To access this command module, execute the following command:

`ZXR10_FW #`**dpi**

To exit from this command module, execute the following command:

`ZXR10_FW #`**end**

1. Customizing one application protocol port binding policy.

Command	Function
`ZXR10_FW.dpi #`**policy add name** *<ftp|tftp|sun_rpc|ms_rpc|sqlnet|rtsp|h225|h225ras|mms|sip|pptp>* **net** *<string1>* **mask** *<string2>* **protocol** *<udp|tcp>* **port** *<number>*	This customizes one application protocol port binding policy.

Parameter Description:

Parameter	Description
add	This adds one application protocol port binding policy.
name	This is one string, specifying the name of protocol which is to re-define port.
tftp|sun_rpc|ms_rpc|sqlnet|rtsp|h225|h225ras|mms|sip|pptp	tftp protocol|sun_rpc protocol|ms_rpc protocol|sqlnet protocol|rstp protocol|h225 protocol|h225ras protocol|mms protocol|sip protocol|pptp protocol
net	This specifies subnet address.
<string1>	This is one standard IP address string.
mask	This specifies subnet mask.

Parameter	Description
<*string2*>	This is one standard address mask string.
protocol	This specifies the name of used protocol.
udp|*tcp*	This selects the protocol to be used: UDP or TCP.
port	This customizes service port of this protocol.
<*number*>	This is the service port id.

2. Modifying one application protocol port binding policy.

Command	Function
ZXR10_FW.dpi #**policy modify id** <*number1*>**name** <*ftp*|*smtp*|*tftp*|*http*|*pop3*|*sun_rpc*|*ms_rpc*|*sqlnet*|*rtsp*|*h225*|*h225ras*|*mms* | *sip*|*imap*|*telnet* > **net** <*string1*> **mask** <*string2*> **protocol** <*udp*|*tcp*> **port** <*number2*>	This modifies one application protocol port binding policy.

Parameter Description:

Parameter	Description
modify	This modifies one application protocol port binding policy.
id	This is the ID of policy to be modified.
<*number1*>	This is one number.
name	This is one string, specifying the name of protocol which is to re-define port.
ftp|*smtp*|*tftp*|*http*|*pop3*|*sun_rpc*|*ms_rpc*|*sqlnet*|*rtsp*|*h225*|*h225ras*|*mms* | *sip*|*imap*|*telnet*	ftp protocol|smtp protocol|tftp protocol|http protocol|pop3 protocol|sun_rpc protocol|ms_rpc protocol|sqlnet protocol|rstp protocol|h225 protocol|h225ras protocol|mms protocol|sip protocol|imap protocol|telnet protocol

Command Illustration:

Command **policy show** can be used to view ID of policy.

3. Deleting one application protocol port binding policy.

Command	Function
ZXR10_FW.dpi #**policy delete id** <*number*>	This deletes one application protocol port binding policy.

Parameter Description:

Parameter	Description
<*number*>	This is one number, which is the ID of policy.

Command Illustration:

Command **policy show** can be used to view ID of policy.

4. Clearing all application protocol port binding policies.

Command	Function
ZXR10_FW.dpi #**policy clean**	This clears all application protocol port binding policies.

5. Showing all application protocol port binding policies.

Command	Function
ZXR10_FW.dpi #**policy show**	This shows all application protocol port binding policies.

6. Restoring default application protocol port.

Command	Function
ZXR10_FW.dpi #**policy reset**	This restores default application protocol port.

Applying Port Binding Configuration Example

To bind HTTP packets sent to subnet 192.168.0.0/255.255.0.0 with destination port 8080, execute the following command:

```
ZXR10_FW.dpi# policy add name http net 192.168.0.0
mask 255.255.0.0 protocol tcp port 8080
```

To delete one application protocol port binding policy whose id is 8547, execute the following command:

```
ZXR10_FW.dpi# policy delete id 8547
```

To modify port id of FTP packets in port binding policy whose id is 8182 to 2121, execute the following command:

```
ZXR10_FW.dpi# policy modify id 8182 name ftp
net 0.0.0.0 mask 0.0.0.0 protocol tcp port 2121
```

Configuring SIP Service

Session Initiation Protocol (SIP) is a signaling control protocol of application layer .

This topic describes how to configure SIP service.

Commands in this module are used for application protocol filtering-related configurations.

To access this command module, execute the following command:

ZXR10_FW #**dpi**

To exit from this command module, execute the following command:

ZXR10_FW #**end**

1. Enabling SIP service.

Command	Function
ZXR10_FW.dpi #**sip start**	This enables SIP service.

2. Disabling SIP service.

Command	Function
ZXR10_FW.dpi #**sip stop**	This disables SIP service.

This page is intentionally blank.

Chapter 8

Intrusion Prevention Configuration

Table of Contents:

Intrusion Prevention Overview

FW card has a built-in IDS module, used to detect and defend common attacks and scanning. Meanwhile, FW card can realize interaction with intrusion detection system of other manufacturers and provides comprehensive and efficient security protection to user intranet.

This chapter has the following content:

- Host intrusion prevention. This topic describes how does FW card provide host intrusion prevention function to all hosts.
- Anti-DOS. This topic describes how does FW card detect and defend common attacks.

Configuring Intrusion Detection Rule

This topic describes commands of configuring intrusion detection rule.

To access this command module, execute the following command:

```
ZXR10_FW #ips
```

To exit from this command module, execute the following command:

```
ZXR10_FW #end
```

1. Adding protected object (host, subnet, range or address group).

Command	Function
ZXR10_FW.ips #**dos rule add protect_name** <*string*> **icmpflood** <*number1*> **ipsweep** <*number2*> **synflood** <*number3*> **udpflood** <*number4*> **portscan** <*number5*>[**log** <*yes*|*no*>][**action** <*pass*|*block*>]	This adds the host or subnet to be protected from intrusion.

Parameter Description:

Parameter	Description
add	This adds one host or subnet to be protected.
protect_name	This sets address resource to be protected, which can be host, subnet or address range. This address resource shall be added in command **define** in advance.
<*string*>	This is one string, indicating the name of address resource.
icmpflood	This sets the max reply requests initiated to protected object per second.
<*number1*>	This is one number, indicating max connection requests, 500 by default, ranging from 1 to 65535.
ipsweep	This sets the max ICMP packets sent from the same one IP to multiple hosts within the specified interval. When packet number reaches this threshold, it believes that addresses are scanned for one time.
<*number2*>	This is one number, ranging from 1 to 65535.
synflood	This sets the max connection requests initiated to protected object per second.
<*number3*>	This is one number, 500 by default, ranging from 1 to 65535.
udpflood	This sets the max UDP packets sent to protected object per second. When the packet number reaches this threshold, UDP flooding attack protection function is enabled.
<*number4*>	This is one number, 1000 by default, ranging from 1 to 65535.

Parameter	Description
portscan	This sets the max IP packets containing TCP SYN segment sent from the same one source IP to multiple ports of destination IP within the specified interval. When packet number reaches this threshold, it believes that ports are scanned for one time.
<*number5*>	This is one number, ranging from 1 to 65535.
log	When attack event occurs, it sets whether to record it into log.
yes|*no*	yes: Record the event into log; no: Don't record the event into log.
action	It sets whether to permit packets to pass through.
pass|*block*	pass: It indicates permitting packets to pass through; block: It indicates denying packets passing through.

2. Modifying intrusion detection rule.

Command	Function
`ZXR10_FW.ips #`**dos rule modify ruleid** <*string*> **stattype** <*synflood*|*udpflood*|*icmpflood*|*portscan*|*ipsweep*> **threshold** <*number*>[**log** <*yes*|*no*>][**action** <*pass*|*block*>]	This modifies intrusion detection rule.

Parameter Description:

Parameter	Description
modify	This modifies intrusion detection rule.
ruleid	This sets ID of the rule to be modified. **dos rule show** can be used to view id of each rule.
<*string*>	This is an ID string.
stattype	This sets statistics type of rule to be modified.
synflood|*udpflood*|*icmpflood*|*portscan*|*ipsweep*	This is the statistics type. User can give choice according to demands.
threshold	This sets threshold of statistics type.
<*number*>	This is one number, which is the threshold.

3. Moving intrusion detection rule.

Command	Function
ZXR10_FW.ips #**dos rule move id** <*number1*> **before** <*number2*>	This moves intrusion detection rule.

Parameter Description:

Parameter	Description
<*number1*>	This is one number, indicating ID of the rule to be modified.
<*number2*>	This is one number, indicating ID of the rule to be referred to.

4. Deleting intrusion detection rule.

Command	Function
ZXR10_FW.ips #**dos rule delete id** <*string*>	This deletes intrusion detection rule.

Parameter Description:

Parameter	Description
<*string*>	This is one string, indicating id of one rule.

5. Clearing all intrusion detection rules.

Command	Function
ZXR10_FW.ips #**dos rule clean**	This clears all intrusion detection rules.

6. Viewing all intrusion detection rules.

Command	Function
ZXR10_FW.ips #**dos rule show**	This views all intrusion detection rules.

7. Clearing all configurations of intrusion detection.

Command	Function
ZXR10_FW.ips #**dos clean**	This clears all configurations of intrusion detection.

8. Showing all configurations of intrusion detection.

Command	Function
ZXR10_FW.ips #**dos config show**	This shows all configurations of intrusion detection.

9. Adding prevension type.

Command	Function
ZXR10_FW.ips #**dos type add** [**abntype** <*land* |*smurf* |*pingofdeath* |*winnuke* |*tcp_sscan* |*ip_option* |*teardrop* |*targa3* |*ipspoof*>]|[**stattype** <*synflood*|*udpflood* |*icmpflood*|*portscan*|*ipsweep*>]	This adds prevention type.

Parameter Description:

Parameter	Description
add	This adds prevention type.
abntype	This sets the type of abnormal packet attack.
land |*smurf* |*pingofdeath* |*winnuke* |*tcp_sscan* |*ip_option* |*teardrop* |*targa3* |*ipspoof*	This shows various types of abnormal packet attack.

10. Deleting prevention type.

Command	Function
ZXR10_FW.ips #**dos type delete** [**abntype** <*land* |*smurf* |*pingofdeath* |*winnuke* |*tcp_sscan* |*ip_option* |*teardrop* |*targa3* |*ipspoof*>]|[**stattype** <*synflood* |*udpflood*|*icmpflood*|*portscan*|*ipsweep*>]	This deletes prevention type.

Parameter Description:

Parameter	Description
delete	This deletes prevention type.

This page is intentionally blank.

Chapter 9

Load Balancing Configuration

Table of Contents:

Load Balancing Overview

High availability means some advanced characteristics of ZXR10 8900 Series Switch FW, including:

Server load balancing. It mainly describes how to define load balancing server and load balancing group, and how to meet user access demands through load balancing.

ZXR10 8900 Series Switch FW can implement load balancing of user server according to user demands and flexible load balancing algorithm so as to guarantee effectiveness of user critical services. ZXR10 8900 Series Switch FW supports session-based load balancing.

There are three ways to realize ZXR10 8900 Series Switch FW server load balancing function:

1. Defining server
2. Defining load balancing group
3. Defining NAT rule

Configuring Load Balancing

Configuring Load Balancing Server

This topic describes configuration commands and examples of load balancing.

User can add, modify or modify and delete server in FW server management. Server here is mainly used for FW load balancing function.

To access this command module, execute the following command:

`ZXR10_FW #`**define**

To exit from this command module, execute the following command:

`ZXR10_FW #`**end**

1. Adding one server.

Command	Function
`ZXR10_FW.define #`**server add name** <*string1*> **host** <*string2*>[**weight**<*number1*>][**probe** <*none*|*host*|*service*>][**port** <*number2*>]	This adds one server.

Parameter Description:

Parameter	Description
add	This adds one service.
name	This sets name for the server.
<*string1*>	This is one string, indicating name of the server.
host	This sets the host used for server.
<*string2*>	This is a string, indicating the name of host.
weight	This sets weight of server.
<*number1*>	This is one number, indicating weight of server, ranging from 1 to 100.
probe	This sets whether to perform detection to server. There are three options: none, host and service.
none|*host*|*service*	None indicates no detection; host indicates host detection; service indicates service detection. User needs to set detection port when selecting service detection.
port	This sets detection port. Detection port needs being set only when service detection is selected.
<*number2*>	This is one number, indicating port id.

Command Illustration:

With detection, working status of server can be found, thus avoiding sending traffics to this server when the server is down or services are abnormal, which makes services requested by user fail to be responded. Two detection modes are available: host detection and service detection. Host detection is to verify if server is online through timely monitoring, which is realized by executing command **ping**. Service detection is to perform detection by selecting corresponding port according to services provided by server and establishing TCP connection, such as

HTTP port 80, FTP port 21 and customized special port. Generally, service detection can reflect actual working status of server.

2. Modifying one server.

Command	Function
ZXR10_FW.define #**server modify name** <*string1*>[**host** <*string 2*>][**weight**<*number1*>][**probe** <*none*|*host*|*service*>][**port** <*number2*>]	This modifies one server.

Parameter Description:

Parameter	Description
modify	This modifies one server.

3. Renaming one server.

Command	Function
ZXR10_FW.define #**server rename oldname** <*string1*> **newname** <*string2*>	This renames one server.

Parameter Description:

Parameter	Description
rename	This renames one server.
oldname	This specifies the name of server to be renamed.
<*string1*>	This is one string, indicating the name of server (the server name has been defined).
newname	This specifies new name for one server.
<*string2*>	This is one string, indicating new name of the server.

4. Deleting one server.

Command	Function
ZXR10_FW.define #**server delete** [**id** <*number*>][**name** <*string*>]	This deletes one server.

Parameter Description:

Parameter	Description
delete	This deletes one server.
id	This specifies ID of the server to be deleted.
<*number*>	This is one number, indicating ID of server.

Parameter	Description
name	This specifies the name of server to be deleted.
<*string*>	This is one string, indicating name of the server.

Command Illustration:

To delete server, it is available to delete the server according to server name, server id or both. However, in case server id and server name are inconsistent, server name shall apply. When no parameter is given, the server not quoted by policy is deleted.

5. This deletes all servers not quoted by policy.

Command	Function
ZXR10_FW.define #**server clean**	This deletes all servers not quoted by policy.

6. Showing all servers.

Command	Function
ZXR10_FW.define #**server show**	This shows all servers.

Configuring Load Balancing Group

This topic describes configuration commands and examples of load balancing group.

User can add, modify and delete load balancing group in management of FW load balancing group.

1. Adding one load balancing group.

Command	Function
ZXR10_FW.define #**virtual_server add name** <*string1*>[**server** <*string2*>][**balance**<*rr*|*wrr*|*lc*|*wlc*|*sh*|*dh*>][**backup** <*number*>]	This adds one load balancing group.

Parameter Description:

Parameter	Description
add	This adds one load balancing group.
name	This sets name for load balancing group.
<*string1*>	This is one string, indicating the name of load balancing group.
server	This sets the server contained in load balancing group.

Parameter	Description
<*string2*>	This is one string, indicating the name of server. As for multiple server names, space is used between each two server names and all server names are quoted with single quotes, such as 'server1 server2'.
balance	This sets load balancing mode. Six modes are available: rr, wrr, lc, wlc, sh and dh.
rr|*wrr*|*lc*|*wlc*|*sh*|*dh*	"rr" indicates selecting server in load balancing group sequentially; "wrr" indicates selecting load balancing server in sequence of weight; "lc" indicates selecting server according to response time, the faster the response time, the higher the priority; "wlc" indicates selecting server according to response time and weight; the shorter the response time and the larger the weight, the larger the priority; "sh" indicates selecting server with HASH query according to source address; "dh" indicates selecting server with HASH query according to destination address.
backup	This specifies ID of backup load balancing group. When all servers in one group stop providing services, data in servers of load balancing group will be backed up to the specified load balancing group.
<*number*>	This is one number, indicating ID of load balancing group.

Command Illustration:

During communication process, if one server is deleted from the load balancing group, connections on this server will not be disconnected and only after re-establishment of connection, can configuration of load balancing group take effect.

If connected server gets disconnected during communication, client will not be connected to another server in load balancing group unless re-establishing connection.

When load balancing mode is set to weight balance algorithm, the total number of connections is assigned according to weight value. Here weight value is not priority.

"dh" and "sh" algorithms proceed HASH query according to IP address. Only when source IP and destination IP are disperse, can connections be allocated to different servers averagely.

Load balancing server can be defined before defining load balancing group. For details, please refer to section **Configuring High Availability**.

2. Modifying one load balancing group.

Command	Function
`ZXR10_FW.define #`**virtual_server modify name** <*string1*>[**server** <*string2*>][**balance**<*rr*|*wrr*|*lc*|*wlc*|*sh*|*dh*>][**backup** <*number*>]	This modifies one load balancing group.

Parameter Description:

Parameter	Description
modify	This modifies one load balancing group.

3. Renaming load balancing group.

Command	Function
ZXR10_FW.define #**virtual_server rename oldname** <*string1*> **newname** <*string2*>	This renames load balancing group.

Parameter Description:

Parameter	Description
rename	This renames load balancing group.
oldname	This specifies the name of load balancing group to be renamed.
<*string1*>	This is one string, indicating the name of load balancing group (the name of load balancing group has been defined).
newname	This specifies new name for load balancing group.
<*string2*>	This is one string, indicating the new name of load balancing group.

4. Deleting one load balancing group.

Command	Function
ZXR10_FW.define #**virtual_server delete** [**id** <*number*>][**name** <*string*>]	This deletes one load balancing group.

Parameter Description:

Parameter	Description
delete	This deletes one load balancing group.
id	This specifies ID of the load balancing group to be deleted.
<*number*>	This is one number, indicating ID of load balancing group.
name	This specifies the name of load balancing group to be deleted.
<*string*>	This is one string, indicating the name of load balancing group.

Command Illustration:

To delete load balancing group, it is available to delete the load balancing group according to load balancing group name, load balancing group id or both. However, in case load balancing

group id and load balancing group name are inconsistent, load balancing group name shall apply.

When no parameter is given, the load balancing group not quoted by policy is deleted.

5. This deletes all load balancing groups not quoted by policy.

Command	Function
ZXR10_FW.define #**virtual_server clean**	This deletes all load balancing groups not quoted by policy.

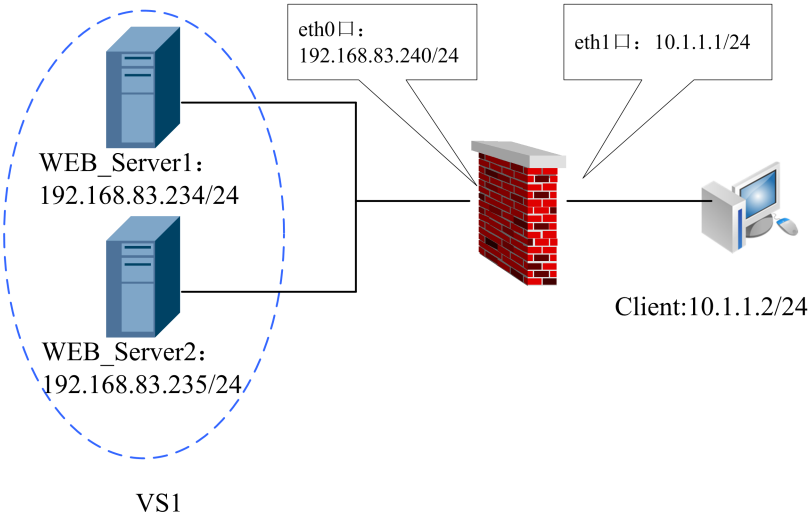
6. Showing all load balancing groups.

Command	Function
ZXR10_FW.define #**virtual_server show**	This shows all load balancing groups.

High Availability Configuration Example

The access traffic of web service provided by an enterprise is large, so this enterprise decides to use two WEB servers to provide web services: WebServer1 (IP 192.168.83.234) and WebServer2 (IP 192.168.83.235). Both two WEB servers use rr algorithm to provide service outwards through vlan1 interface (IP: 192.168.83.240) of FW. FW is connected with extranet through vlan interface (IP: 10.1.1.1). HTTP connection request coming from extranet is scheduled by way of polling.

FIGURE 7 HIGH AVAILABILITY CONFIGURATION EXAMPLE



Configuration Points:

- Adding routes on two web servers
- Configuring IP and GW on client host;
- Configuring FW interface attributes (IP addresses of areas that eth0 and eth1 belong to)
- Configuring host
- Configuring load balancing server
- Configuring load balancing group
- Configuring NAT rule
- Verifying if HTTP connection request can be scheduled by way of polling

1. Configuring FW interface attributes (areas and IP addresses that vlan1 and vlan2 belong to, set gei_3/1 to be in vlan1, connected with two webservers; set gei_3/2 to be in vlan2, connected with client).

   Configuration on main board:

   ```
   ZXR10(config)#vlan 2 ZXR10(config-vlan2)#exit
   ZXR10(config)#interface vlan 2
   ZXR10(config-if-vlan2)ip address 10.1.1.1 255.255.255.0
   ZXR10(config)#interface vlan 1
   ZXR10(config-if-vlan1)ip address192.168.83.240 255.255.255.0

   ZXR10(config)#vlan 1 ZXR10(config-vlan1)#exit
   ZXR10(config)#interface vlan 1
   ZXR10(config-if-vlan1)ip address 192.168.83.240 255.255.255.0
   ZXR10(config-if-vlan1)exit ZXR10(config)interface gei_3/1
   ZXR10(config-gei_3/1)switchport access vlan 1
   ZXR10(config-gei_3/1)exit

   ZXR10(config)interface gei_3/2
   ZXR10(config-gei_3/2)switchport access vlan 2
   ZXR10(config-gei_3/1)exit
   ```

   Configuration on FW:

   ```
   ZXR10_FW #define area add name area_vlan1
   attribute gei_3/1 access on
   ZXR10_FW #define area add name area_vlan2
   attribute gei_3/2 access on
   ```

2. Configuring host

   Configuring host WebServer1:

   ```
   ZXR10_FW #define host add name doc_server ipaddr
   192.168.83.234
   ```

   Configuring host WebServer2:

   ```
   ZXR10_FW #define host add name WebServer2 ipaddr
   192.168.83.235
   ```

   Configuring addresses on two web servers for accessing external network:

   ```
   ZXR10_FW #define host add name WebServer ipaddr
   192.168.83.219
   ```

3. Configuring load balancing server

   Configuring load balancing server S1:

```
ZXR10_FW #define server add name S1 host
WebServer1 weight 10 probe host
```

Configuring load balancing server S2:

```
ZXR10_FW#define server add name S2 host
WebServer2 weight 20 probe host
```

4. Configuring load balancing group:

```
ZXR10_FW #define virtual_server add name
VS1 server ’ S1 S2’ balance rr
```

5. Configuring NAT rule:

```
ZXR10_FW #nat policy add orig_src any orig_dst
WebServer orig_service HTTP trans_dst VS1 enable yes
```

6. Verifying if HTTP connection request can be scheduled by way of polling:

   Open IE browser, input "http://192.168.83.219", turn to Web-Server1 (IP192.168.83.234) page, as shown in Figure 8.

FIGURE 8 BACKING UP AND RESTORING USER INFORMATION

Due to setting of polling mechanism, when refreshing the page, it turns to WebServer2 (IP: 192.168.83.235) page, as shown in Figure 9.

FIGURE 9 DOCUMENT GROUP SERVER

Notes:

- During configuration process, make sure that no NAT policy and block policy conflict with this rule.
- In communication, if one server is deleted from the balancing group, connections on this server will not be disconnected and configuration can get effective only after re-connection.
- In communication, if connected server gets disconnected, client will not be connected with other active server unless re-connection.
- When host sends ping packets, if service being accessed is disabled but host still works, the host will still be assigned with connections. Since service is unavailable, host keeps in the status of failing to be connected.

Chapter 10

Log and Alarm Configuration

Table of Contents:

Log and Alarm Overview

To debug, monitor and manage ZXR10 8900 Series Switch FW service card FW module conveniently, ZXR10 8900 Series Switch FW service card FW module provides log management and alarm function for user. Log and alarm function has three parts:

- Log configuration
- Log Viewing
- Alarm

Log Configuration

ZXR10 8900 Series Switch FW service card FW module provides all-around logging and alarm service functions, which is convenient for user tracing working status of ZXR10 8900 Series Switch FW service card FW module. ZXR10 8900 Series Switch FW service card FW module can record log in WELF format, transmit the log to preset log server over Syslog protocol, and use the third-party software to perform statistics and analysis to log.

Viewing Log

ZXR10 8900 Series Switch FW service card FW module can buffer partial log data according to performance of hardware device. This is convenient for user to view system log and trace working status of ZXR10 8900 Series Switch FW service card FW module timely.

Alarms

ZXR10 8900 Series Switch FW service card FW module has comprehensive alarm prompt function, supports mail alarming, voice alarming, console alarming and other alarming modes. Firstly, administrator needs to add alarm rules and set alarm objects and parameters. Then set security events triggering alarm rules, including device faults and administrator predefined security events. When security event occurs, FW will trigger corresponding alarm message according to rule.

Configuring Logs and Alamrs

Configuring Log

This topic describes how to configure log.

To access this command module, execute the following command:

#**log**

To exit from this command module, execute the following command:

#**end**

1. Setting log server.

Command	Function
ZXR10_FW.log #**log set ipaddr** <*ipaddress*>[**port** <*string*>][**logtype** <*syslog*|*welf*>][**trans**<*enable*|*disable*>]	This sets log server.

Parameter Description:

Parameter	Description
set	This sets log server.
ipaddr	This set IP address of log server.
<*ipaddress*>	This is one string, indicating IP address.
port	This sets port of log server.
<*string*>	This is one string, indicating port id, in format of tcp:80 or udp:80.
logtype	This sets log transmission format: syslog or welf
syslog|*welf*	This is log transmission format, syslog by default.

Parameter	Description
trans	This sets whether to transmit log.
enable|*disable*	Enables indicates transmitting log and disable indicates not transmitting log.

Command Illustration:

Generated log data needs to be managed by log server. Log server can be any reachable network node (such as PC). This command can set IP address, port id, transmission format and other information of log server. By default, IP address of log server is 192.168.1.253, protocol and port id is udp: 514, log transmission format is syslog, and log is not transmitted.

2. Setting log level.

Command	Function
ZXR10_FW.log #**log level_set** <*number*>	This sets the levels at which logs are transmitted to log server.

Parameter Description:

Parameter	Description
<*number*>	This is one number, indicating log level, ranging from 0 to 8, where 0 indicates serious errors that cause unavailability of system, 1 indicates alarm messages, 2 indicates errors that cause unavailability of partial system functions, 3 indicates common error messages, 4 indicates all attacks and unauthorized accesses (except for communication log), 5 indicates operation record of administrator, 6 indicates common event record, 7 indicates debugging information of developer, and 8 indicates diagnosis log.

3. Adding log type transmitted to log server.

Command	Function
ZXR10_FW.log #**log type_set add** <*string*>	This adds log type transmitted to log server.

Parameter Description:

Parameter	Description
<*string*>	This is one string, indicating log type, including: mgmt, system, pf, conn, ac, secure, dpi, vpn, avse, sslvpn_conn, sslvpn_admin, sslvpn_system, all or none.

4. Deleting log type transmitted to log server.

Command	Function
ZXR10_FW.log #**log type_set delete** <*string*>	This deletes log type transmitted to log server.

Example

1. To set log server to 192.168.1.25, protocol and port to TCP: 524, log transmission type to syslog, and permit log transmission, execute the following command:

   ```
   ZXR10_FW.log # log set ipaddr 192.168.1.25 port tcp:524
   logtype syslog trans enable
   ```

   To transmit level 5 (and below) log to log server, execute the following command:

   ```
   ZXR10_FW.log # log level_set 5
   ```

2. To set log server on each FW where log analysis is needed, execute the following command:

   ```
   ZXR10_FW.log # log set ipaddr 10.200.2.111 port udp:514
   logtype syslog trans enable
   ```

   To set log level, execute the following command:

   ```
   ZXR10_FW.log # log level_set 6
   ```

   To set log type to System Running, execute the folloing command:

   ```
   ZXR10_FW.log # log type_set add system
   ```

   To show System Running log, execute the following command:

   ```
   ZXR10_FW.log # log show keyword system from 1 to 10
   ```

Viewing Log

This topic describes how to view log.

1. Viewing the total number of logs.

Command	Function
ZXR10_FW.log #**log count**	This views the total number of logs.

2. Viewing configuration information of log server.

Command	Function
ZXR10_FW.log #**log set_show**	This views configuration information of log server.

3. Viewing log information.

Command	Function
ZXR10_FW.log #**log show from**<*number1*> **to** <*number2*>[**keyword** <*string*>]	This views log information.

Parameter Description:

Parameter	Description
show	This views log information.
from	This sets from which log to view.
<*number1*>	This is one number, indicating the number of log.
to	This sets to which log to view.
<*number2*>	This is one number, indicating the number of log.
keyword	This is optional. It sets the keyword in log to view.
<*string*>	This is one string, indicating keyword.

Example To view the total number of logs, execute the following command:

```
ZXR10_FW.log # log count Total log : 351
```

To view information of no. 10 to no. 100 logs (containing string "log"), execute the following command:

```
ZXR10_FW.log # log show from 10 to 100 keyword log
```

Configuring Alarms

This topic describes how to manage alarming mode.

It is available to trigger alarms according to predefined policies and security alarm events and send alarm messages to administrator by way of mail alarm. User can set related parameters here.

1. Setting event alarming modes according to event types.

Command	Function
ZXR10_FW.log #**alarmevent set** <*manage*|*system*|*security*|*policy*|*communication*|*hardware*|*recover*|*noticetest*|*all*> **noticeid** <*number*>| **noticename** <*string*>| **notice** <*empty*>	This sets event alarming modes according to event types.

Parameter Description:

Parameter	Description
set	This sets event alarming mode.
<*manage*|*system*|*security*|*policy*|*communication*|*hardware*|*recover*|*noticetest*|*all*>	Manage indicates management system alarm, system indicates system alarm, security indicates security alarm, policy indicates policy alarm, communication indicates communication alarm, hardware indicates hardware alarm, recover indicates recover alarm, noticetest indicates test alarm and all indicates alarm of all events.

Parameter	Description
noticeid	This sets id of alarming mode.
<*number*>	This is one number, indicating ID of alarming mode.
noticename	This sets name of alarming mode.
<*string*>	This is one string, indicating name of alarming mode.
notice	This specifies that no alarm event rule is contained in alarming mode and removes all alarm events.
<*empty*>	This indicates not generating alarms.

2. Showing an alarm event.

Command	Function
ZXR10_FW.log #**alarmevent show** <*manage*|**system**|**security**|**policy**|**communication**|**hardware**|**recover**|**noticetest**|**all**>	This shows an alarm event.

3. Showing all alarm events in system.

Command	Function
ZXR10_FW.log #**alarmevent show**	This shows all alarm events in system.

4. Adding one mail alarm.

Command	Function
ZXR10_FW.log #**alarmnotice add** <*mail*> **name** <*string1*> **srvaddr** <*ipaddress*> **srvport** <*number*> **mailaddr** <*string2*>[**subject** <*string3*>]	This adds one mail alarm.

Parameter Description:

Parameter	Description
add	This adds one mail alarm.
<*mail*>	This indicates the alarming mode is mail alarm.
name	This sets alarm name.
<*string1*>	This is a string, indicating the name of alarm.
srvaddr	This sets IP address for SMTP server that is to send mail.
<*ipaddress*>	This is one string, indicating IP address, in format of A.B.C.D.

Parameter	Description
srvport	This sets port of SMTP server.
<*number*>	This is one number, indicating port id.
mailaddr	This sets mail account that receives alarm mail. `
<*string2*>	This is one string, indicating ip address of mail address, such as abc@topsec.com.cn.
subject	This sets the subject of alarm mail.
<*string3*>	This is one string, indicating subject of mail.

5. Modifying one mail alarm-related parameter.

Command	Function
ZXR10_FW.log #**alarmnotice modify** <*mail*> **name** <*string1*> **srvaddr** <*ipaddress*> **srvport** <*number*> **mailaddr** <*string2*>[**subject** <*string3*>]	This modifies one mail alarm-related parameter.

Parameter Description:

Parameter	Description
modify	This modifies one mail alarm.
<*mail*>	This indicates the alarming mode is mail alarm.

6. Modifying authentication attribute of mail alarming mode.

Command	Function
ZXR10_FW.log #**alarmnotice modify** <*mail*> **name** <*string1*> **auth** <*on*|**off**> **username** <*string2*> **password** <*string3*>	This modifies authentication attribute of mail alarming mode: if authentication is needed for server.

Parameter Description:

Parameter	Description
modify	This modifies mail alarm.
<*mail*>	This indicates the alarming mode is mail alarm.
mail	This specifies the name of alarm to be modified.
<*string1*>	This is a string, indicating the name of alarm.
auth	This specifies if the mail server of mail alarm needs authentication.

Parameter	Description
on|**off**	Off indicates authentication is not needed and when selecting this option, it doesn't need to set the following parameters; on indicates authentication is needed.
username	This is the username of authentication on mail server.
<*string2*>	This is a string, indicating username.
password	This is the password of authentication on mail server.
<*string3*>	This is a string, indicating user password.

7. Deleting one alarm.

Command	Function
`ZXR10_FW.log #`**alarmnotice delete name** <*string*>| **id** <*number*>	This deletes one alarm.

Parameter Description:

Parameter	Description
<*string*>	This is a string, indicating the name of alarm.
<*number*>	This is one number, indicating ID.

Command Illustration:

Only when alarm event of this alarm rule is null, can the alarm be deleted, or it will prompt error.

8. Clearing all alarm rules.

Command	Function
`ZXR10_FW.log #`**alarmnotice clean**	This clears all alarm rules. Only when alarm events contained in alarm rules are null, can alarm rules be deleted.

9. Showing alarms.

Command	Function
`ZXR10_FW.log #`**alarmnotice show** [**name** <*string*>]|[**id** <*number*>]	This shows alarms.

Parameter Description:

Parameter	Description
<*string*>	This is a string, indicating the name of alarm.
<*number*>	This is one number, indicating ID.

10. Testing one alarm.

Command	Function
ZXR10_FW.log #**alarmnotice test**	This tests one alarm.

Command Illustration:

For convenience of user, FW provides alarm testing function. User can verify effectiveness of alarm rules through testing after successfully adding alarming mode and setting alarm-triggered security event.

Example To add alarm mail1 sent to user@zte.com.cn and set ip address of SMTP mail server to 192.168.1.2, port id to 25 and subject of alarm mail to "Mail Alarm", execute the following command:

```
ZXR10_FW.log # alarmnotice add mail name mail1 srvaddr
192.168.1.2 srvport 25 mailaddr user@zte.com.cn subject Mail alarm
```

To change the destination mail address to user2@zte.com.cn, execute the following command:

```
ZXR10_FW.log # alarmnotice modify mail name mail1 srvaddr
192.168.1.2 srvport 25 mailaddr user2@zte.com.cn subject Mail alarm
```

To change mail1 to need authentication and set authentication username/password to user/user, execute the following command:

```
ZXR10_FW.log # alarmnotice modify mail name mail1 auth
on username user password user
```

To delete mail1, execute the following command:

```
ZXR10_FW.log # alarmnotice delete name mail1
```

To show the alarm named mail1, execute the following command:

```
ZXR10_FW.log # alarmnotice show name mail1
```

To show all alarms, execute the following command:

```
ZXR10_FW.log # alarmnotice show
```

This page is intentionally blank.

Figures

This page is intentionally blank.

Tables

This page is intentionally blank.

Glossary

DPI - Deep Packet Inspection

FTP - File Transfer Protocol

IP - Internet Protocol

IPv4 - Internet Protocol version 4

MAC - Media Access Control

NAT - Network Address Translation

NTP - Network Time Protocol

PPTP - PPP Tunnel Protocol

RSTP - Rapid Spanning Tree Protocol

TELNET - Telecommunication Network Protocol

TFTP - Trivial File Transfer Protocol

VLAN - Virtual Local Area Network

VPN - Virtual Private Network