Introduction to ServerIron ADX Application Switching and Load Balancing

Module 6: Content Switching (CSW)

Revision 0310
Objectives

- Upon completion of this module the student will be able to:
  - Define layer 7 switching / Content Switching (CSW)
  - Describe Cookie Switching
  - Describe Cookie Hashing
  - Define the difference between Cookie Hashing and Cookie Switching
  - Describe URL Switching
  - Implement Cookie Switching and URL Switching using CLI
Session Persistence

- The ability to persist all the sessions for a given user to the same server for the duration of an application transaction.
  - Identify the user
  - Recognize when an application transaction begins or ends

- Types of Session Persistence:
  - Source IP, Virtual IP, Port
  - Port Tracking
  - Concurrent
  - Sticky
  - Cookie Based Persistence
    - Switching
    - Hashing
What is L7 Content Switching?

- Load balancing based on rules and actions defined by users
- Load balancing based on any specified HTTP header
- Load balancing based on XML content
- Load-balancing decisions based on multiple HTTP headers or XML tags
- Redirecting requests to alternate URLs or domains, persisting requests to servers, and simple forwarding actions.
- Content rewrite, insertion and deletion functions
Common Layer 7 CSW Features

- Cookie Switching / Insertion
- Client-IP Header Insertion
- URL Switching
- URL Rewrite
- URL Redirect (HTTP Redirect)
- HTTP to HTTPS Rewrite
- Insert Custom Response Header
Layer 7 Switching Methods

- Cookie Switching
  - Uses cookie to direct to specific server or server group
- Cookie Hashing
  - Hashes on a cookie to direct client's request to a specific server
- URL Switching
  - Uses a user specified piece of the URL to direct to specific server or server group
- URL Hashing
  - Hashes the URL string to direct to a specific server or server group
Cookie Switching Overview

- Provides a standards based way for the server to communicate with the load balancer
- Cookie name is user-configurable
- Ensures persistence to the same server
Cookie Switching Configuration

Step 1: Configure the Server to set a cookie:  
```
SetCookie("ServerID","1024")
```

Step 2: Configure the real server:
```
ServerIron ADX (config)# server real-name rs100 192.168.1.100
ServerIron ADX (config-rs-rs100)# port http server-id 1024
ServerIron ADX (config-rs-rs100)# exit
```
Cookie Hashing Overview

- Ensures that a given set of cookie names and value will always be sent to the same server
1. ServerIron ADX examines the Cookie header in an HTTP request.
2. ServerIron ADX logically reduces the Cookie header to a number between 0-255.
3. The number corresponds to one of 256 internal “hashing buckets” on the ServerIron ADX.
4. Using its load balancing metric, the ServerIron ADX allocates a real server to the hashing bucket.
5. The ServerIron ADX sends the HTTP request to the real server allocated to the cookie’s hashing bucket.
URL Switching Overview

- Specify URL rules based on prefix, suffix or a pattern
- Up to 256 URL rules
- Benefits of URL switching without losing source IP information
- Support for URL hashing: Select a server by hashing the whole URL or a specific segment
Layer 7: Using a Cookie

• The cookie can be used to distinguish individuals

• The cookie can maintain session persistence between client and server

• Direct HTTP request to server or server groups based on information in the cookie
Cookie Switching

• Value of cookie defines real server’s ID
  – Cookie: ServerID=2 (go to server 2)

• Persistence cannot be guaranteed
  – One cookie can have multiple IDs

Cookie:ServerID=2; ServerID=1; address=San Jose;
Cookie Hashing

- The calculation of the checksum or hash key can be based on one of the following strings:
  - Value of certain cookie – the check sum can be based on the value of “ServerID” which is 1;
  - Value of the whole cookie header – the checksum of :ServerID=1; comment= “This is a long string. Checksum based on the whole string will be time consuming.; will be calculated.
Cookie Hashing (Cont.)

- Hash on cookie
- Assigns hash value to a server
- Resolution to same hash value get same server
- Hash value can be based on:
  - Value of portion of cookie
  - Value of compound (entire) cookie
Cookie Insertion

1. ServerIron ADX inserts cookie in response from server
2. Next client request is directed to server or server group based on information in cookie
3. ServerIron ADX can also delete a cookie
Cookie Insertion (Cont.)

• The ServerIron ADX will insert a cookie when:
  – There is no cookie header
  – The cookie header exists but it does not contain the cookie name specified by the port http cookie-name command.
  – The cookie name is found, but the cookie value is out of range. The cookie value must be between 1 – 2047.
  – The cookie name is found, but the real server or server group indicated by the cookie value is not available.
Layer 7 Content Switching (CSW) Introduction
Layer 7 CSW : 3-Step Configuration

- Content Switching requires a 3 step configuration process in order to define the content switching rules and policies.
  1. Define a CSW Rule

  2. Create a policy
     - Policies “match rules” and take action

  3. Bind and enable policy to a Virtual Server
Step 1: Define a CSW Rule

- Specifies content to match in HTTP traffic
  1. Header
     Example: (config)# csw-rule rule4 header host exists
     (also: header prefix/suffix/pattern/equals/search)
  2. URL
     Example: (config)# csw-rule rule3 url exists
     (also: url prefix/suffix/pattern/equals/search)
  3. Method
     Example: (config)# csw-rule rule1 method eq PUT
     (also: GET, HEAD, POST, PUT, DELETE, TRACE, OPTIONS, or CONNECT)
  4. Version
     Example: (config)# csw-rule rule2 version eq 1.1
Step 2: Create a policy

• Specifies action to take when rule is matched

1. Create a policy

   (config)# csw-policy p1
   (config-csw-p1)#

2. Match rule/take action in one statement

   a) Forward

      (config-csw-p1)# match rule1 forward 1029

   b) Redirect

      (config-csw-p1)# match rule1 redirect "*" "*" ssl

   c) Rewrite

      (config-csw-p1)# match rule1 rewrite request-insert client-ip
Step 3: Bind policy and Enable CSW

- Bind policy & turn on csw to a particular VIP

```
(config)# server virtual cswVIP 192.168.10.100

(config-vs-cswVIP)# port http

(config-vs-cswVIP)# port http csw-policy p1

(config-vs-cswVIP)# port http csw

(config-vs-cswVIP)# bind http rs1 http

(*) must bind at least 1 real server to http and it must be Active
```
Examples: CSW Rules and Policies

Global Policy
• Example: create a policy called Policy1
  - SLB(config)# csw-policy "Policy1"

Rules
• url pattern- matches a string in the url header
• header - matches a string in the header

• Example: Redirect the client to SSL, default goes SSL.
  - SLB(config-csw-Policy1)# default redirect "*
  ssl

  first "**" is match all domains; second "**" is match all urls
CSW Primary and Secondary Commands

Primary Commands

- **Persist** - sends requests with similar content to the same server
- **Reset-client** - sends a reset to the client to terminate the connection
- **Reply-error** - replies a 403 error back to the client
- **Redirect** - redirects client traffic
- **Forward** - forwards traffic to a specified server or server group

Example: default is to forward traffic to server group 10

```
  - SLB(config-csw-Policy1)# default forward 10
```

Secondary Commands (*)

- **Log** - logs to external log server when a rule is matched
- **Rewrite** - modifies the HTTP header, insert or deletes content

Example: modifies HTTP header, inserting client IP address

```
  - SLB(config-csw-p1)# default rewrite request-insert client-ip
```

(*) A primary command must exist, before a secondary can be used
Cookie Session Persistence

- The server can have cookies inserted based on the server being in:
  - A server group
    - Servers are in a group and session persistence is done on the servers in the group
  - An individual server
    - There servers are stand alone and each server has it’s own server ID.
Configure Server Group ID

ServerIron ADX (config)# server real-name rs1 10.10.10.201
ServerIron ADX (config-rs-rs1)# port http group-id 10 10
ServerIron ADX (config-rs-rs1)# exit

ServerIron ADX (config)# server real-name rs2 10.10.10.202
ServerIron ADX (config-rs-rs2)# port http group-id 10 10
ServerIron ADX (config-rs-rs2)# exit

• Syntax: [no] port http group-id <server-group-id-range>
Configure Server ID

ServerIron ADX (config)# server real-name rs1 10.10.10.201
ServerIron ADX (config-rs-rs100)# port http server-id 1024
ServerIron ADX (config-rs-rs100)# exit

ServerIron ADX (config)# server real-name rs2 10.10.10.202
ServerIron ADX (config-rs-rs100)# port http server-id 1025
ServerIron ADX (config-rs-rs100)# exit

• Syntax: [no] port http server-id <server-id>
Enable Cookie Switching Policy

(config)# server virtual cookieVIP 206.65.10.20
(config-vs-cookieVIP)# port http
(config-vs-cookieVIP)# port http cookie-name “ServerID”
(config-vs-cookieVIP)# port http csw-policy “myCookie”
(config-vs-cookieVIP)# port http csw
(config-vs-cookieVIP)# bind http rs1 http rs2 http
Define CSW Rule for Cookie Switching

- Specifies content to match in HTTP traffic

- Example of Header Rule:

  (config)# csw-rule r1 header "cookie" search "ServerID"

  (also: header prefix/suffix/pattern/equals/search)
Create a Policy for Cookie Switching

• Specifies action to take when rule is matched

1. Create a policy
   
   (config)# csw-policy myCookie
   
   (config-csw-myCookie)#

2. Match rule & take action in one statement
   
   a) Persist
      
      (config-csw-myCookie)# match r1 persist offset 0 length 4 group-or-server-id

   b) Forward
      
      (config-csw-myCookie)# default forward 10

   c) Rewrite
      
      (config-csw-myCookie)# default rewrite insert-cookie
Cookie Insertion

Lab 6-1
Lab 6-1: Cookie Insertion

Remote Client: 144.100.10.2/24
GW: 144.100.10.1

MGMT: 169.144.10.11
GW: 169.144.10.1
cookie VIP: 169.144.10.100
Server
Source IP: 10.10.10.50

RS1 10.10.10.201/24
RS2 10.10.10.202/24
RS1 & RS2 GW 10.10.10.50
Cookie Domains

• A cookie domain identifies a server that sent a cookie
• The browser will only send the cookie to 
  http://server.domain1.com
  or to
  a.us.oracle.com
what ever the domain is, that is the server the cookie will be sent to.
Cookie Domains (Cont.)

- A cookie domain identifies a server that sent a cookie
  
  - ServerIron ADX (config)# server virtual cookieVIP 192.168.1.241

  - ServerIron ADX (config-vs-cookieVIP)# port http cookie-domain "brocade.com"

  - ServerIron ADX (config-vs-cookieVIP)# exit

- Syntax: [no] port <virtual port> cookie-domain <domain>
Cookie Path

• Sets the URL that the cookie is valid for.

  - ServerIron ADX (config)# server virtual cookieVIP 192.168.1.241

  - ServerIron ADX (config-vs-cookieVIP)# port http cookie-path "/services/documentation/"

  - ServerIron ADX (config-vs-cookieVIP)# exit

• Syntax: [no] port <virtual port> cookie-path <path>
Cookie Age

• ServerIron ADX (config)# server virtual cookieVIP 192.168.1.241

• ServerIron ADX (config-vs-cookieVIP)# port http cookie-age 10

• ServerIron ADX (config-vs-cookieVIP)# exit

• Syntax: [no] port <virtual port> cookie-age <minutes>
Display Cookie Information

ServerIron ADX # **show cookie-info**

<table>
<thead>
<tr>
<th>Cookie:</th>
<th></th>
<th>Total Insertion Error: 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Inserted:</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Total Deleted:</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>Total Destroyed:</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Content Rewrites:</th>
<th></th>
<th>Total Frees: 34</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Allocated:</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>Used Now:</td>
<td>0</td>
<td>Allocation Failures: 0</td>
</tr>
</tbody>
</table>

Total Memory Already Consumed: 1600 KB.
Total Memory Can Be Reached: 25600 KB.
HTTP/1.1 Packet Analysis

GET /images/homeNav/HomeNav1.gif HTTP/1.1
Accept: */*
Referer: http://206.65.10.10
Accept-Encoding: gzip, deflate
If-Modified-Since: Wed, 27 Oct 1999 20:10:54 GMT; length=1054
Host: 206.65.10.10
Connection: Keep-Alive
Cookie: Training_Cookie_001=0

HTTP/1.0 304 Not Modified
Set-Cookie: Training_Cookie_001=0 U; path=/
Date: Thu, 30 Oct 2003 19:12:58 GMT
Connection: close
ETag: "40818-41e-38175c4e"

HTTP/1.0 304 Not Modified
Set-Cookie: Training_Cookie_001=1111; path=/
Date: Thu, 30 Oct 2003 20:26:17 GMT
Server: Apache/1.3.6 (Unix) mod_perl/1.21 mod_ssl/2.2.8 OpenSSL/0.9.2b
Connection: close
ETag: "9887-3696-3c64d382"
URL Switching for Personal Websites

- Inspecting the URL to serve multiple web sites
  - General pattern matching on string

```
www.isp.com/user/adam
www.isp.com/user/pat
www.isp.com/user/roger
```
URL Switching Static and Dynamic Content

- Separate static content requests from dynamic content requests
  - Pattern matching on suffix and string

![Diagram of URL switching static and dynamic content](URLSwitching_static-dynamic.png)
Define CSW Rule for URL Switching

Specifies string to match in URL

- Example of URL Prefix Rule

```
(config)#csw-rule "products" url prefix "/PRODUCTS" case-insensitive

(config)#csw-rule "support" url prefix "/SUPPORT" case-insensitive
```
Create a Policy for URL Switching

• Specifies action to take when rule is matched

1. Create a policy

   (config)# csw-policy “myUrlPolicy” case-insensitive
   (config-csw-myUrlPolicy)#

2. Match rule & take action in one statement

   a) Forward
   (config-csw-myUrlPolicy)# match “products” forward 201

   b) Forward
   config-csw-myUrlPolicy)# match “support” forward 202

   c) Forward
   (config-csw-myUrlPolicy)# default forward 201
Enable URL Switching Policy

Configuration Steps:

ADX (config)# server virtual url-sw-VIP 206.65.10.20
ADX (config-vs-url-sw-VIP)# port http
ADX (config-vs-url-sw-VIP)# port http csw-policy "myUrlPolicy"
ADX (config-vs-url-sw-VIP)# port http csw
ADX (config-vs-url-sw-VIP)# bind http rs1 http rs2 http
URL Switching

Lab 6-2
Lab 6-2: URL Switching

Remote Client: 144.100.10.2/24
GW: 144.100.10.1

MGMT: 169.144.10.11
GW: 169.144.10.1
myUrlVIP: 169.144.10.100
Server
Source IP: 10.10.10.50

RS1 10.10.10.201/24
RS2 10.10.10.202/24
RS1 & RS2 GW 10.10.10.50
End of Module 6: Content Switching (CSW)