



Alteon
34.x

Alteon Level 1 Lab Manual Introduction to AppShape++



Table of Contents

Overview	3
Objectives	3
Lab Preparations: Restore Standard Setup.....	3
Lab Activities	4
“No Service” AppShape++ Script	4
Insert X-Forwarded-For Header Using AppShape++	6
URL Filtering Using AppShape++	7
Perform Persistency On X-Forwarded-For Header Or ClientIP	9

Overview

In various cases there is a need to extend the Alteon functionality beyond built in features. In such case you can use AppShape++ scripting language to achieve your goals.

Objectives

After viewing a training module on AppShape++ scripting and completing this lab, you should be able to:

- Use AppShape++ script to return “no service” page
- Use AppShape++ script to insert a HTTP header (X-Forwarded-For)
- Use AppShape++ script to block traffic based on the request host name
- Use AppShape++ script to perform persistency on a HTTP Header (X-Forwarded-For) or Client-IP

Lab Preparations: Restore Standard Setup

Before you begin this lab:

- You should have successfully completed ADC SLB setup.
- Access Alteon management port and login.
 - Import SLB SETUP if changes were made after the previous lab.
 - Set the group metric to Round Robin


```
/c/slb/group <group>/metric roundrobin
```
 - Disable persistent binding (pbind). PBind takes precedence over string load balancing.


```
/c/slb/virt <virt server>/service <virt service>/pbind disable
```
- Verify your SLB SETUP is properly working before going on.
 - Clear the session table.


```
/oper/slb/clear
```
 - Generate test traffic to your servers (VIP)
 - View statistics on your virtual server connection


```
/st/slb/virt <virt>
```

```
/st/slb/clear
```

Lab Activities

Here is a summary of tasks in this lab:

1. Use AppShape++ to return a “no service” page.
2. Use AppShape++ to remove an HTTP header.

“No Service” AppShape++ Script

1. Include an AppShape++ script to display a sorry page (no service page) in case all servers go down.

AppShape++ Script: Sorry Page

```
when INIT {
set static::STATUS_CODE "200"
set static::CONTENT "<html><head><title>Sorry
Page</title></head><body><img
src=\"https://www.radware.com/RadwareNext/images/logo.svg
\"><br><h1>Please try again later, service will be back
soon!</h1></body></html>"
}

when HTTP_REQUEST {
if {[group count active_servers [LB::server group]] == 0 } {
    HTTP::respond $static::STATUS_CODE content $static::CONTENT
}
}

-----END
```



GUI:

- Configuration -> Application Delivery -> AppShape++ Scripts
- Click Import icon [to create script]
- Click Enable [to enable script]
- Enter Script ID no_service_page
- Browse [to select file] if you saved the above script as file, or use Text if you copy&paste it
- Click Import
- Apply
- Save

IMPORTANT: If it is necessary to type the AppShape++ script into a plain-text editor to create it -- USE A PLAIN-TEXT EDITOR. Radware WARNS against using word processors (ie. copy/paste) because they use hidden formatting codes that, when included in your script, will cause errors.



You can use the first example (L4) AppShape++ script available in Radware's Knowledge Base (https://support.radware.com/app/answers/answer_view/a_id/15995)



For more help, see AppShape++ User Guide. AppShape++ eLearning is available.

- a. Add the AppShape++ script on the service and enable the feature “alwayson” / “Service Always Up” (CLI/GUI)



GUI:

- Configuration -> Application Delivery -> Virtual Services
- Edit Virt1
- Edit Virtual Service 80
- AppShape++ tab
- Add the new no_service_page script
- Script priority = 1
- Select the no_service_page script from the dropdown list
- Submit
- Set the Service Always Up = Enable
- Close
- Close
- Apply
- Save

- b. Test your “no_service_page” script:

- Disable both servers (WebServer1 and WebServer2), then open the VIP to see the sorry page display.



GUI:

- Monitoring -> Application Delivery -> Server Resources -> Real Servers
 - Highlight both real server and disable WebServer1 and WebServer2 by clicking the Disable button
- i. Re-enable both servers (WebServer1 and WebServer2) before continuing with the lab exercises.

Access the VIP on http and you should see the sorry page:



- Use the Monitoring -> Application Delivery -> Server Resources -> Real Servers to enable both server again

Insert X-Forwarded-For Header Using AppShape++

We add the X-Forwarded-For header with an AppShape++ script. This would allow for example to use a different header name like X-IP or something else. In the lab we use X-Forwarded-For, since this header will be visible at the server.

1. Make sure your current server group for port 80 is send to the web server on port 80 to see the “Welcome to the Server” page.
Go to <http://www.radware.lab> or <http://192.168.175.50>

2. You should see no X-Forwarded-For IP is displayed:

Your IP is	: 10.200.1.42
Host	: www.radware.lab

3. Create an AppShape++ script to add the X-Forwarded-For header.

```
when HTTP_REQUEST {
    HTTP::header insert X-Forwarded-For [IP::client_addr]
}
-----END
```



GUI:

- Configuration -> Application Delivery -> AppShape++ Scripts
- Click Import icon [to create script]
- Click Enable [to enable script]
- Enter Script ID [insertxff]
- Browse [to select file]
- Click Import

4. Attach the AppShape++ script to virtual server Virt1 service 80.



GUI:

- Configuration -> Application Delivery -> Virtual Services
- Edit Virt1
- Edit Virtual Service 80
- AppShape++ tab
- Add the new insertxff script
- Script priority = 2
- Select the insertxff script from the dropdown list
- Submit
- Close
- Apply
- Save

5. Verify your configuration by browsing to your VIP from your RDP PC, you should see the client IP as X-Forwarded-For

```

Your visible IP is: 10.200.1.42
X-Forwarded-For IP: 192.168.175.20

Host : www.radware.lab

```

URL Filtering Using AppShape++

In this exercise we use an AppShape++ script to block requests to the VIP if they have specific host names. The training web server serve multiple applications: hackazon, bwapp, juiceshop and the training pages.

The script is blocking access to the VIP if the bwapp or juiceshop is requested.

We use a data class and the script check if the requested URL (Host header value) appears in it. If so, the Alteon responds with "Access Denied" message back to the client..

1. Make sure your current server group for port 80 is send to the web server on port 80 to see the "Welcome to the Server" page.

Go to <http://bwapp.radware.lab> you should see the bWapp application:

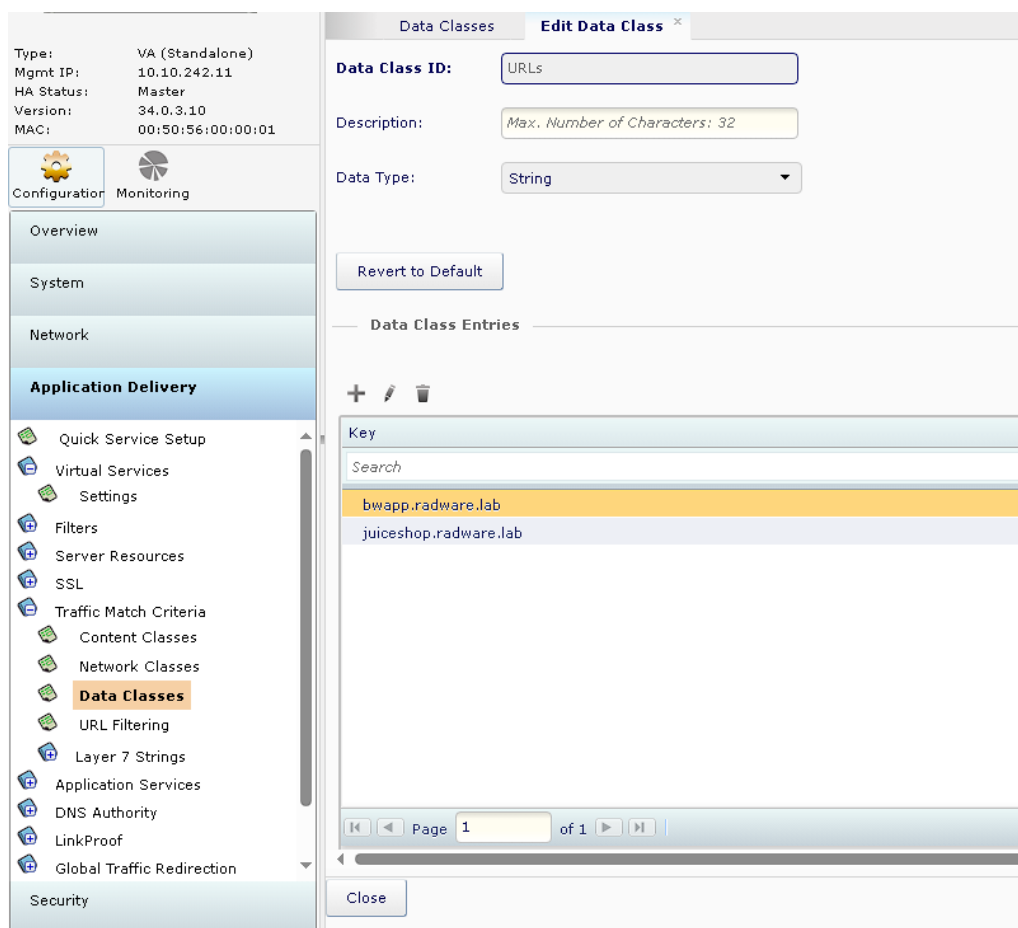


2. We use a data class to specify the host names we want to block:
On the CLI of the Alteon add this commands:

```

/c/slb/dataclass/class URLs string manual
data "bwapp.radware.lab"
data "juiceshop.radware.lab"

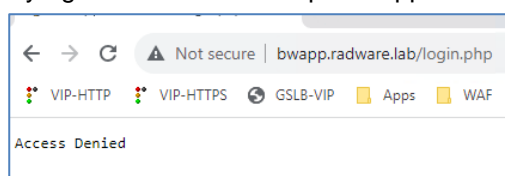
```

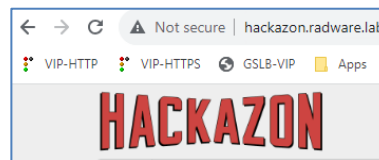
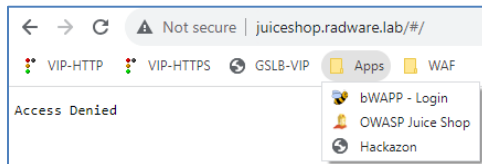
3. Create an AppShape++ script to add them to block the special urls and add a log entry to the

```
when HTTP_REQUEST {
  set host [HTTP::host]
  if {[class match $host equals URLs] == 1} {
    HTTP::respond 200 content "Access Denied"
    log -a "Time: [clock seconds], Source IP: [IP::client_addr], URL: $host, Action: blocked"
  }
}
-----END
```

4. Add the script to the virtual service as learned previously.
5. Apply the changes
6. Try again to connect to <http://bwapp.radware.lab> and you should see the "Access Denied" message:



- Try as well <http://juiceshop.radware.lab> you should see the deny message as well, but hackzon should work



Perform Persistency On X-Forwarded-For Header Or ClientIP

In this exercise we use an AppShape++ script to perform persistency on the X-Forwarded-For header value and if the header field is absent, we perform persistency on the Client-IP.

- Make sure your current server group for port 80 is send to the web server on port 80 to see the “Welcome to the Server” page.
Go to <http://www.radware.lab> or <http://192.168.175.50>
- You should see no X-Forwarded-For IP is displayed (**disable the “insertxff” script ID from the Virtual services if you activated it previously**):

```
Your IP is      : 10.200.1.42
Host           : www.radware.lab
```

- Create an AppShape++ script to use the X-Forwarded-For header or the client IP for persistency.

```
when HTTP_REQUEST {
    if {[HTTP::header exists X-Forwarded-For]} {
        persist usid [HTTP::header value X-Forwarded-For] 3600
    } else {
        persist usid [IP::client_addr] 3600
    }
}
-----END
```

- Add the script to the virtual service as learned previously.
- Apply the changes
- Connect to the ssh of the Alteon
- Use the following command to see that no persistent entry is in the dynamic datastore yet:

```
/info/slb/ddstore/persist/dump
```

```
Total number of printed session IDs: 0
```
- In case you see entries you can use the command below to clear the table

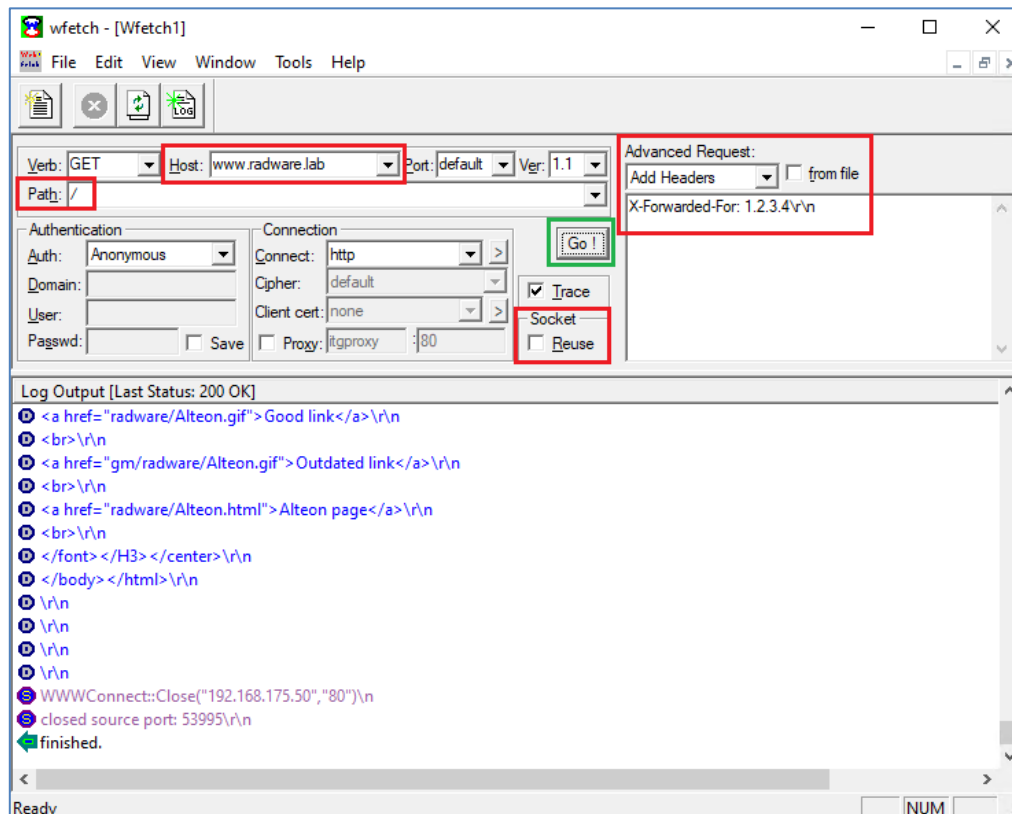
```
/oper/slb/ddstore/prstdel all
```
- We use the application called WFetch on your RDP client to test the script. The application is located at C:\Tools\WFetch
- In the Application set the following parameters
 - Host: www.radware.lab
 - Path: /
 - Socket: uncheck Reuse
- Click on “Go !” to trigger an http request.

12. In the dynamic datastore now the IP of the client should be added as persistency criteria

```
/info/slb/ddstore/persist/dump
192.168.175.20, d:192.168.175.50 80, g:Both_Server rs:Server1 80, age 3590 U
Total number of printed session IDs: 1
```

13. Now let's add the X-Forwarded-For header in the WFetch application

- Advanced Request: Select "Add Headers"
- Add into the box below: X-Forwarded-For: 1.2.3.4\r\n



14. Click on "Go !" to trigger an http request with XFF value.

15. In the dynamic datastore now the IP of the XFF header should be added as persistency criteria

```
/info/slb/ddstore/persist/dump
192.168.175.20, d:192.168.175.50 80, g:Both_Server rs:Server1 80, age 3590 U
1.2.3.4, d:192.168.175.50 80, g:Both_Server rs:Server2 80, age 3595 U
Total number of printed session IDs: 2
```

16. You can try use different values in the X-Forwarded-For header and you should see additional values tracked

```
/info/slb/ddstore/persist/dump
8.8.8.8, d:192.168.175.50 80, g:Both_Server rs:Server1 80, age 3584 U
192.168.175.20, d:192.168.175.50 80, g:Both_Server rs:Server1 80, age 3208 U
1.2.3.4, d:192.168.175.50 80, g:Both_Server rs:Server2 80, age 3397 U
6.7.8.9, d:192.168.175.50 80, g:Both_Server rs:Server2 80, age 3598 U
Total number of printed session IDs: 4
```

Export configuration as a backup. Name the file **BACKUP APPSHAPE++**.



© 2024 Radware Ltd. All rights reserved. The Radware products and solutions mentioned in this document are protected by trademarks, patents and pending patent applications of Radware in the U.S. and other countries. For more details, please see: <https://www.radware.com/LegalNotice/>. All other trademarks and names are property of their respective owners.

For questions, contact training@Radware.com