JUNIPER NETWORKS

Junos® OS

HTTP Redirect Feature Guide for Subscriber Services

Release
14.1

Published: 2014-04-25

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Junos® OS HTTP Redirect Feature Guide for Subscriber Services
14.1

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at http://www.juniper.net/support/eula.html. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

List of Tables

About the Documentation

Documentation and Release Notes

To obtain the most current version of all Juniper Networks® technical documentation, see the product documentation page on the Juniper Networks website at http://www.juniper.net/techpubs/.

If the information in the latest release notes differs from the information in the documentation, follow the product Release Notes.

Juniper Networks Books publishes books by Juniper Networks engineers and subject matter experts. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration. The current list can be viewed at http://www.juniper.net/books.

Supported Platforms

For the features described in this document, the following platforms are supported:

- MX Series

Using the Examples in This Manual

If you want to use the examples in this manual, you can use the **load merge** or the **load merge relative** command. These commands cause the software to merge the incoming configuration into the current candidate configuration. The example does not become active until you commit the candidate configuration.

If the example configuration contains the top level of the hierarchy (or multiple hierarchies), the example is a *full example*. In this case, use the **load merge** command.

If the example configuration does not start at the top level of the hierarchy, the example is a *snippet*. In this case, use the **load merge relative** command. These procedures are described in the following sections.

Merging a Full Example

To merge a full example, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration example into a text file, save the file with a name, and copy the file to a directory on your routing platform.

   For example, copy the following configuration to a file and name the file **ex-script.conf**. Copy the **ex-script.conf** file to the **/var/tmp** directory on your routing platform.

   ```
   system {
     scripts {
       commit {
         file ex-script.xsl;
       }
     }
   }
   interfaces {
     fxp0 {
       disable;
       unit 0 {
         family inet {
           address 10.0.0.1/24;
         }
       }
     }
   }
   ```

2. Merge the contents of the file into your routing platform configuration by issuing the **load merge** configuration mode command:

   ```
   [edit]
   user@host# load merge /var/tmp/ex-script.conf
   load complete
   ```

Merging a Snippet

To merge a snippet, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration snippet into a text file, save the file with a name, and copy the file to a directory on your routing platform.

   For example, copy the following snippet to a file and name the file **ex-script-snippet.conf**. Copy the **ex-script-snippet.conf** file to the **/var/tmp** directory on your routing platform.

   ```
   commit {
     file ex-script-snippet.xsl; }
   ```

2. Move to the hierarchy level that is relevant for this snippet by issuing the following configuration mode command:

```
[edit]
user@host# edit system scripts
[edit system scripts]
```

3. Merge the contents of the file into your routing platform configuration by issuing the **load merge relative** configuration mode command:

```
[edit system scripts]
user@host# load merge relative /var/tmp/ex-script-snippet.conf
load complete
```

For more information about the **load** command, see the *CLI User Guide*.

Documentation Conventions

Table 1 on page ix defines notice icons used in this guide.

Table 1: Notice Icons

Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.
	Tip	Indicates helpful information.
	Best practice	Alerts you to a recommended use or implementation.

Table 2 on page ix defines the text and syntax conventions used in this guide.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the **configure** command: user@host> **configure**

Table 2: Text and Syntax Conventions *(continued)*

Convention	Description	Examples
`Fixed-width text like this`	Represents output that appears on the terminal screen.	`user@host>` **show chassis alarms** `No alarms currently active`
Italic text like this	• Introduces or emphasizes important new terms. • Identifies guide names. • Identifies RFC and Internet draft titles.	• A policy *term* is a named structure that defines match conditions and actions. • *Junos OS CLI User Guide* • RFC 1997, *BGP Communities Attribute*
Italic text like this	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# **set system domain-name** *domain-name*
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	• To configure a stub area, include the **stub** statement at the **[edit protocols ospf area area-id]** hierarchy level. • The console port is labeled **CONSOLE**.
< > (angle brackets)	Encloses optional keywords or variables.	**stub <default-metric** ***metric*****>;**
| (pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	**broadcast | multicast** **(*string1* | *string2* | *string3*)**
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	**rsvp { # Required for dynamic MPLS only**
[ ] (square brackets)	Encloses a variable for which you can substitute one or more values.	**community name members [** ***community-ids*** **]**

Indention and braces ({ })	Identifies a level in the configuration hierarchy.	[edit]
		routing-options {
		  static {
		  route default {
		  nexthop *address*;
		  retain;
		  }
		  }
		}

; (semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
GUI Conventions		
Bold text like this	Represents graphical user interface (GUI) items you click or select.	• In the Logical Interfaces box, select **All Interfaces**. • To cancel the configuration, click **Cancel**.

Table 2: Text and Syntax Conventions *(continued)*

Convention	Description	Examples
> (bold right angle bracket)	Separates levels in a hierarchy of menu selections.	In the configuration editor hierarchy, select **Protocols>Ospf**.

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can send your comments to **techpubs-comments@juniper.net**, or fill out the documentation feedback form at https://www.juniper.net/cgi-bin/docbugreport/. If you are using e-mail, be sure to include the following information with your comments:

- Document or topic name
- URL or page number
- Software release version (if applicable)

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or JNASC support contract, or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf.
- Product warranties—For product warranty information, visit http://www.juniper.net/support/warranty/.
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: http://www.juniper.net/customers/support/
- Search for known bugs: http://www2.juniper.net/kb/
- Find product documentation: http://www.juniper.net/techpubs/
- Find solutions and answer questions using our Knowledge Base: http://kb.juniper.net/
- Download the latest versions of software and review release notes: http://www.juniper.net/customers/csc/software/

- Search technical bulletins for relevant hardware and software notifications: http://kb.juniper.net/InfoCenter/
- Join and participate in the Juniper Networks Community Forum: http://www.juniper.net/company/communities/
- Open a case online in the CSC Case Management tool: http://www.juniper.net/cm/

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: https://tools.juniper.net/SerialNumberEntitlementSearch/

Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Management tool in the CSC at http://www.juniper.net/cm/.
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, see http://www.juniper.net/support/requesting-support.html.

PART 1

Overview

CHAPTER 1

HTTP Redirect

Redirecting HTTP Requests Overview

HTTP request traffic from subscribers is aggregated from access networks onto a Broadband Remote Access Server (B-RAS) router, where HTTP traffic can be intercepted and redirected to a captive portal. A captive portal provides authentication and authorization services for redirected subscribers before granting access to protected servers outside of a walled garden. A walled garden defines a group of servers where access is provided to subscribers without reauthorization through a captive portal. You can use a captive portal page as the initial page a subscriber sees after logging in to a subscriber session and as a page used to receive and manage HTTP requests to unauthorized Web resources.

The HTTP redirect service implements a data handler and a control handler and registers them with service rules applicable to the HTTP applications. These rules are parsed by the captive-portal-content-delivery process on the routing engine. The data handler applies the rules to HTTP data flows and handles rewriting the IP destination address or sending an HTTP 302 response with a preconfigured redirect URL. In addition, the control handler maintains a connection with the captive-portal-content-delivery process on the routing engine to learn configuration changes, such as the redirect URL and the rewrite IP destination and port pair. To achieve faster performance, the control handler maintains a cache of relevant configured entities, such as URLs on Multiservices DPC.

Packet flow differs depending on the following configurations:

- Walled garden as a service filter—HTTP traffic destined to servers within the walled garden does not flow to Multiservices DPC. However, any HTTP traffic destined outside of the walled garden flows to the Multiservices DPC.
- Walled garden as an HTTP policy term—All HTTP traffic flows to the Multiservices DPC. The HTTP service handler determines whether traffic is allowed to go to a walled garden.
- HTTP request packet—If the flow is destined to servers within the walled garden, no action is taken.

An HTTP redirect service can be attached to either a static or dynamic interface. For dynamic subscriber management, HTTP services can be attached dynamically at subscriber login or by using a change of authorization (CoA).

Redundant multiservice PIC and DPC support for HTTP redirect distributes captive portal content delivery rules to both PICs to leverage all framework support (for IPv4 only). Data traffic is sent only to the active PIC and rule processing is performed on the active PIC.

Related Documentation

- *Configuring a Basic Dynamic Profile*
- *Configuring a Dynamic Profile for Various Levels of Services*
- *Junos OS Predefined Variables*
- *Associating Service Sets with Interfaces in a Dynamic Profile*

Remote HTTP Redirect Server Operation Flow

You can use the remote HTTP redirect feature in configurations where the redirect server resides outside of the router and on a policy server, such as Session and Resource Control (SRC).

An HTTP redirect remote server that resides in a walled garden behind routers processes HTTP requests redirected to it and responds with a redirect URL to a captive portal. When you use a remote HTTP redirect server, you need to configure an HTTP service rule to rewrite the IP-DA of the incoming HTTP requests on the service router so that the requests reach the remote HTTP redirect server before being redirected to a captive portal.

The following general sequence occurs during access configuration for a remote HTTP redirect server deployment:

1. The subscriber logs in.
2. RADIUS authenticates the subscriber and sends a service activate (IP-DA rewrite), which redirects traffic to the redirect policy server in a walled garden.
3. The subscriber attempts to access the content server.
4. The router first redirects the HTTP traffic to SRC, which redirects it to the captive portal.
5. The captive portal sends an authorization page back to the subscriber.
6. The subscriber enters credentials to obtain authorization.
7. The captive portal verifies the subscriber credentials.
8. The captive portal authorizes the subscriber and notifies SRC.
9. SRC checks the subscriber database and formulates a policy to allow the subscriber access to the content server.

10. SRC sends the policy directly to the router or notifies the RADIUS server, which in turn sends a change of authorization (CoA) to the router.
11. The router attaches the new policy, overriding the initial IP-DA write.

The subscriber now has access to the content server.

The following example shows a configuration for IP-DA rewrite:

```
[edit services captive-portal-content-delivery]
rule ipda-rewrite {
  match-direction input-output;
  term 1 {
    from {
      applications http {
        destination-port 80;
      }
    }
    then {
      rewrite destination-address 100.20.1.2;
    }
  }
}
```

Related Documentation

- Local HTTP Redirect Server Operation Flow on page 5

Local HTTP Redirect Server Operation Flow

You can use the local HTTP redirect feature in configurations where the redirect server resides locally on the router.

An HTTP redirect local server that resides locally on a router processes HTTP requests redirected to it and responds with a redirect URL to a captive portal. You can implement the local server as a service within a service set, which provides more scalability and better performance. When you use a local HTTP redirect server, you need to configure an HTTP service rule to redirect HTTP requests to a captive portal within a walled garden.

The following general sequence occurs during access configuration for a local HTTP redirect server deployment:

1. The subscriber logs in.
2. RADIUS authenticates the subscriber and sends a service activate (HTTP redirect), which redirects HTTP traffic to the captive portal in a walled garden.
3. The subscriber attempts to access the content server (HTTP traffic).
4. The subscriber's HTTP traffic is redirected to the captive portal by the router.
5. The captive portal sends an authorization page back to the subscriber.
6. The subscriber enters credentials to obtain authorization.
7. The captive portal verifies the subscriber credentials.
8. The captive portal authorizes the subscriber.

The subscriber now has access to the content server.

The following example shows a configuration for HTTP redirect:

```
[edit services captive-portal-content-delivery]
rule redirect {
   match-direction input-output
   term 1 {
  from {
     applications junos-http;
  }
  then {
     redirect http://100.20.2.10/index.html; # this is the captive portal page       }
   }
}
```

Related Documentation

- Remote HTTP Redirect Server Operation Flow on page 4

PART 2

Configuration

CHAPTER 2

Configuration Tasks

Configuring HTTP Redirect Services

You can configure a walled garden with services and policies.

To configure the HTTP redirect service:

1. Configure the packet and installation.

```
[edit chassis]
fpc 1 {
  pic 0 {
    adaptive-services {
      service-package {
        extension-provider {
          control-cores 1;
          data-cores 7;
          object-cache-size 1024;
          policy-db-size 64;
          package jservices-cpcd;
          syslog {
            daemon any;
            external any;
          }
        }
      }
    }
  }
}
```

2. Configure the units and assign the VLAN IDs.

```
[edit interfaces]
ge-0/0/1 {
  vlan-tagging;
  unit 1 {
    vlan-id 100;
    family inet {
      address 100.20.1.1/24;
    }
  }
```

```
}
```

3. Configure the policy options.

```
policy-options {
  prefix-list google {
    74.125.19.0/24;
  }
}
```

4. Configure the service options.

```
firewall {
  family inet {
    service-filter walled {
      term google {
        from {
          destination-prefix-list {
            google;
          }
        }
        then skip;
      }
      term http {
        from {
          destination-port [ 80 8080 443 ];
        }
        then service;
      }
      term skip {
        then skip;
      }
    }
    service-filter fromSRC {
      term SRC {
        from {
          source-address {
            10.1.2.3/32;
          }
          source-port 8800;
        }
        then service;
      }
      term skip {
        then skip;
      }
    }
    service-filter test {
      term t1 {
        from {
          protocol icmp;
        }
        then service;
      }
    }
  }
}
```

5. Configure the captive portal content delivery services.

```
services {
  captive-portal-content-delivery {
    rule test {
      match-direction input;
      term t1 {
        then {
          rewrite;
        }
      }
    }
    profile ipda-rewrite {
      cpcd-rules test;
      ipda-rewrite-options {
        destination-address 10.1.2.3;
        destination-port 8800;
      }
    }
    traceoptions {
      file cpcdd;
      flag all;
    }
  }
  service-set sset1 {
    captive-portal-content-delivery-profile ipda-rewrite;
    interface-service {
      service-interface ms-1/0/0;
    }
  }
  stateful-firewall {
    rule Rule1 {
      match-direction input-output;
      term 1 {
        from {
          applications [ junos-icmp-all junos-dhcp-server junos-tftp junos-http ];
        }
        then {
          accept;
        }
      }
      term 2 {
        from {
          applications SRC;
        }
        then {
          accept;
        }
      }
    }
  }
}
```

6. Configure the applications.

```
applications {
  application SRC {
```

```
        protocol tcp;
        destination-port 8800;
    }
}
```

Related Documentation

- Redirecting HTTP Requests Overview on page 3

CHAPTER 3

Examples

Example: Walled Garden as a Service Filter

Service filters are configured under the firewall and are not specific to captive portal content delivery. The following example shows a walled garden with one server, which is the captive portal:

```
[edit firewall family inet]
root@host# show
service-filter walled {
   term 1 {
      from {
         destination-address {
            100.20.2.3/32; ## this is the address of captive portal
         }
         destination-port 80;
      }
      then skip; ## skip service DPC for http traffic
      ## destined to captive portal
   }
}
```

The following example shows a walled garden within a subnet:

```
service-filter walled-net {
   term 2 {
      from {
         destination-prefix-list {
            100.20.2.0/24; ## '100.20.2.0/24' is not defined
         }
      }
      then skip;
   }
```

```
}
```

The following example shows the configuration of an IPv6 walled garden:

```
[edit services captive-portal-content-delivery]
  rule walled-garden {
  match-direction input-output
  term 1 {
    from {
      destination-address 2001:2002:0:1::/64; ## captival portal resides here
      destination-port 80;
    }
    then {
      accept;
    }
  }
}
```

Related Documentation

- Redirecting HTTP Requests Overview on page 3

Example: Walled Garden as an HTTP Service Rule

HTTP service rule configuration resides under the services hierarchy and uses the captive portal and content delivery (captive-portal-content-delivery) service. The following example shows a walled garden configured as an HTTP service rule:

```
[edit services captive-portal-content-delivery]
rule walled-garden {
  match-direction input-output
  term 1 {
    from {
      destination-address 100.20.2.3/32; ## captive portal
      destination-port 80;
    }
    then {
      accept;
    }
  }
}
```

When a remote HTTP redirect server is used, you need to configure an HTTP service rule to rewrite the IP-DA of incoming HTTP requests on the service router so that the requests reach the remote HTTP redirect server before being redirected to a captive portal. If the destination port is not specified, the default behavior is determined by the rewrite configuration. If no rewrite configuration is available, the destination port is not rewritten. The following example shows a configuration for IP-DA rewrite:

```
[edit services captive-portal-content-delivery]
rule ipda-rewrite {
  match-direction input-output;
  term 1 {
    from {
      applications junos-http;
    }
```

```
        then {
          rewrite destination-address 100.20.2.10; # this is the remote
          # redirect server.
        }
      }
    }
```

Related Documentation

- Redirecting HTTP Requests Overview on page 3

Example: HTTP Service Attached to a Static Interface

The following example shows an HTTP service set attached to a static interface:

```
[edit interfaces ge-1/0/1]
root@hostr# show
unit 0 {
  family inet {
    service {
      input {
        service-set http-redirect-walled;
      }
      output {
        service-set http-redirect-walled;
      }
    }
    address 10.1.3.2/24;
  }
}
```

The following example uses a service filter as a walled garden by defining a rule named redirect, referencing the rule in a profile named http-redirect, configuring a service set named http-redirect that references the http-redirect captive portal content delivery profile, and attaching the http-redirect service set to static interface ge-1/0/1.0.

```
[edit services]
captive-portal-content-delivery {
  rule redirect {
    match-direction input;
    term t1 {
      from {
        destination-address {
          100.0.1.1/32;
        }
      }
      then {
        redirect http://www.google.com;
      }
    }
  }
  profile http-redirect {
    cpcd-rules redirect;
  }
}
service-set http-redirect {
```

```
    captive-portal-content-delivery-profile http-redirect;
    interface-service {
      service-interface ms-1/0/0;
    }
  }
```

```
  [edit interfaces ge-1/0/1]
  unit 0 {
    family inet {
      service {
        input {
          service-set http-redirect service-filter walled;
        }
        output {
          service-set http-redirect;
        }
      }
      address 10.1.3.2/24;
    }
  }
```

The following example shows an IPv6 static service attachment:

```
  [edit interfaces ge-1/0/1]
  unit 0 {
    family inet6 {
      service {
        input {
          service-set http-redirect6 service-filter walled6;
        }
        output {
          service-set http-redirect6 service-filter walled6;
        }
      }
      address 2001:2002::1;
    }
  }
```

This example configures the service filter for walled6:

```
  firewall {
    family inet6 {
      service-filter walled6 {
        term google {
          from {
            destination-prefix-list {
              google6;
            }
          }
          then skip;
        }
        term http {
          from {
            destination-port [ 80 8080 443 ];
          }
          then service;
        }
```

```
        term skip {
          then skip;
        }
      }
    }
  }
```

Related Documentation

- Redirecting HTTP Requests Overview on page 3

Example: HTTP Service Attached to a Dynamic Interface

A dynamic service attachment uses a dynamic profile. In the following dynamic profile example, the name of the service set can be populated dynamically for each subscriber at instantiation time. This dynamic profile encapsulates a service attachment point associated with a statically preprovisioned service set sset-1.

```
dynamic-profiles {
  profile prof-2 { # parameterized service attachment
    interfaces {
      $junos-interface-ifd-name {
        unit $junos-interface-unit {
          family inet {
            service {
              input {
                service-set $junos-service-set service-filter $junos-service-filter;
                post-input-filter $junos-post-input-filter ;
              }
              output {
                service-set $junos-service-set;
              }
            }
          }
        }
      }
    }
  }
}
```

To handle scalability more efficiently, in the following example the name of the service set can be populated dynamically for each subscriber at instantiation time.

```
dynamic-profiles {
  profile prof-2 { # parameterized service attachment
    interfaces {
      $junos-interface-ifd-name {
        unit $junos-interface-unit {
          family inet {
            service {
              input {
                service-set $junos-service-set service-filter $junos-service-filter;
                post-input-filter $junos-post-input-filter ;
              }
              output {
                service-set $junos-service-set;
```

```
                }
              }
            }
          }
        }
      }
    }
  }
```

The following attaches a service set dynamically at family inet6:

```
dynamic-profiles {
  profile prof-1 {
    interfaces {
      $junos-interface-ifd-name {
        unit $junos-interface-unit {
          family inet6 {
            service {
              input {
                service-set sset-1 service-filter fltr-1;
                post-input-filter pfltr-1 ;
              }
              output {
                service-set sset-1 service-filter fltr-1;
              }
            }
          }
        }
      }
    }
  }
}
```

Related Documentation

- Redirecting HTTP Requests Overview on page 3

Example: Configuring Destination Address Rewrite for HTTP Redirect

Requirements

- Multiservices DPC PIC

Overview

This procedure shows how to configure an DA rewrite rule. The destination port is not specified and the default behavior is determined by the rewrite configuration. If no rewrite configuration is available, the destination port is not rewritten.

Configuration

Example: Configuring a Rewrite Rule

Step-by-Step Procedure

1. Configure the service rule:

   ```
   [edit services captive-portal-content-delivery]
   user@host# set rule da-rewrite
   ```

2. Specify the term name:

   ```
   [edit services captive-portal-content-delivery da-rewrite]
   user@host# set term t1
   ```

3. Specify the match conditions for the term:

   ```
   [edit services captive-portal-content-delivery da-rewrite inet-filter term t1]
   user@host# set from applications junos-http
   ```

4. Specify the actions to take if the packet matches all the conditions in that term:

   ```
   [edit services captive-portal-content-delivery da-rewrite inet-filter term t1]
   user@host# set then rewrite desitnation-address 2001:2002::1;
   ```

Results

Confirm the configuration by entering the **show services** configuration command. If the command output does not display the intended configuration, repeat the instructions in this procedure to correct the configuration.

```
[edit services captive-portal-content-delivery]
rule da-rewrite {
   match-direction input-output
   term 1 {
      from {
         applications junos-http;
      }
      then {
         rewrite destination-address 2001:2002::1; # this is the remote redirect server.
      }
   }
}
```

The following example shows the configuration for an IPv6-DA rewrite service rule. Because the destination port is not specified, the default behavior is determined by the rewrite configuration. If no rewrite configuration is available, the destination port is not rewritten.

```
[edit services captive-portao-content-delivery]
rule ipv6da-rewrite {
   match-direction input-output
   term 1 {
      from {
         applications junos-http;
      }
      then {
         rewrite destination-address 2001:2002::1; # this is the remote
         # redirect server.
      }
```

```
    }
}
```

Verification

Displaying HTTP Redirect configuration

Purpose Verify the HTTP requests are redirected to the server.

Action user@host> **show services detail**

Related Documentation
- *Failover of the Control Service PICs*

Example: Configuring Redundant Multiservice

Requirements

- Multiservices DPC PIC

Overview

This procedure shows how to configure redundant multiservice support.

Configuration

Example: Configuring Redundant Multiservice for IPv4

Step-by-Step Procedure

1. Configure the interface:

   ```
   [edit interfaces]
   user@host# set interface rms0
   ```

2. Configure the redundant multiservice service set:

   ```
   [edit services]
   user@host# set service-interface rms0
   ```

3. Configure the redundant multiservice service set attachment:

   ```
   [edit interfaces]
   user@host# set ge-1/0/0 unit 100
   ```

Results Confirm the configuration by entering the **show redundancy-options** configuration command.

```
show redundancy-options
  redundancy-options {
```

```
    primary ms-2/1/0;
    secondary ms-3/1/0;
    hot-standby;
  }
    unit 0 {
      family inet;
    }
```

Confirm the service set configuration by entering the **show captive-portal-content-delivery-profile** configuration command.

```
show captive-portal-content-delivery-profile httpRedirect
  interface-service {
    service-interface rms0;
  }
```

Confirm the service set attachment by entering the **show show vlan-id** configuration command.

```
show vlan-id 100
  family inet {
    service {
      input {
        service-set sset10 service-filter walled;
      }
      output {
        service-set sset10;
      }
    }
    address 192.1.4.1/24;
  }
```

Verification

Displaying Redundant Multiservice Configuration

Purpose Verify the redundant multiservice configuration.

Action user@host> **show interfaces redundancy detail**

Related Documentation

- *Failover of the Control Service PICs*

CHAPTER 4

Configuration Statements

[edit services captive-portal-content-delivery] Hierarchy Level

```
services {
  captive-portal-content-delivery {
    rule rule-name {
      match-direction (input | output | input-output);
      term term-name {
        from {
          application [junos-http, junos-https, junos-httpproxy];
          destination-address address <except>;
          destination-prefix-list list-name <except>;
        }
        then {
          accept;
          redirect <url>;
          rewrite <destination-address address> <destination-port port-number>;
          syslog;
        }
      }
    }
    rule-set rule-set-name {
```

```
            [rule rule-name];
        }
    }
}
```

Related Documentation

- *Notational Conventions Used in Junos OS Configuration Hierarchies*
- *[edit services] Hierarchy Level*

application (Captive Portal Content Delivery)

Syntax application *application-name*;

Hierarchy Level [edit services captive-portal-content-delivery rule *rule-name* term *term-name* from (Captive Portal Content Delivery)]

Release Information Statement introduced in Junos OS Release 10.4.

Description Identify the application for inclusion in a rule.

Options ***application-name***—Application for rule to match, **junos-http**, **junos-https**, or **junos-httpproxy**.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Redirecting HTTP Requests Overview on page 3

captive-portal-content-delivery (Captive Portal Content Delivery)

Syntax

```
captive-portal-content-delivery {
  rule rule-name {
    match-direction (input | output | input-output);
    term term-name {
      from {
        application [junos-http, junos-https, junos-httpproxy];
        destination-address address <except>;
        destination-prefix-list list-name <except>;
      }
      then {
        accept;
        redirect <url>;
        rewrite <destination-address address> <destination-port port-number>;
        syslog;
      }
    }
  }
  rule-set rule-set-name {
    [rule rule-name];
  }
}
```

Hierarchy Level [edit services]

Release Information Statement introduced in Junos OS Release 10.4.

Description Configure the HTTP redirect service by specifying the location to which a subscriber's initial Web browser session is redirected, enabling initial provisioning and service selection for the subscriber.

The remaining statements are explained separately.

Required Privilege Level services—To view this statement in the configuration.
services–control—To add this statement to the configuration.

Related Documentation

- Redirecting HTTP Requests Overview on page 3

destination-address (Captive Portal Content Delivery)

Syntax destination-address *address* <except>;

Hierarchy Level [edit services captive-portal-content-delivery rule *rule-name* term *term-name* from (Captive Portal Content Delivery)]

Release Information Statement introduced in Junos OS Release 10.4.

Description Specify the destination address for rule matching.

Options ***address***—Destination IPv4 or IPv6 address or prefix value.

except—(Optional) Exclude the specified prefix list from rule matching.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Redirecting HTTP Requests Overview on page 3

destination-prefix-list (Captive Portal Content Delivery)

Syntax destination-prefix-list *list-name* <except>;

Hierarchy Level [edit services captive-portal-content-delivery rule *rule-name* term *term-name* from]

Release Information Statement introduced in Junos OS Release 10.4.

Description Specify the destination prefix list for rule matching. You configure the prefix list by including the **prefix-list** statement at the **[edit policy-options]** hierarchy level.

Options ***list-name***—Destination prefix list.

except—(Optional) Exclude the specified prefix list from rule matching.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Redirecting HTTP Requests Overview on page 3
- *Understanding Prefix Lists for Use in Routing Policy Match Conditions*

from (Captive Portal Content Delivery)

Syntax

```
from {
  application [junos-http, junos-https, junos-httpproxy];
  destination-address address <except>;
  destination-prefix-list list-name <except>;
}
```

Hierarchy Level [edit services captive-portal-content-delivery rule *rule-name* term *term-name*]

Release Information Statement introduced in Junos OS Release 10.4.

Description Specify input conditions for a captive portal term.

Options The remaining statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Redirecting HTTP Requests Overview on page 3
- *Firewall Filter Match Conditions Based on Address Fields*

match-direction (Captive Portal Content Delivery)

Syntax match-direction (input | output | input-output);

Hierarchy Level [edit services captive-portal-content-delivery rule (Captive Portal Content Delivery) *rule-name*]

Release Information Statement introduced in Junos OS Release 10.4.

Description Specify the direction in which the rule match is applied.

Options

input—Apply the rule match on the input side of the interface.

output—Apply the rule match on the output side of the interface.

input-output—Apply the rule match bidirectionally.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Redirecting HTTP Requests Overview on page 3

rule (Captive Portal Content Delivery)

Syntax

```
rule rule-name {
  match-direction (input | output | input-output);
  term term-name {
    from {
      application [junos-http, junos-https, junos-httpproxy];
      destination-address address <except>;
      destination-prefix-list list-name <except>;
    }
    then {
      accept;
      redirect <url>;
      rewrite <destination-address address> <destination-port port-number>;
      syslog;
    }
  }
}
```

Hierarchy Level [edit services captive-portal-content-delivery (Captive Portal Content Delivery)]

Release Information Statement introduced in Junos OS Release 10.4.

Description Specify the rule the router uses when applying this service.

Options *rule-name*—Identifier for the collection of terms that constitute this rule.

The remaining statements are explained separately.

Required Privilege Level services—To view this statement in the configuration.
services–control—To add this statement to the configuration.

Related Documentation

- Redirecting HTTP Requests Overview on page 3

rule-set (Captive Portal Content Delivery)

Syntax

```
rule-set rule-set-name {
    [rule rule-name];
}
```

Hierarchy Level [edit services captive-portal-content-delivery (Captive Portal Content Delivery)]

Release Information Statement introduced in Junos OS Release 10.4.

Description Define a set of captive portal content delivery rules that the router uses when applying this service.

Options

rule-set-name—Identifier for the collection of rules that constitute this rule set.

rule *rule-name*—Name of a rule defined at the **[edit services captive-portal-content-delivery]** hierarchy level.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Redirecting HTTP Requests Overview on page 3

services (Captive Portal Content Delivery)

Syntax

```
services {
  ...
  captive-portal-content-delivery {
    rule rule-name {
      match-direction (input | output | input-output);
      term term-name {
        from {
          application [junos-http, junos-https, junos-httpproxy];
          destination-address address <except>;
          destination-prefix-list list-name <except>;
        }
        then {
          accept;
          redirect <url>;
          rewrite <destination-address address> <destination-port port-number>;
          syslog;
        }
      }
    }
    rule-set rule-set-name {
      [rule rule-name];
    }
  }
  ...
}
```

Hierarchy Level [edit]

Release Information Statement introduced in Junos OS Release 10.4.

Description Define the captive portal and content delivery set of the rules statements to be applied to traffic.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Redirecting HTTP Requests Overview on page 3

term (Captive Portal Content Delivery)

Syntax

```
term term-name{
  from {
    application [junos-http, junos-https, junos-httpproxy];
    destination-address address <except>;
    destination-prefix-list list-name <except>;
  }
  then {
    accept;
    redirect <url>;
    rewrite <destination-address address> <destination-port port-number>;
    syslog;
  }
}
```

Hierarchy Level [edit services captive-portal-content-delivery rule *rule-name*]

Release Information Statement introduced in Junos OS Release 10.4.

Description Define the term match and action properties for the captive portal content delivery rule.

Options ***term-name***—Identifier for the term.

The remaining statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Redirecting HTTP Requests Overview on page 3

then (Captive Portal Content Delivery)

Syntax

```
then {
  accept;
  redirect <url>;
  rewrite <destination-address address> <destination-port port-number>;
  syslog;
}
```

Hierarchy Level [edit services captive-portal-content-delivery rule *rule-name* term *term-name*]

Release Information Statement introduced in Junos OS Release 10.4.

Description Define the term actions and any optional action modifiers for the captive portal content delivery rule.

Options

action—Actions to accept, redirect, or rewrite packets and all subsequent packets in flows that match the rules.

- **accept**—Accept the packets and all subsequent packets in flows that match the rules.
- **redirect**—Redirect the packet and all subsequent packets in flows that match the rules. You can optionally configure the following action modifier:
  - ***url***—(Optional) URL destination for the redirected packet. The URL must begin with **http://** or **https://**.
- **rewrite**— Rewrite the packet and all subsequent packets in flows that match the rules. You can optionally configure one or both of the following action modifiers:
  - **destination-address *address***—(Optional) Destination address for the rewritten packet.
  - **destination-port *port-number***—(Optional) Destination port for the rewritten packet.
- **syslog**— Log information about the packet to a system log file.

***action-modifiers* (Optional)**—Additional actions to accept, redirect, or rewrite packets and all subsequent packets in flows that match the rules.

- *destination-address*—(Optional) Destination address of the rewrite packet.
- *destination-port* —(Optional) Destination address and destination port of the rewrite packet.
- *url*—(Optional) URL of the redirect packet.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Redirecting HTTP Requests Overview on page 3
- *Firewall Filter Match Conditions Based on Address Fields*

traceoptions (Captive Portal Content Delivery)

Syntax

```
traceoptions {
  file filename <files number> <match regular-expression> <size size> <world-readable |
    no-world-readable>;
  flag flag;
  no-remote-trace;
}
```

Hierarchy Level [edit services captive-portal-content-delivery (Captive Portal Content Delivery)]

Release Information Statement introduced in Junos OS Release 10.4.

Description Define tracing operations for captive-portal-content-delivery processes.

Options

file *filename*—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. All files are placed in the directory **/var/log**. Ensure that filenames are unique for each logical system or routing instance in which Mobile IP is configured.

NOTE: Global messages (common to all logical systems and routing instances) are always saved in /var/log/mipd. Messages that are specific to a logical system or routing instance are never saved in /var/log/mipd. If you do not configure a trace filename for a logical system or routing instance, then nothing is traced for that entity.

size *size*—(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). If you specify a maximum file size, you also must specify a maximum number of trace files with the **files** option.

Syntax: **xk** to specify KB, **xm** to specify MB, or **xg** to specify GB

Range: 10 KB through 1 GB

Default: 128 KB

files *number*—(Optional) Maximum number of trace files. When a trace file named **trace-file** reaches its maximum size, it is renamed **trace-file.0**, then **trace-file.1**, and so on, until the maximum number of trace files is reached. Then the oldest trace file is overwritten. If you specify a maximum number of files, you also must specify a maximum file size with the **size** option.

Range: 2 through 1000

Default: 3 files

flag *flag*—Tracing operation to perform. To specify more than one tracing operation, include multiple **flag** statements. You can include the following flags:

- **all**—Trace all operations.
- **clicommand**—Trace CLI command operations.

- **configuration**—Trace home agent state machine operations.
- **general**—Trace general operations.
- **gres**—Trace graceful routing switchover operations.
- **ipc**—Trace Inter-Process Communication (IPC) messages between the PIC and the Routing Engine.
- **rtsock**—Trace routing socket operations.
- **rules**—Trace rules operations.
- **ssets**—Trace service sets operations.
- **statistics**—Trace statistics operations.

Required Privilege Level

trace—To view this statement in the configuration.
trace-control—To add this statement to the configuration.

Related Documentation

- Redirecting HTTP Requests Overview on page 3

PART 3

Administration

CHAPTER 5

Verifying and Managing Configurations

Verifying HTTP Redirect Requests

Purpose View information and statistics for the HTTP redirect configuration.

Action

- To display services statistics:

  user@host> **show services captive-portal-content-delivery statistics**

- To display services flows:

  user@host> **show services captive-portal-content-delivery flows**

- To clear services statistics:

  user@host> **clear services captive-portal-content-delivery statistics**

Related Documentation

- *Configuring HTTP Redirect Services*

CHAPTER 6

Monitoring Commands

- clear services captive-portal-content-delivery statistics
- show services captive-portal-content-delivery

clear services captive-portal-content-delivery statistics

Syntax **clear services captive-portal-content-delivery statistics** <interface *pic-name*>

Release Information Command introduced in Junos OS Release 10.4.

Description Clear captive portal content delivery statistics.

Options **interface**—Clear statistics by PIC name.

Required Privilege Level clear

Related Documentation
- show services captive-portal-content-delivery on page 43

Output Fields When you enter this command, you receive feedback on the status of your request.

clear services captive-portal-content-delivery statistics

```
user@host> clear services captive-portal-content-delivery statistics interface ms-5/0/0

user@host> show services captive-portal-content-delivery statistics interface ms-5/0/0

service-set interface: ms-5/0/0

Packets received   Packets altered
0                     0

Note that the stats are cleared.
```

show services captive-portal-content-delivery

Syntax

show services captive-portal-content-delivery
<pic *pic-name*>
<profile *profile-name*>
<rule *rule-name*> <term *term-name*>
<ruleset *ruleset-name*>
<sset *sset-name*> <brief> <detail> <summary>
<statistics> <interface *pic-name*>

Release Information

Command introduced in Junos OS Release 10.4.

Description

Display the current operational state of all captive portal interfaces.

Options

brief—(Optional) Display brief service set database information.

detail—(Optional) Display detailed service set database information.

pic—Display the PIC database.

profile—Display the profile database.

rule—Display the rule database.

ruleset—Display the rule set database.

sset—Display the service set database.

statistics—Display captive portal and content delivery statistics about a PIC.

summary—(Optional) Display a summary of service set database information.

term—(Optional) Display term information for the rule database.

Required Privilege Level

view

Related Documentation

- clear services captive-portal-content-delivery statistics on page 42

List of Sample Output

show services captive-portal-content-delivery on page 43

Sample Output

show services captive-portal-content-delivery

```
user@host> show services captive-portal-content-delivery pic ms-5/0/0
Name          Index
ms-5/0/0         20

user@host> show services captive-portal-content-delivery profile
Profile         Rules or Rule Sets
http-redirect   1
ipda-rewrite    1
```

```
user@host> show services captive-portal-content-delivery http-redirect
Profile         Rules or Rule Sets
http-redirect   1

user@host> show services captive-portal-content-delivery rule
Rule Name          Term Name
redirect           t2
rewrite            t1

user@host> show services captive-portal-content-delivery profile ipda-rewrite
Profile         Rules or Rule Sets
ipda-rewrite    1

user@host> show services captive-portal-content-delivery rule redirect
Rule Name          Term Name
redirect           t2

user@host> show services captive-portal-content-delivery rule rewrite
Rule Name          Term Name
rewrite            t1

user@host> show services captive-portal-content-delivery rule rewrite term t1
Rule name: rewrite
Rule match direction: input-output
Term name: t1
Term action: rewrite
Term action option: null

user@host> show services captive-portal-content-delivery rule redirect term t2
Rule name: redirect
Rule match direction: input
Term name: t2
Term action: redirect
Term action option: http://www.google.net

user@host> show services captive-portal-content-delivery sset sset1 detail
Service Set        Id       Profile         Compiled Rules
sset1              1        ipda-rewrite    1

user@host> show services captive-portal-content-delivery statistics interface ms-5/0/0
service-set interface: ms-5/0/0

Packets received   Packets altered
5                     3
```

PART 4

Troubleshooting

CHAPTER 7

Acquiring Troubleshooting Information

Collecting Subscriber Access Logs Before Contacting Juniper Technical Support

Problem When you experience a subscriber access problem in your network, we recommend that you collect certain logs before you contact Juniper Technical Support. This topic shows you the most useful logs for a variety of network implementations. In addition to the relevant log information, you must also collect standard troubleshooting information and send it to Juniper Technical Support in your request for assistance.

Solution To collect standard troubleshooting information:

- Redirect the command output to a file.

  user@host> **request support information | save rsi-1**

To configure logging to assist Juniper Technical Support:

1. Review the following blocks of statements to determine which apply to your configuration.

   ```
   [edit]
   set system syslog archive size 100m files 25
   set system auto-configuration traceoptions file filename
   set system auto-configuration traceoptions file filename size 100m files 25
   set protocols ppp-service traceoptions file filename size 100m files 25
   set protocols ppp-service traceoptions level all
   set protocols ppp-service traceoptions flag all
   set protocols ppp traceoptions file filename size 100m files 25
   set protocols ppp traceoptions level all
   set protocols ppp traceoptions flag all
   set protocols ppp monitor-session all
   set interfaces pp0 traceoptions flag all
   set demux traceoptions file filename size 100m files 25
   set demux traceoptions level all
   set demux traceoptions flag all
   set system processes dhcp-service traceoptions file filename
   set system processes dhcp-service traceoptions file size 100m
   set system processes dhcp-service traceoptions file files 25
   set system processes dhcp-service traceoptions flag all
   set class-of-service traceoptions file filename
   set class-of-service traceoptions file size 100m
   set class-of-service traceoptions flag all
   set class-of-service traceoptions file files 25
   set routing-options traceoptions file filename
   set routing-options traceoptions file size 100m
   set routing-options traceoptions flag all
   set routing-options traceoptions file files 25
   set interfaces traceoptions file filename
   set interfaces traceoptions file size 100m
   set interfaces traceoptions flag all
   set interfaces traceoptions file files 25
   set system processes general-authentication-service traceoptions file filename
   set system processes general-authentication-service traceoptions file size 100m
   set system processes general-authentication-service traceoptions flag all
   set system processes general-authentication-service traceoptions file files 25
   ```

2. Copy the relevant statements into a text file and modify the log filenames as you want.
3. Copy the statements from the text file and paste them into the CLI on your router to configure logging.
4. Commit the logging configuration to begin collecting information.

NOTE: The maximum file size for DHCP local server and DHCP relay log files is 1 GB. The maximum number of log files for DHCP local server and DHCP relay is 1000.

BEST PRACTICE: Enable these logs only to collect information when troubleshooting specific problems. Enabling these logs during normal operations can result in reduced system performance.

Related Documentation

- *Compressing Troubleshooting Logs from /var/logs to Send to Juniper Technical Support*

PART 5

Index

Index

T

W