F5 Firewall Solutions Documentation

F5 Networks, Inc.

Feb 25, 2020

Agility 2020 Hands-on Lab Guide

Contents:

1 Class 1: AFM – The Data Center Firewall 5

2 Advanced Multi-Layer Firewall Protection 99

3 Class - F5 BIG-IP DDoS and DNS DoS Protections 165

4 Flowmon Integrated Out-of-path DDoS Solution 207

1

Class 1: AFM – The Data Center Firewall

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.

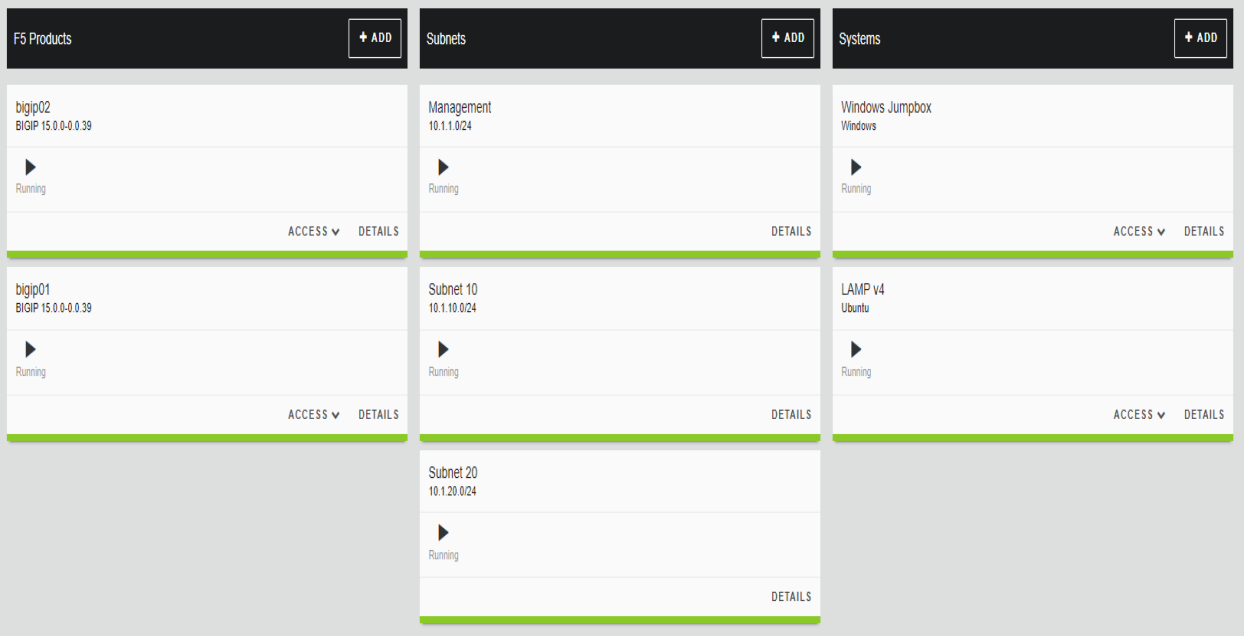
1.1.1 Lab Topology

The training lab is accessed over remote desktop connection.

Your administrator will provide login credentials and the URL.

Within each lab environment there are the following Virtual Machines:

- **Windows 7 Jumpbox**
- username: external_user password: P@ssw0rd!
- **Two BIG-IP Virtual Editions (VE) – running TMOS 15.1**
- username: admin password: f5DEMOs4u
- **LAMP Server (Web Servers)**



Lab Components

Below are all the IP addresses that will be used during the labs. Please refer back to this page and use the IP addresses assigned to your site.

	IP Addresses
Lampserver	10.1.20.11, 10.1.20.12, 10.1.20.13 ,10.1.20.14, 10.1.20.15

1.2 Lab 1 – Advanced Firewall Manager (AFM)

1.2.1 Lab Overview

During this lab, you will configure the BIG-IP system to permit traffic to multiple backend servers. You will then run simulated user flows against BIG-IP and verify the traffic flow, reporting and logging of these flows.

1.2.2 Base BIG-IP Configuration

In this lab, the VE has been configured with the basic system settings and the VLAN/self-IP configurations required for the BIG-IP to communicate and pass traffic on the network. Inspect the Virtual Servers which have been configured. Note that they are Wwildcard (listen for all traffic) and have SNAT auto-map enabled. We'll now need to configure the BIG-IP to pass it to the back-end server.

1.2.3 Advanced Firewall Manager

Welcome to Initech! Today is your first day as the principal firewall engineer, congratulations! The employee you are replacing, Milton, is rumored to be sitting on a beach in Key West sipping Mai Tai's and took his red stapler but left no documentation. . .

The marketing team, now led by Bill Lumbergh, launched a new campaign for Initech's TPS reports overnight and no one can access the web server. The only information the web server administrators know is that the IP address of the Web server is 10.30.0.50 and that Mr. Lumbergh is furious the world does not know about the glory of TPS reports!!

Let's start by testing the web server to verify. On your workstation open a browser (we prefer you use the Chrome shortcut labeled BIG-IP UI, all the tabs are pre-populated) and enter the address of the web server (http://10.1.20.11). No Bueno! Let's see if we can even ping the host. Launch a command prompt (startrun cmd) and type 'ping 10.1.20.11'. Bueno! Looks like the server is up and responding to pings, as such, this is likely not a network connectivity issue.

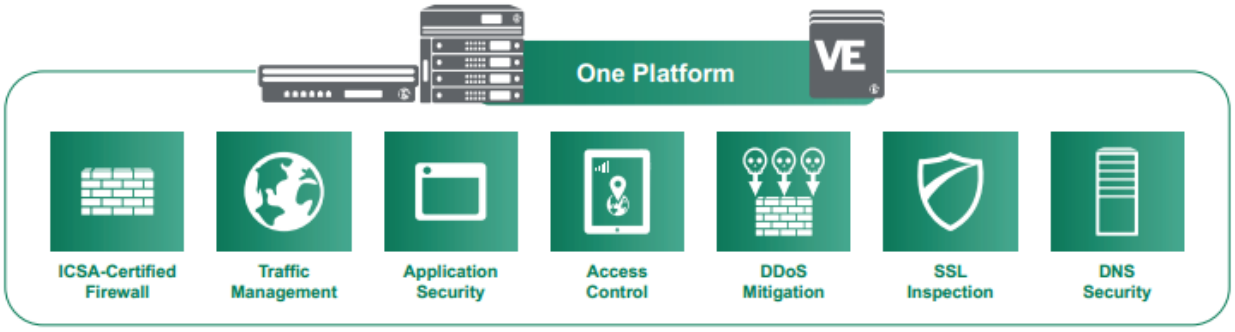
You ask one of your colleagues, who just got out of his meeting with the Bob's, if he knows the IP address of the firewall. He recalls the firewall they would traverse for this communication is bigip01.f5demo.com and its management IP address is 10.1.1.4. In your browser, open a new tab and navigate to https://10.1.1.4. The credentials to log into the device are username: admin and password: f5DEMOs4u (these can also be found on the login banner of the device for convenience). Note if you receive a security warning it is ok to proceed to the site and add as a trusted site.

F5? F5 makes a data center firewall? Maybe I should do a little reading about what the F5 firewall is before I proceed deeper into the lab. . .

1.2.4 Advanced Firewall Manager (AFM)

Advanced Firewall Manager (AFM) is a module that was added to TMOS in version 11.3. F5 BIG-IP Advanced Firewall Manager™ (AFM) is a high-performance ICSA certified, stateful, full-proxy network firewall designed to guard data centers against incoming threats that enter the network on the most widely deployed protocols—including HTTP/S, SMTP, DNS, SIP, and FTP.

By aligning firewall policies with the applications, they protect, BIG-IP AFM streamlines application deployment, security, and monitoring. With its scalability, security and simplicity, BIG-IP AFM forms the core of the F5 application delivery firewall solution.



Some facts below about AFM and its functionality:

- Advanced Firewall Manager (AFM) provides "Shallow" packet inspection while Application Security Manager (ASM) provides "Deep" packet inspection. By this we mean that AFM is concerned with source IP address and port, destination IP address and port, and protocol (this is also known as 5-tuple/quintuple filtering).
- AFM is used to allow/deny a connection before deep packet inspection ever takes place, think of it as the first line of firewall defense.
- AFM is many firewalls in one. You can apply L4 firewall rules to ALL addresses on the BIG-IP or you can specify BIG-IP configuration objects (route domains, virtual server, self-IP, and Management-IP).

- AFM runs in 2 modes: **ADC mode** and **Firewall** mode. **ADC mode** is called a “blacklist”, all traffic is allowed to BIG-IP except traffic that is explicitly DENIED (this is a negative security model). **Firewall mode** is called a “whitelist”, all traffic is denied to BIG-IP except traffic that is explicitly ALLOWED. The latter is typically used when the customer only wants to use us as a firewall or with LTM.
- We are enabling “SERVICE DEFENSE IN DEPTH” versus traditional “DEFENSE IN DEPTH”. This means, instead of using multiple shallow and deep packet inspection devices inline increasing infrastructure complexity and latency, we are offering these capabilities on a single platform.
- AFM is an ACL based firewall. In the old days, we used to firewall networks using simple packet filters. With a packet filter, if a packet doesn’t match the filter it is allowed (not good). With AFM, if a packet does not match criteria, the packet is dropped.
- AFM is a stateful packet inspection (SPI) firewall. This means that BIG-IP is aware of new packets coming to/from BIG-IP, existing packets, and rogue packets.
- AFM adds more than 100 L2-4 denial of service attack vector detections and mitigations. This may be combined with ASM to provide L4-7 protection.
- Application Delivery Firewall is the service defense in depth layering mentioned earlier. On top of a simple L4 network firewall, you may add access policy and controls from L4-7 with APM (Access Policy Manager), or add L7 deep packet inspection with ASM (web application firewall), You can add DNS DOS mitigation with LTM DNS Express and GTM + DNSSEC. These modules make up the entire Application Delivery Firewall (ADF) solution.

1.2.5 Creating AFM Network Firewall Rules

For this lab, you will complete the following sections:

Default Actions

The BIG-IP® Network Firewall provides policy-based access control to and from address and port pairs, inside and outside of your network. Using a combination of contexts, the network firewall can apply rules in many ways, including: at a global level, on a per-virtual server level, and even for the management port or a self IP address. Firewall rules can be combined in a firewall policy, which can contain multiple context and address pairs, and is applied directly to a virtual server.

By default, the Network Firewall is configured in **ADC mode**, a default allow configuration, in which all traffic is allowed through the firewall, and any traffic you want to block must be explicitly specified.

The system is configured in this mode by default so all traffic on your system continues to pass after you provision the Advanced Firewall Manager. You should create appropriate firewall rules to allow necessary traffic to pass before you switch the Advanced Firewall Manager to Firewall mode. In **Firewall mode**, a default deny configuration, all traffic is blocked through the firewall, and any traffic you want to allow through the firewall must be explicitly specified.

This lab has been pre-configured in **Firewall mode**.

You can change the BIG-IP AFM Network Firewall mode by modifying the Default Firewall Action setting. When you enable Firewall mode, the AFM system allows access only when specific firewall rules are put in place. While this method reduces the overall attack surface, it may impact services that you are not be aware of. ADC mode is currentl the default and most popular choice. These steps change the AFM mode from the default ADC mode to firewall mode.

The BIG-IP® Network Firewall provides policy-based access control to and from address and port pairs, inside and outside of your network. By default, the network firewall is configured in ADC mode, which is a **default allow** configuration, in which all traffic is allowed to virtual servers and self IPs on the system, and

any traffic you want to block must be explicitly specified. This applies only to the Virtual Server & Self IP level on the system.

Important: Even though the system is in a default allow configuration, if a packet matches no rule in any context on the firewall, a Global Drop rule drops the traffic.

Rule Hierarchy

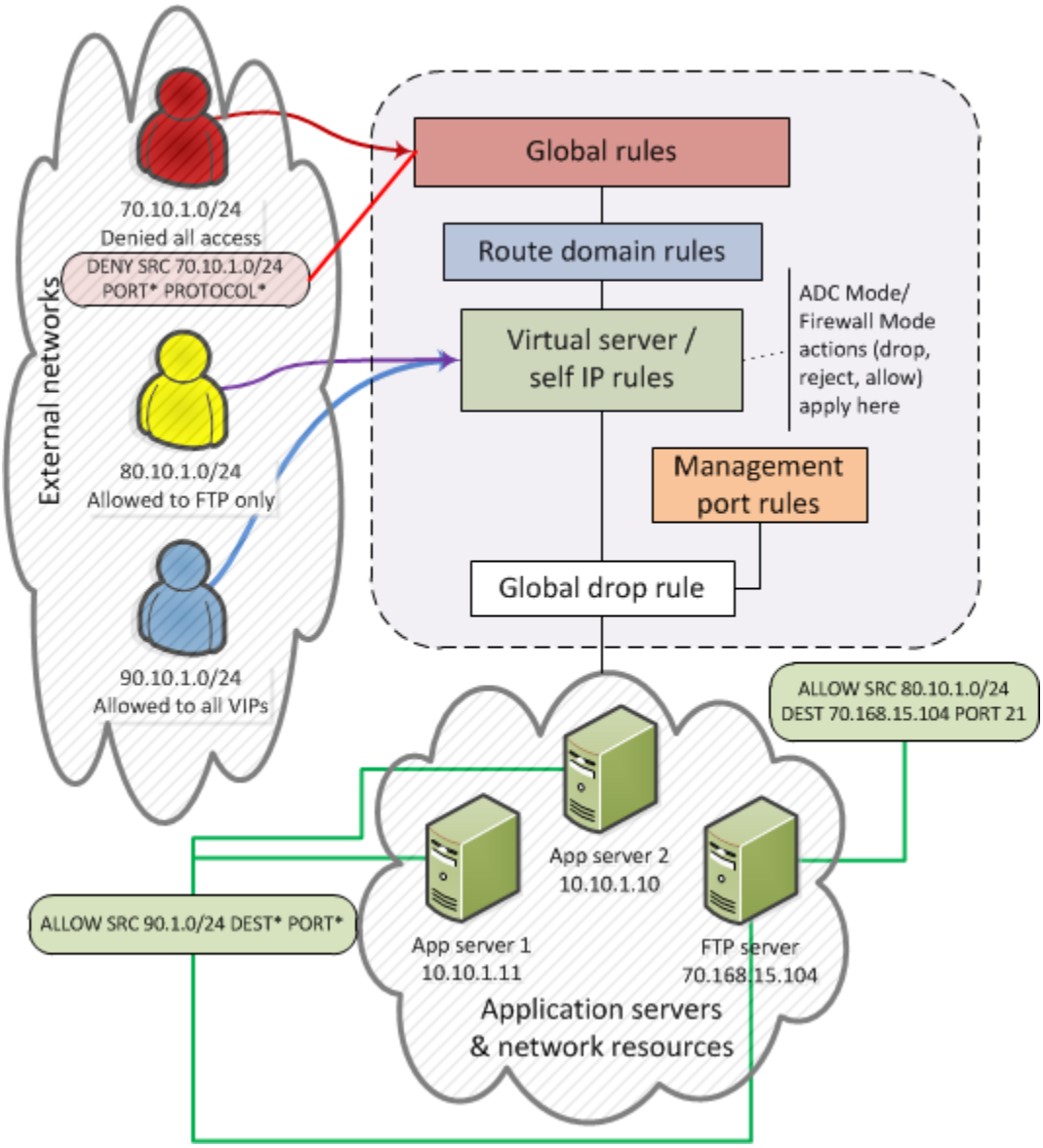
With the BIG-IP® Network Firewall, you use a context to configure the level of specificity of a firewall rule or policy. For example, you might make a global context rule to block ICMP ping messages, and you might make a virtual server context rule to allow only a specific network to access an application.

Context is processed in this order:

- Global
- Route domain
- Virtual server / self IP
- Management port*
- Global drop*

The firewall processes policies and rules in order, progressing from the global context, to the route domain context, and then to either the virtual server or self IP context. Management port rules are processed separately, and are not processed after previous rules. Rules can be viewed in one list, and viewed and reorganized separately within each context. You can enforce a firewall policy on any context except the management port. You can also stage a firewall policy in any context except management.

Tip: You cannot configure or change the Global Drop context. The Global Drop context is the final context for traffic. Note that even though it is a global context, it is not processed first, like the main global context, but last. If a packet matches no rule in any previous context, the Global Drop rule drops the traffic.



pyCreate and View Log Entries

In this section, you will generate various types of traffic through the firewall as you did previously, but now you will view the log entries using the network firewall log. Open your web browser and once again try to access http://10.1.20.11. Also, try to ping 10.1.20.11.

Open the **Security > Event Logs > Network > Firewall** page on bigip01.f5demo.com (10.1.1.4). The log file shows the ping requests are being accepted and the web traffic is being dropped:

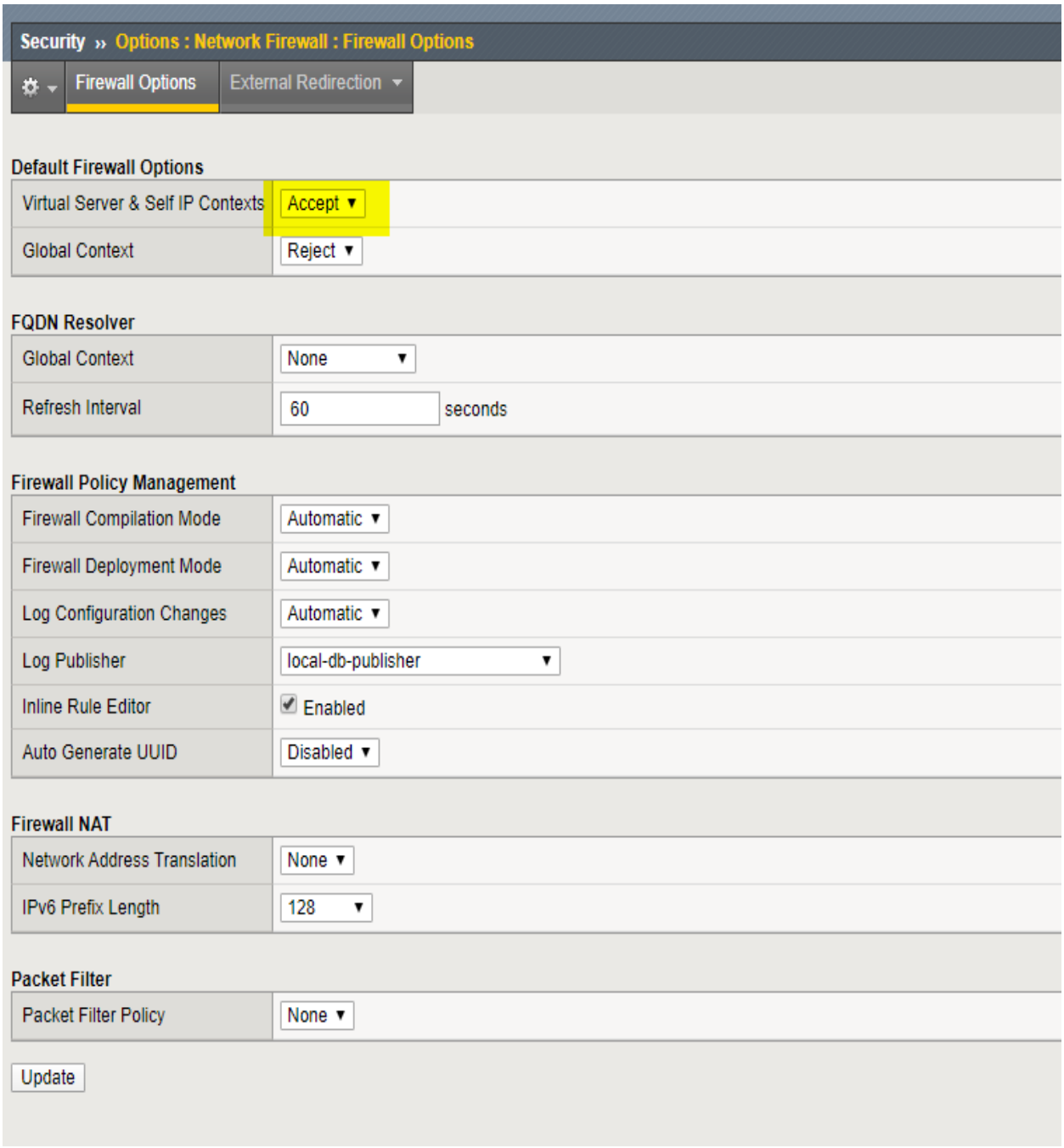
Time	Context	Name	Policy Type	Policy Name	Rule	Subscriber ID	Subscriber Group	Region	FQDN	Address	Port	VLAN / Tunnel	Region	FQDN	Address	Port	Route Domain	Virtual Server	Protoc
2018-06-20 02:11:39	Global	/Common/global-firewall-rules	Enforced	/Common/Global	Global_Drop	unknown	unknown	Unknown	unknown	10.20.0.200	51507	/Common/OUTSIDE	Unknown	unknown	10.30.0.50	80	0		TCP
2018-06-20 02:11:38	Global	/Common/global-firewall-rules	Enforced	/Common/Global	Global_Drop	unknown	unknown	Unknown	unknown	10.20.0.200	51506	/Common/OUTSIDE	Unknown	unknown	10.30.0.50	80	0		TCP
2018-06-20 02:11:38	Global	/Common/global-firewall-rules	Enforced	/Common/Global	Global_Drop	unknown	unknown	Unknown	unknown	10.20.0.200	51505	/Common/OUTSIDE	Unknown	unknown	10.30.0.50	80	0		TCP
2018-06-20 02:11:33	Global	/Common/global-firewall-rules	Enforced	/Common/Global	Global_Drop	unknown	unknown	Unknown	unknown	10.20.0.200	51507	/Common/OUTSIDE	Unknown	unknown	10.30.0.50	80	0		TCP
2018-06-20 02:11:32	Global	/Common/global-firewall-rules	Enforced	/Common/Global	Global_Drop	unknown	unknown	Unknown	unknown	10.20.0.200	51506	/Common/OUTSIDE	Unknown	unknown	10.30.0.50	80	0		TCP
2018-06-20 02:11:32	Global	/Common/global-firewall-rules	Enforced	/Common/Global	Global_Drop	unknown	unknown	Unknown	unknown	10.20.0.200	51505	/Common/OUTSIDE	Unknown	unknown	10.30.0.50	80	0		TCP
2018-06-20 02:11:30	Global	/Common/global-firewall-rules	Enforced	/Common/Global	Global_Drop	unknown	unknown	Unknown	unknown	10.20.0.200	51507	/Common/OUTSIDE	Unknown	unknown	10.30.0.50	80	0		TCP
2018-06-20 02:11:29	Global	/Common/global-firewall-rules	Enforced	/Common/Global	Global_Drop	unknown	unknown	Unknown	unknown	10.20.0.200	51506	/Common/OUTSIDE	Unknown	unknown	10.30.0.50	80	0		TCP
2018-06-20 02:11:29	Global	/Common/global-firewall-rules	Enforced	/Common/Global	Global_Drop	unknown	unknown	Unknown	unknown	10.20.0.200	51505	/Common/OUTSIDE	Unknown	unknown	10.30.0.50	80	0		TCP
2018-06-20 02:11:17	Global	/Common/global-firewall-rules	Enforced	/Common/Global	Ping	unknown	unknown	Unknown	unknown	10.20.0.200 10.20.0.200	1 1	/Common/OUTSIDE /Common/DMZ	Unknown	unknown	10.30.0.50 10.30.0.50	2048 2048	0 0		ICMP ICMP

Although we will not configure external logging in this lab, you should be aware that the BIG-IP supports high speed external logging in various formats including **SevOne**, **Splunk** and **ArcSight**.

Navigate ** Security > Options > Network Firewall > Firewall Options **

Default Firewall options configuration determine if the system is in ADC mode or Firewall Mode. In the screenshot below note the Virtual Server & Self IP Contexts Value. If it is set to Accept (system default) the Firewall is in ADC mode. For ths lab we will use Firewall Mode with the value set to Reject

Local-db-publisher is linked to the global-network logging profile in the next step



Add a log publisher to the log configuration

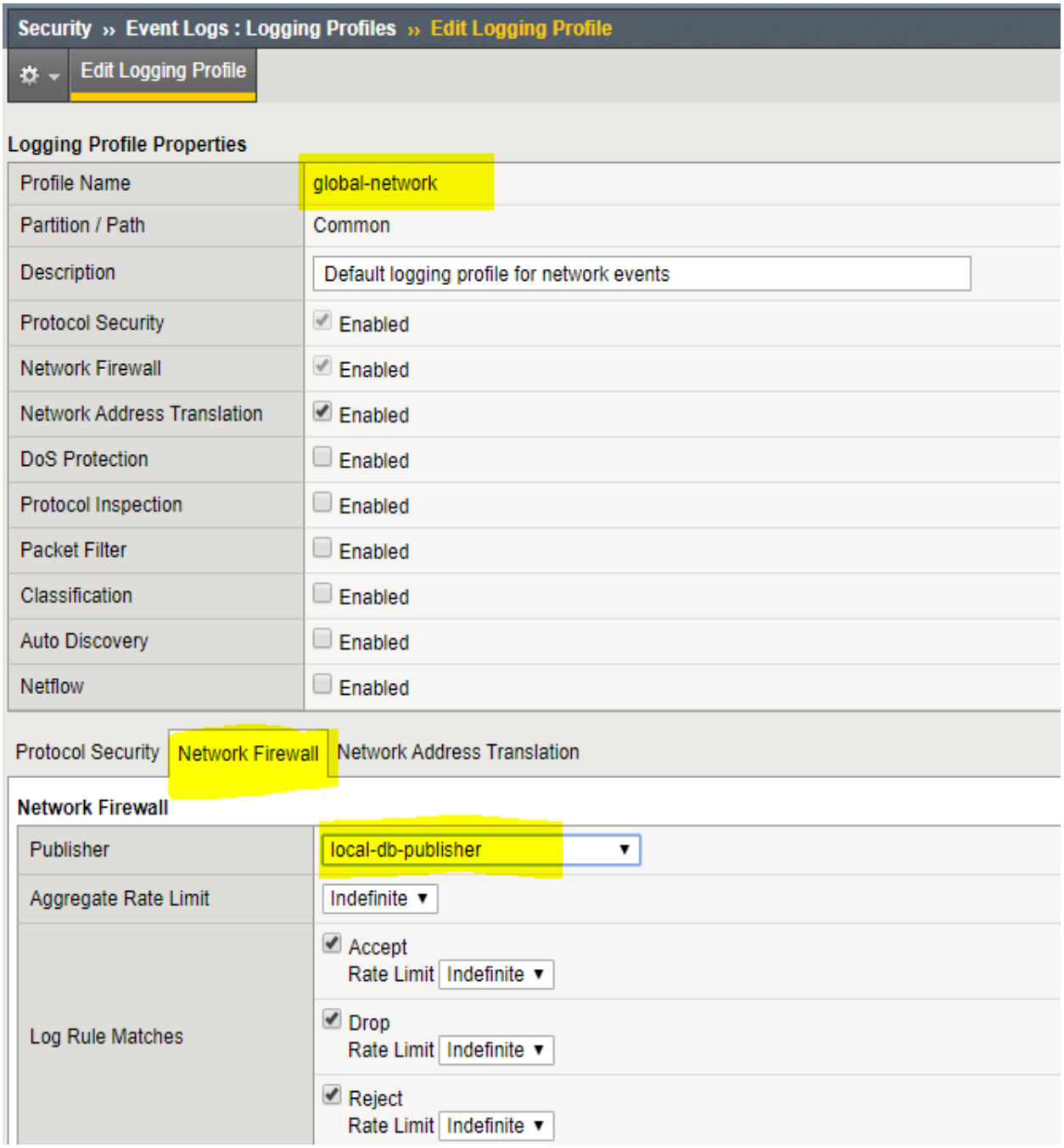
Navigate Security>>Event Logs>>Logging Profiles

Navigate Select **Global Network**

Navigate Click on the **Network Firewall** Tab

Navigate Use the publisher pulldown to select **local-db-publisher**

Review the configuration. The Storage Format section allows you to select the values included in the log.



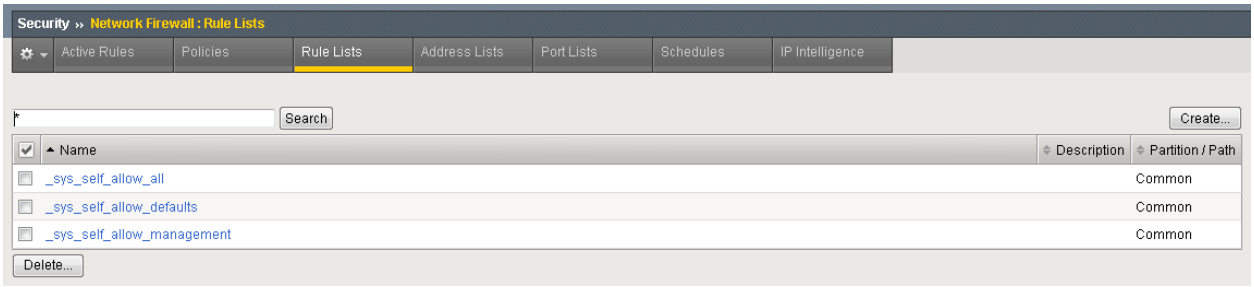
Create a Rule List

Rule lists are a way to group a set of individual rules together and apply them to the active rule base as a group. A typical use of a rule list would be for a set of applications that have common requirements for access protocols and ports. As an example, most web applications would require TCP port 80 for HTTP and TCP port 443 for SSL/TLS. You could create a Rule list with these protocols, and apply them to each of your virtual servers.

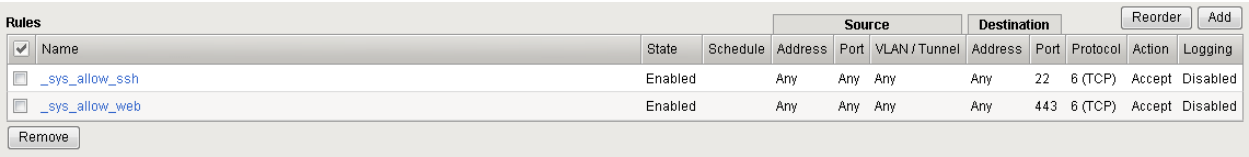
Let's examine some of the default rule lists that are included with AFM.

Go to **Security >Network Firewall > Rule Lists**. They are:

- _sys_self_allow_all
- _sys_self_allow_defaults
- _sys_self_allow_management



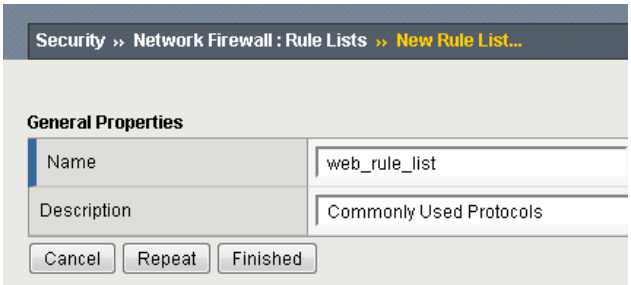
If you click on **_sys_self_allow_management** you'll see that it is made up of two different rules that will allow management traffic (port 22/SSH and port 443 HTTPS). Instead of applying multiple rules over and over across multiple servers, you can put them in a rule list and then apply the rule list as an ACL.



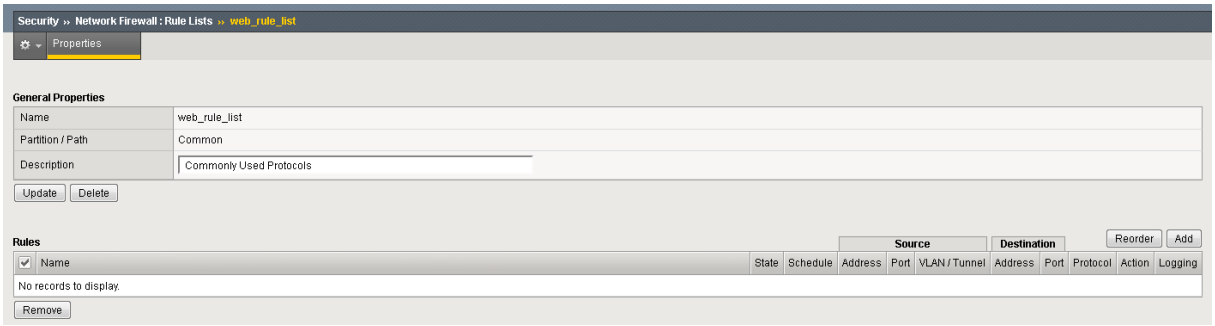
On bigip01.f5demo.com (10.1.1.4) create a rule list to allow Web traffic. A logical container must be created before the individual rules can be added. You will create a list with two rules, to allow port 80 (HTTP) to servers 10.1.20.11 through 10.1.20.13 and another to allow port 443 (HTTPS) to servers 10.1.20.13 through 10.1.20.15. First you need to create a container for the rules by going to:

Security > Network Firewall > Rule Lists and select **Create.**

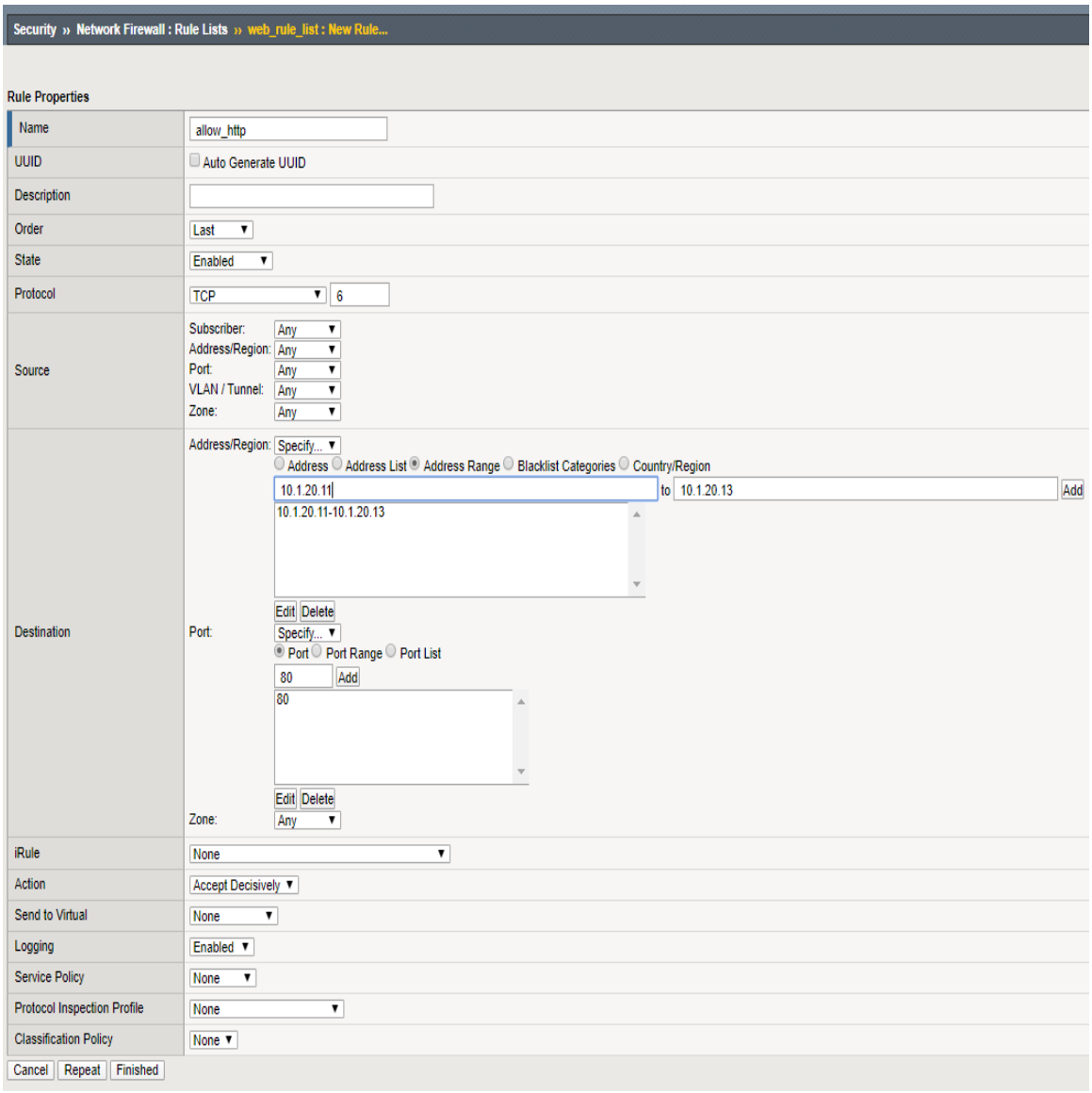
For the **Name** enter **web_rule_list**, provide an optional description and then click **Finished.**



Edit the **web_rule_list** by selecting it in the Rule Lists table, then click the **Add** button in the Rules section. Here you will add two rules into the list; the first is a rule to allow HTTP.



Name	allow_http
Protocol	TCP
Source	Leave at Default of **Any**
Destination Address	**Specify. . .** 10.1.20.11, 10.1.20.13 then click **Add**
Destination Port	**Specify. . .** Port **80**, then click **Add**
Action	**Accept-Decisively**
Logging	Enabled



Select **Repeat** when done.

Name	allow_https
Protocol	TCP
Source	Leave at Default of **Any**
Destination Address	**Specify...** 10.1.20.14, 10.1.20.15 then click **Add**
Destination Port	**Specify...** Port **443**, then click **Add**
Action	**Accept-Decisively**
Logging	Enabled

Select **Finished** when completed. When you exit, you'll notice the reject rule is after the **allow_https** rule. This means that HTTP traffic from 10.1.20.0/24 will be accepted, while all other traffic from this subnet will be rejected based on the ordering of the rules as seen below:

Create a Policy with a Rule List

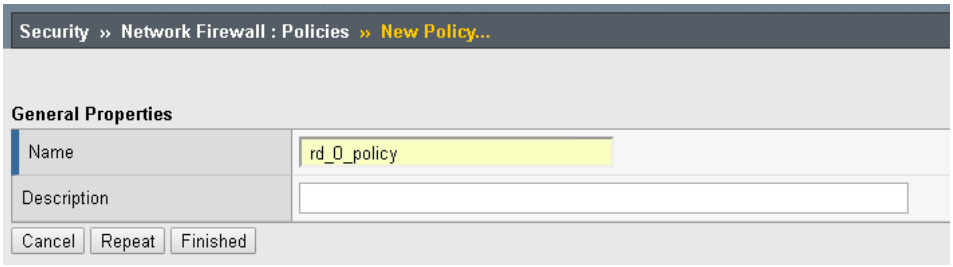
Policies are a way to group a set of individual rules together and apply them to the active policy base as a group. A typical use of a policy list would be for a set of rule lists that have common requirements for access protocols and ports.

Create a policy list to allow the traffic you created in the rule list in the previous section. A logical container must be created before the individual rules can be added. First you need to create a container for the policy by going to:

Security > Network Firewall > Policies and select **Create.**

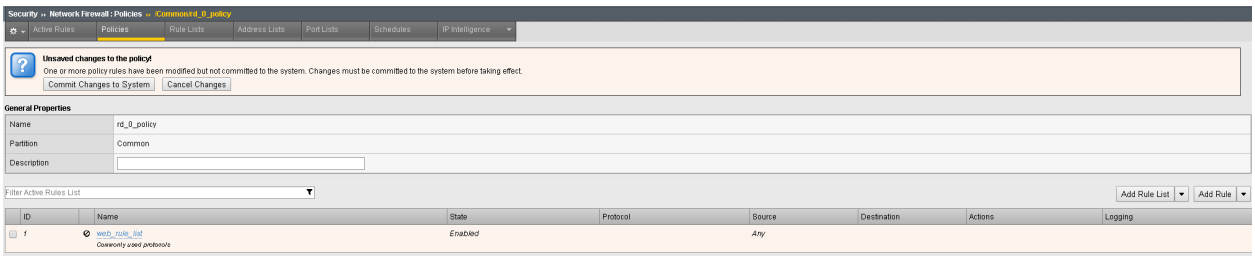
You'll notice that before Milton detached from Initech, he created a global policy named **'Global'** to allow basic connectivity to make troubleshooting easier.

For the **Name** enter **rd_0_policy**, provide an optional description and then click **Finished. (Note: We commonly use "RD" in our rules to help reference the "Route Domain", default is 0)**



Edit the **rd_0_policy** by selecting it in the Policy Lists table, then click the **Add Rule List** button. Here you will add the rule list you created in the previous section. For the **Name,** start typing **web_rule_list**, you will notice the name will auto complete, select the rule list /**Common/web_rule_list**, provide an optional description and then click **Done Editing.**

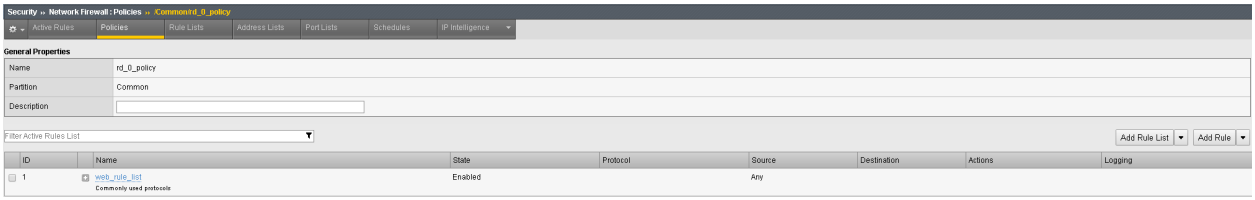
When finished your policy should look like the screen shot below.



You will notice the changes are unsaved and need to be committed to the system. This is a nice feature to have enabled to verify you want to commit the changes you've just made without a change automatically being implemented.

To commit the change, simply click **"Commit** Changes **to System"** located at the top of the screen.

Once committed you'll notice the rule now becomes active and the previous commit warning is removed.



Add the Rule List to a Route Domain

In this section, you are going to attach the rule to a route domain using the **Security** selection in the top bar within the **Route Domain** GUI interface.

Go to **Network**, then click on **Route Domains**, then select the hyperlink for route domain **0**.

Now click on the **Security** top bar selection, which is a new option that was added in version 11.3.

In the Network Firewall section, set the Enforcement: to **"Enabled. . . ".**

Select the Policy you just created, "**rd_0_policy**" and click Update.

Review the rules that are now applied to this route domain by navigating to:

Security > Network Firewall > Active Rules.

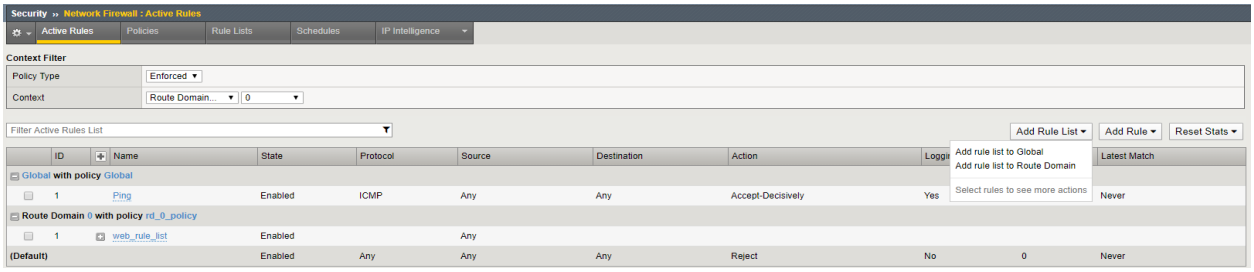
From the **Context Filter** select **Route Domain 0**.

Click on the **Add Rule List to Global** from the ypper right corner of the screen and click **Cancel** (note:this is a GUI bug)

Click on the **Add Rule List to Route Domain** from the ypper right corner of the screen and click **Cancel** (note:this is a GUI bug)

your screen should show the web_rule _list you assigned earlier through the Route Domain Screen.

The sceen should look similar to the below screen shot.



Test the New Firewall Rules

Once again you will generate traffic through the BIG-IP AFM and then view the AFM (firewall) logs.

- Ping 10.1.20.11, 10.1.20.12, 10;120.13, 10.120.14, and 10.1.20.15 (why does ths work?)

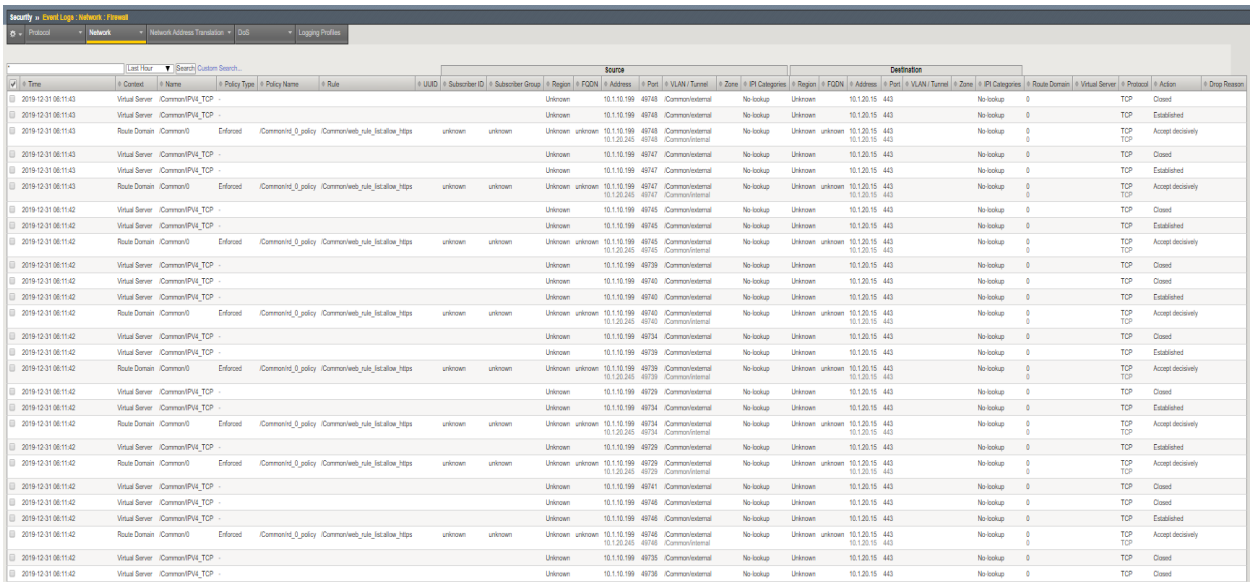
- Open a new Web browser and access http://10.1.20.11
- Open a new Web browser and access https://10.1.20.11 (this should fail–why?)
- Open a new Web browser and access https://10.1.20.14

In the Configuration Utility, open the **Security > Event Logs > Network > Firewall** page.

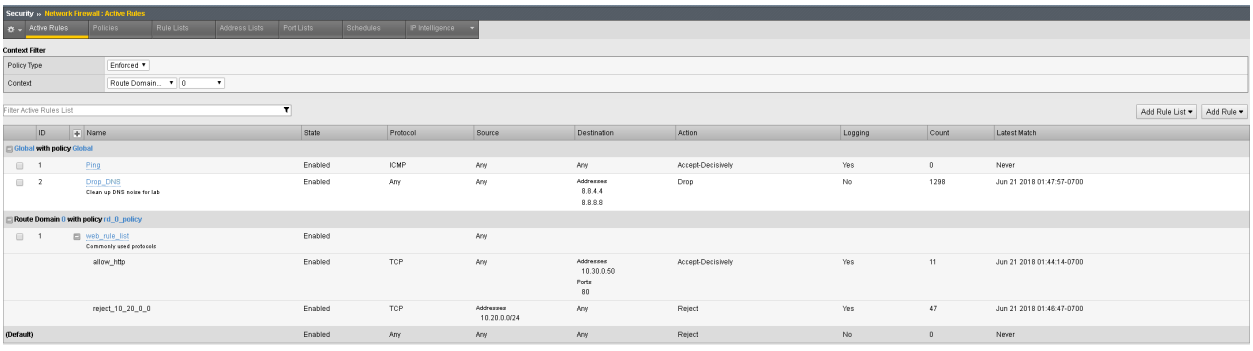
Access for ports 80 / 443 was granted to a host using the web_rule_list: **allow_http** and **allow_https**rule.

Note the source address of the user (10.1.10.199). The IP forwarding VIPs are configured with SNAT auto-map. Packets forwarded by the BIG-IP to the servers have a source address 10.1.20.245. This arrangement is common in cloud deplouments since it simplifies traffic routing.

Denied connections are not logged in this configuration. These are dropped by the final reject rule in the global policy



You may verify this, by going to **Security > Network Firewall > Active Rules**, then selecting the context for route domain 0. Note the **Count** field next to each rule as seen below. Also note how each rule will also provide a **Latest Matched** field so you will know the last time each rule was matched: (Investigating Counter behavior)



Creating an Additional Rule List for Additional Services

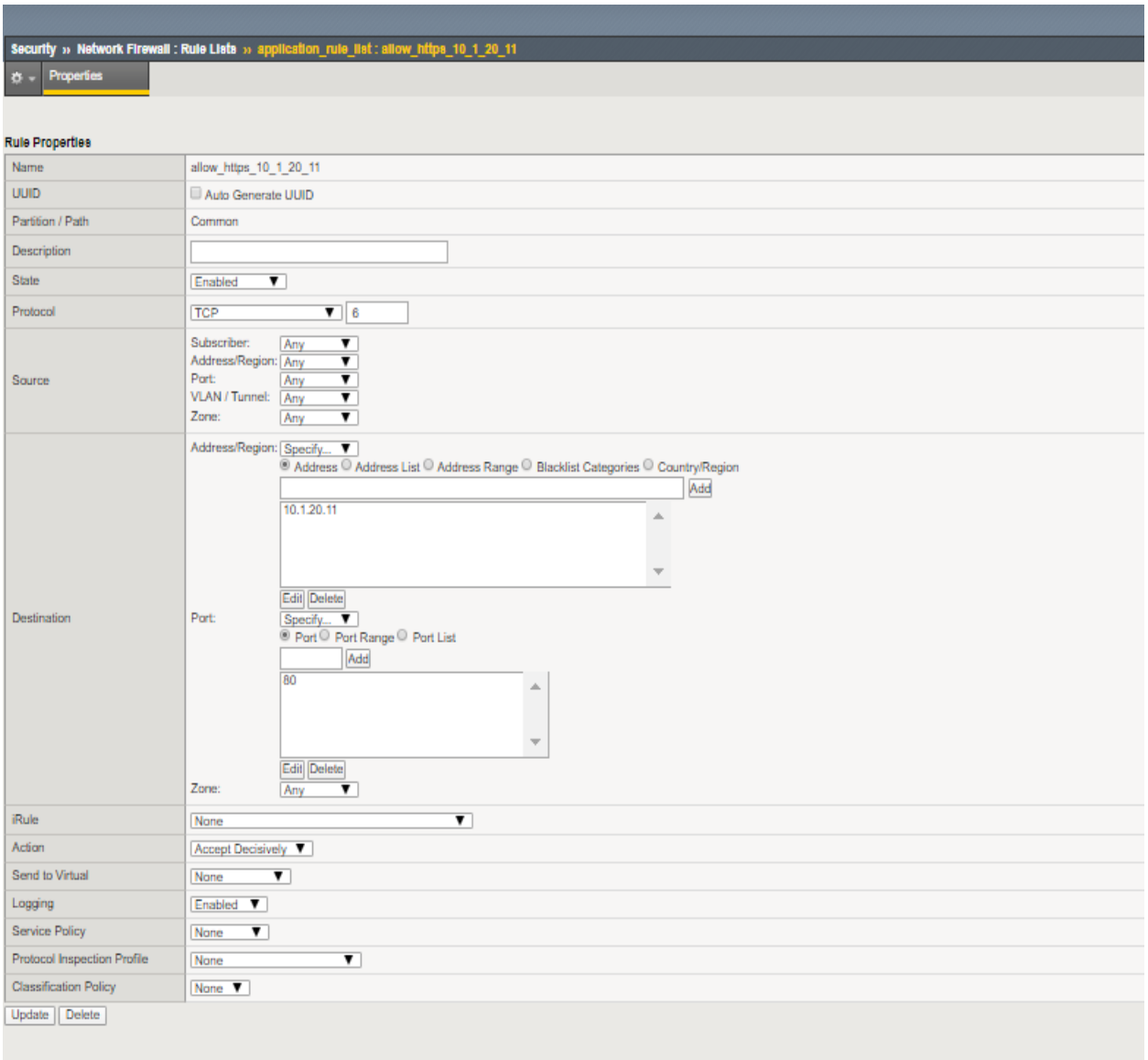
Rules and Rule Lists can also be created and attached to a context from the Active Rules section of the GUI. Go to the

Security > Network Firewall > Rule Lists

Create a **Rule List** called **application_rule_list** then click **Finished**.

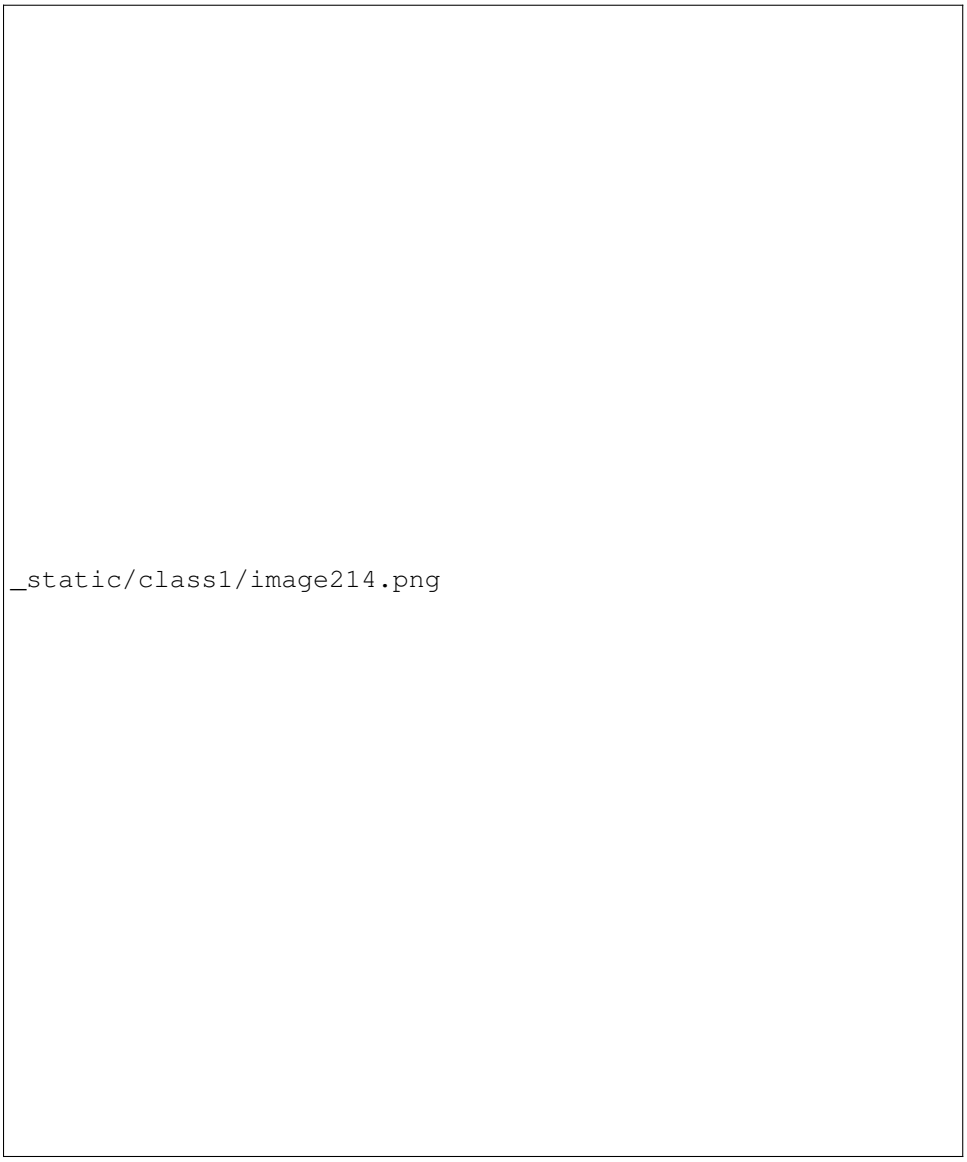
Enter the rule list by clicking on its hyperlink, then in the **Rules** section click **Add**, and add the following information, then click **Finished**.

Name	allow_http_8081 10.1.20.11
Protocol	TCP
Source	Leave at Default of **Any**
Destination Address	**Specify...** 10.1.20.11, then click **Add**
Destination Port	**Specify...** Port **8081**, then click **Add**
Action	**Accept-Decisively**
Logging	Enabled



Enter the rule list by clicking on its hyperlink, then in the **Rules** section click **Add**, and add the following information, then click **Finished**.

Name	allow_ssh 10.1.20.11
Protocol	TCP
Source	Leave at Default of **Any**
Destination Address	**Specify...** 10.1.20.11, then click **Add**
Destination Port	**Specify...** Port **22**, then click **Add**
Action	**Accept-Decisively**
Logging	Enabled



Add Another Rule List to the Policy

Use the **Policies** page to add the new firewall rule list to the **rd_0_policy**.

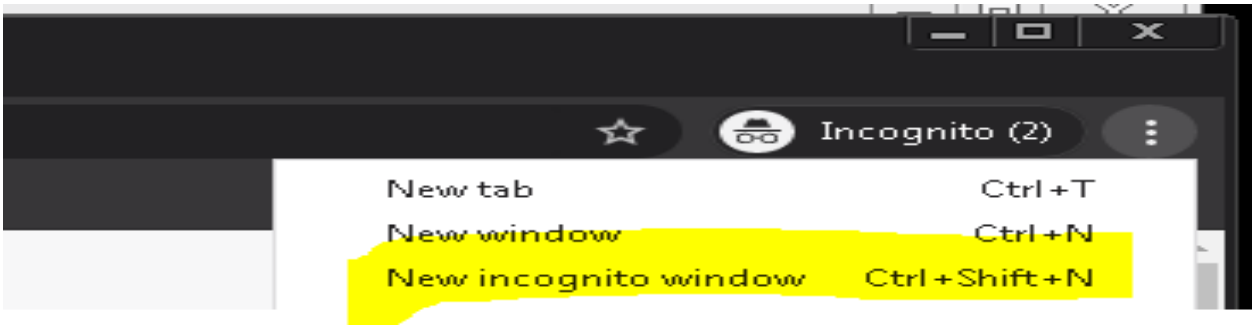
Open the **Security > Network Firewall > Policies** page.

Click on the policy name to modify the policy.

The only current active rule list is for the **web_policy**. Click on the arrow next to **Add Rule List, then select, Add the rule list AT END)** to add the new rule list you just created. For **Name** begin typing 'application_rule_list', select /Common/application_rule_list, then click **Done Editing**.

Remember to Commit the changes to system before proceeding.

Once completed, you should see a policy similar to the one below:



Review the rules that are now applied to this route domain by navigating to:

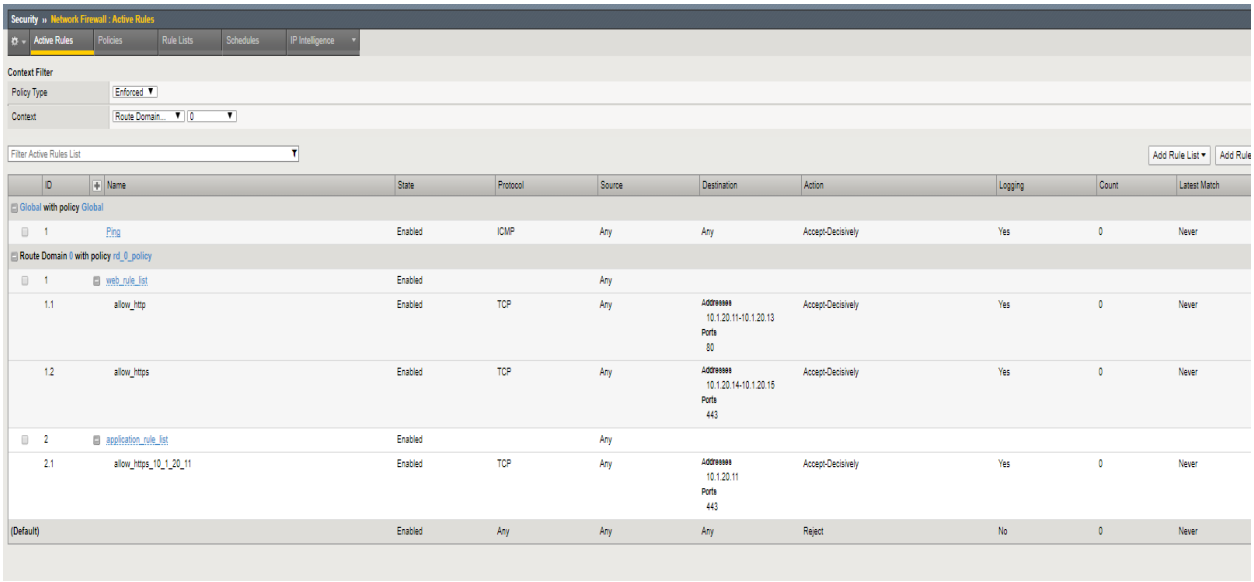
Security > Network Firewall > Active Rules.

From the **Context Filter** select **Route Domain 0**.

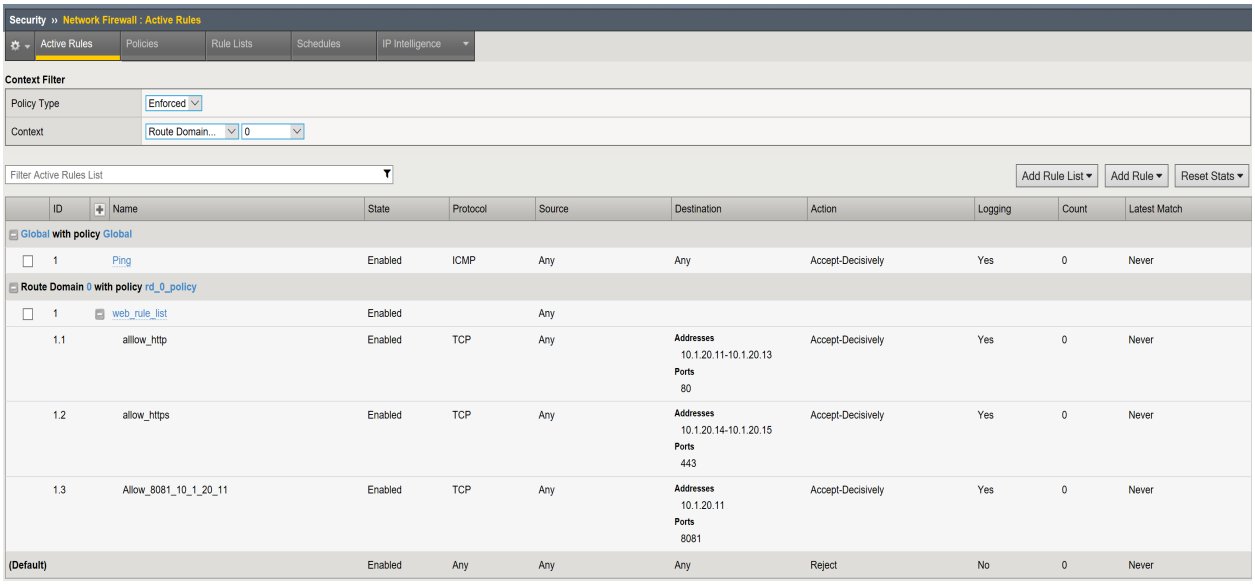
Click on the **Add Rule List to Global** from the ypper right corner of the screen and click **Cancel** (note:this is a GUI bug)

Click on the **Add Rule List to Route Domain** from the ypper right corner of the screen and click **Cancel** (note:this is a GUI bug)

your screen should show the web_rule _list you assigned earlier through the Route Domain Screen.



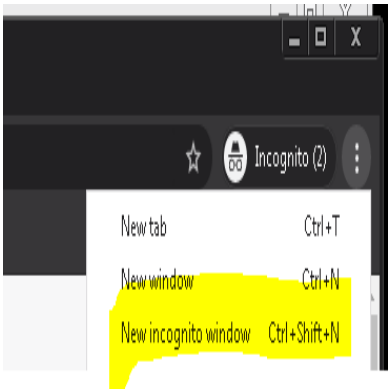
The new ordering should look something like the screen shot below:



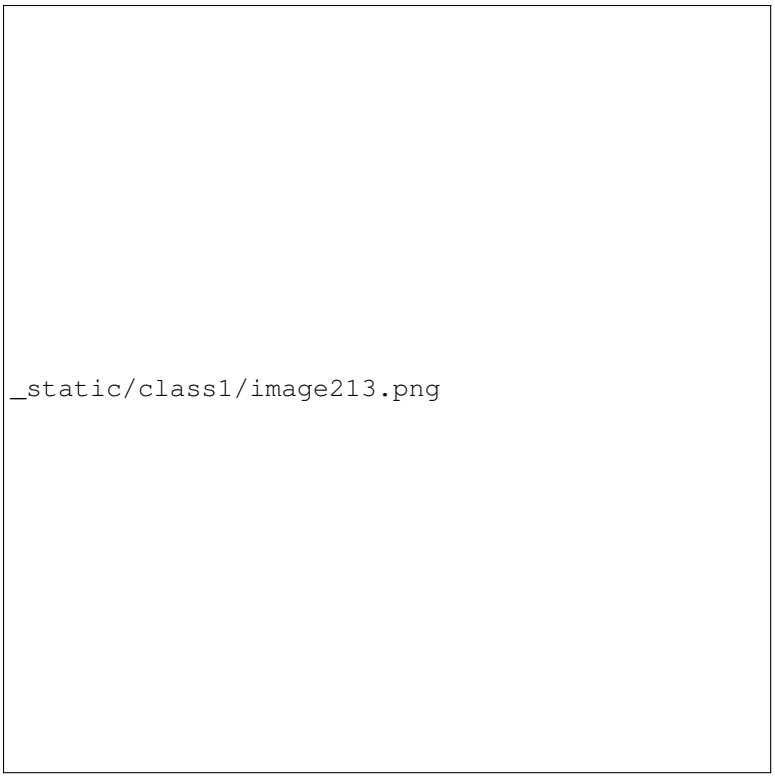
Test Access to the Servers

- Open the incognito Web browser and access http://10.1.20.11:8081
- Open a new Web browser and access http://10.1.20.11

Success!!



- Next open Putty Application on the Desktop, select Lamp Server-10.1.20.11.
- For the login as: type in **f5** and hit **<Enter>**



- If you received a login prompt in your Putty terminal

Success!!

Test Server Access 8081 & SSH

Before we continue let's clean up the rules just a little for best practices. Use the **Rule Lists** page to consolidate the firewall rule **'web_rule_list'** with the **'application_rule_list'** since these rules would ttypically be in the same rule list

Open the **Security > Network Firewall > Polocies** page.

Select the RD_0_policy

Check the box in front of **'application_rule_list'** and press the Delete button

Commit Changes to System

Open the **Security > Network Firewall > RuleLists** page.

Check the box in front of **'application_rule_list'** and press the Delete button (2x-Confirm action)

Click on the rule list **'web_rule_list'** to modify the rule list.

Enter the rule list by clicking on its hyperlink, then in the **Rules** section click **Add**, and add the following information, then click **Repeat**.

Name	allow_http_8081_10_1_20_11
Protocol	TCP
Source	Leave at Default of **Any**
Destination Address	**Specify. . .** 10.1.20.11, then click **Add**
Destination Port	**Specify. . .** Port **8081**, then click **Add**
Action	**Accept-Decisively**
Logging	Enabled

Remove the IPaddress and Port, enter the following information, then click **Finished**.

Name	allow_ssh 10.1.20.12
Protocol	TCP
Source	Leave at Default of **Any**
Destination Address	**Specify. . .** 10.1.20.12, then click **Add**
Destination Port	**Specify. . .** Port **22**, then click **Add**
Action	**Accept-Decisively**
Logging	Enabled

Inspect the properties of the rule list to verify the changes were made

Review the rules that are now applied to this route domain by navigating to:

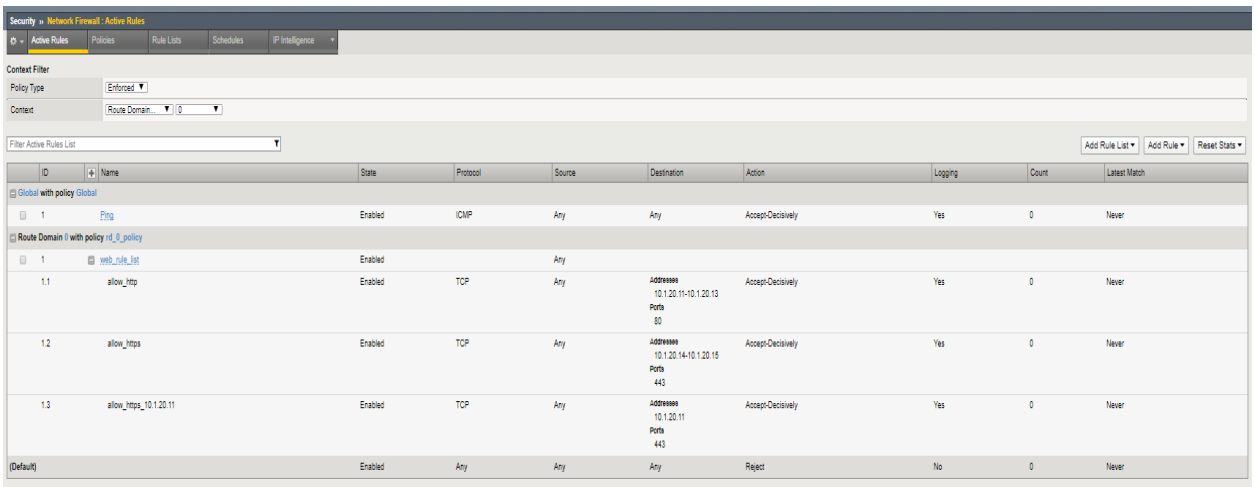
Security > Network Firewall > Active Rules.

From the **Context Filter** select **Route Domain 0**.

Click on the **Add Rule List to Global** from the upper right corner of the screen and click **Cancel** (note:this is a GUI bug)

Click on the **Add Rule List to Route Domain** from the ypper right corner of the screen and click **Cancel** (note:this is a GUI bug)

your screen should show the web_rule _list you assigned earlier through the Route Domain Screen.



Test the New Firewall Rules

Once again you will generate traffic through the BIG-IP AFM and then view the AFM (firewall) logs.

- Ping 10.1.20.11

- Open a new Web browser and access http://10.1.20.11
- Open a new Web browser and access http://10.1.20.11:8081
- Open a new Web browser and access https://10.1.20.12 (site cant be reached)
- Next open Putty Application on the Desktop, select Lamp Server-10.1.20.12. login as: type in **f5** and **<Enter>**

In the Configuration Utility, open the **Security > Event Logs > Network > Firewall** page.

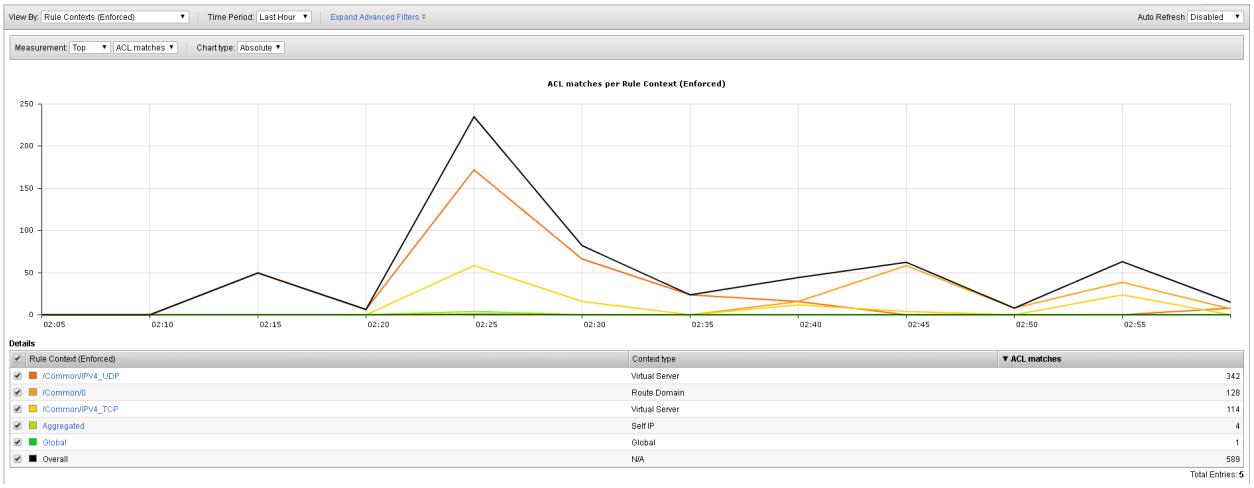
Inspect for the expected log entries

During this lab we have used Rules/Rule Lists applied to global and Route Domain objects. This is typical in a “Data Center” firewall implemntation where BIG-IP is positioned as a standalone firewall.

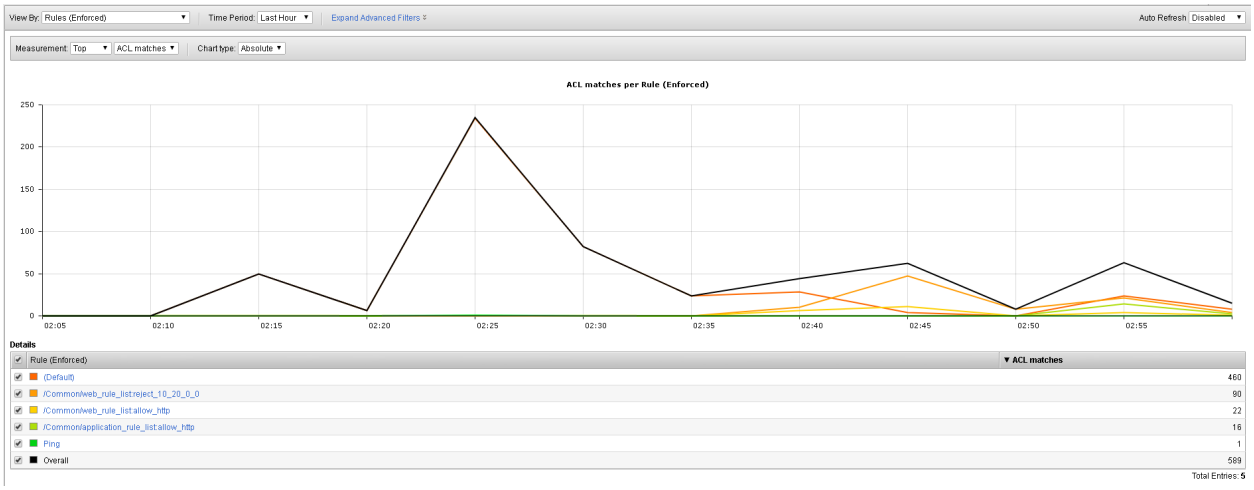
The BIG-IP Firewall module can also be used on a BIG-IP configured as an Application Delivery Controller/Load Balancer. For these environments additional granularity and East - West traffic control can be implemented by applying Ruls/Rule Lists to specific Virtual Servers or Self-IP’s

View Firewall Reports

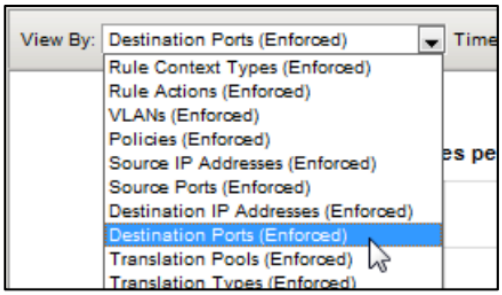
View several of the built-in network firewall reports and graphs on the BIG-IP system. Open the **Security >Reporting > Network > Enforced Rules** page. The default report shows all the rule contexts that were matched in the past hour.



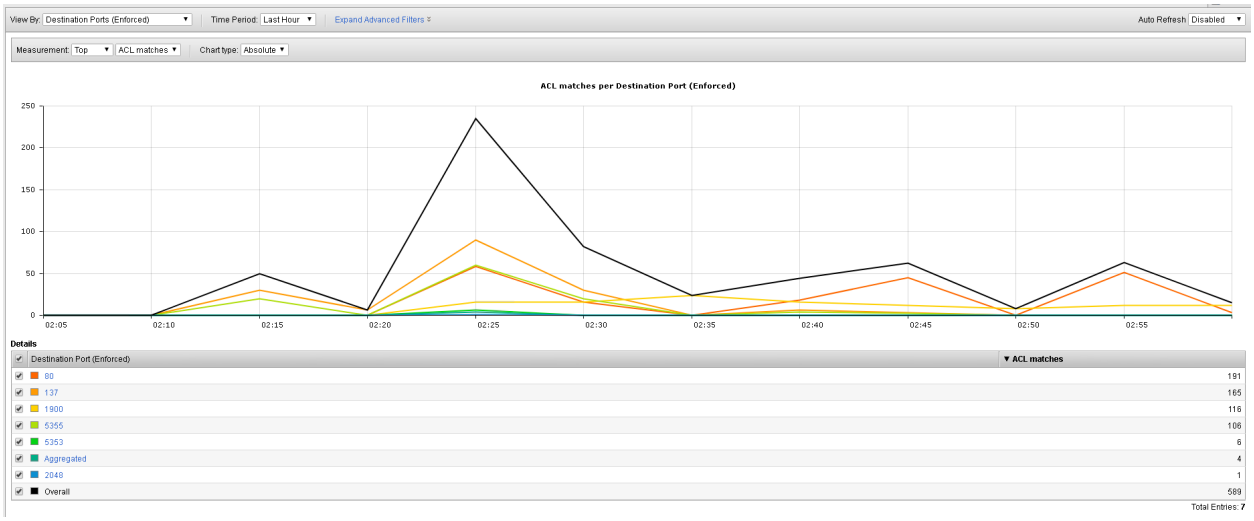
The default view gives reports per Context, in the drop-down menu select **Enforced Rules**.



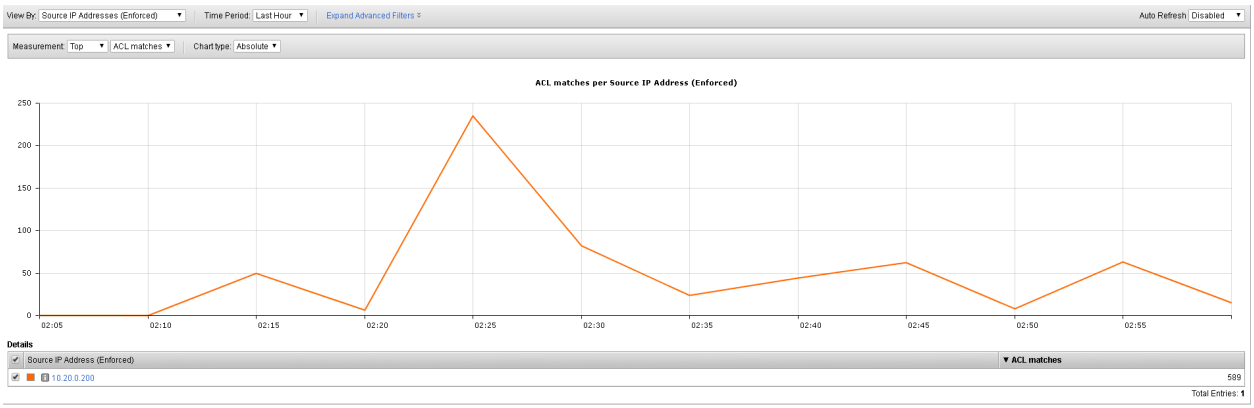
From the **View By** list, select **Destination Ports (Enforced)**.



This redraws the graph to report more detail for all the destination ports that matched an ACL.



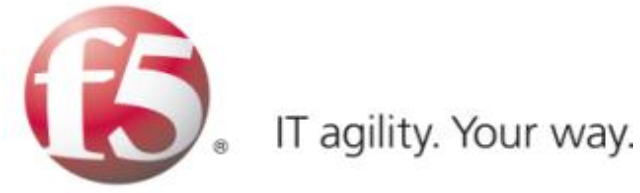
From the **View By** list, select **Source IP Addresses (Enforced)**. This shows how source IP addresses matched an ACL clause:



1.2.6 AFM Reference Material

- Network World Review of AFM: F5 data center firewall aces performance test:
  http://www.networkworld.com/reviews/2013/072213-firewall-test-271877.html
- AFM Product Details on **www.f5.com**:
  http://www.f5.com/products/big-ip/big-ip-advanced-firewall-manager/overview
- AFM Operations Guide:
  https://support.f5.com/content/kb/en-us/products/big-ip-afm/manuals/product/f5-afm-operations-guide/_jcr_content/pdfAttach/download/file.res/f5-afm-operations-guide.pdf

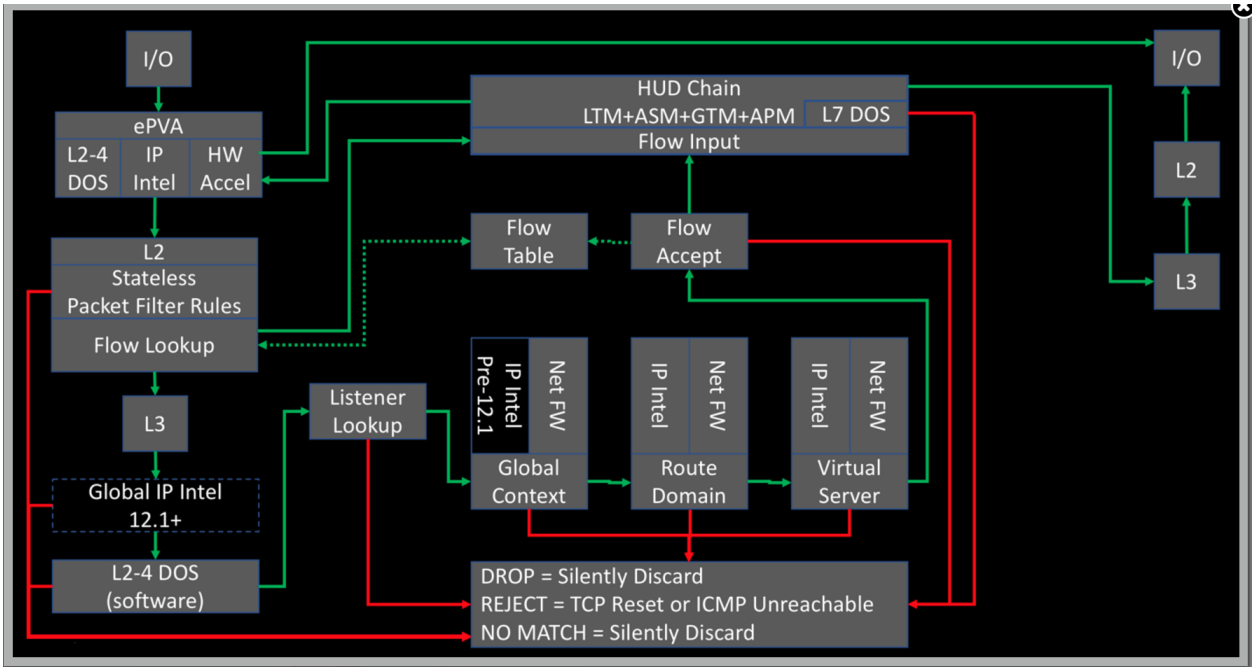
**Written for TMOS 15.1.0



1.3 Lab 2 - AFM Packet Tester, Flow Inspector, Stale Rule Lab

1.3.1 Lab Overview

New in the v13 release of the BIG-IP Advanced Firewall Manager is the capability to insert a packet trace into the internal flow so you can analyze what component within the system is allowing or blocking packets based on your configuration of features and rule sets.



The packet tracing is inserted at L3 immediately prior to the Global IP intelligence. Because it is after the L2 section, this means that:

- we cannot capture in tcpdump so we can't see them in flight, and
- no physical layer details will matter as it relates to testing.

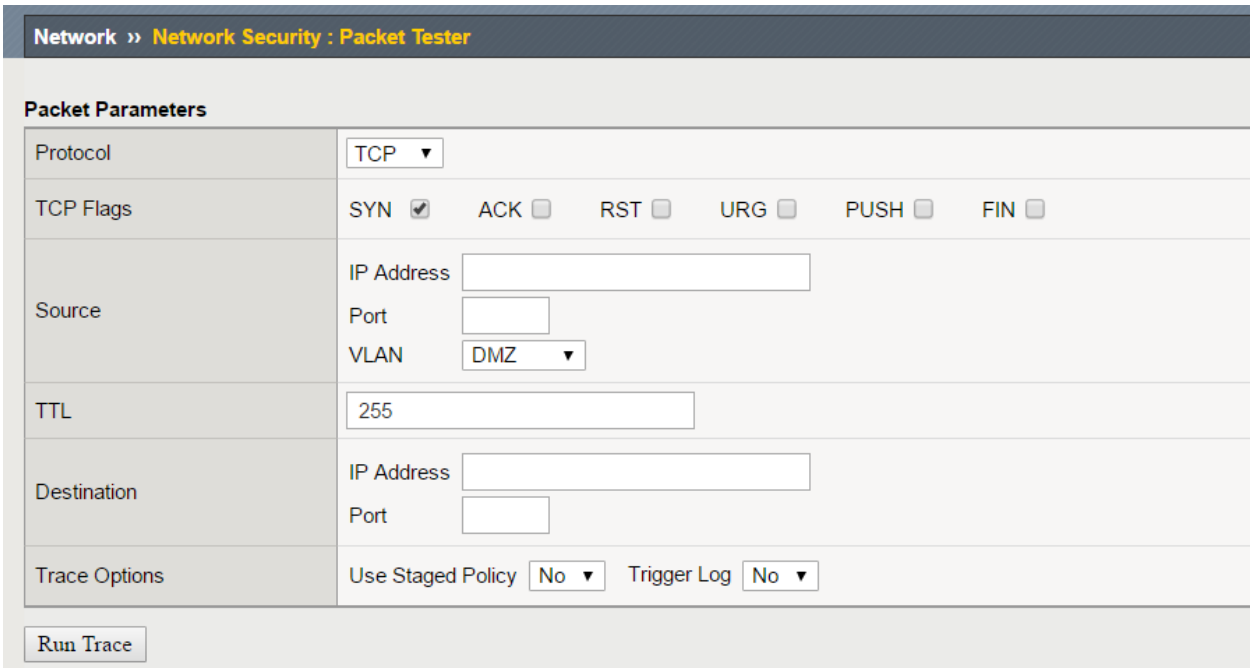
That said, it's incredibly useful for what is and is not allowing your packets through. You can insert tcp, udp, sctp, and icmp packets, with a limited set of (appropriate to each protocol) attributes for each.

1.3.2 Advanced Firewall Manager (AFM) Packet Tracer

Create and View Packet Tracer Entries

In this section, you will generate various types of traffic as you did previously, but now you will view the flow using the network packet tracer. Login to bigip01.f5demo.com

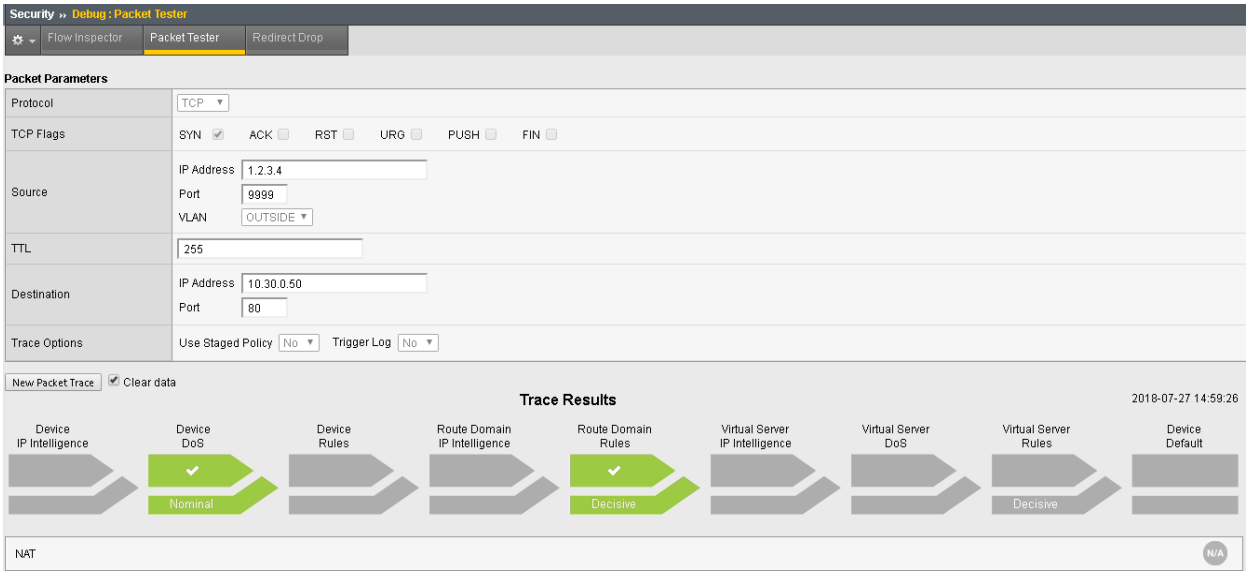
(10.1.1.4), navigate to **Security > Debug > Packet Tester**.



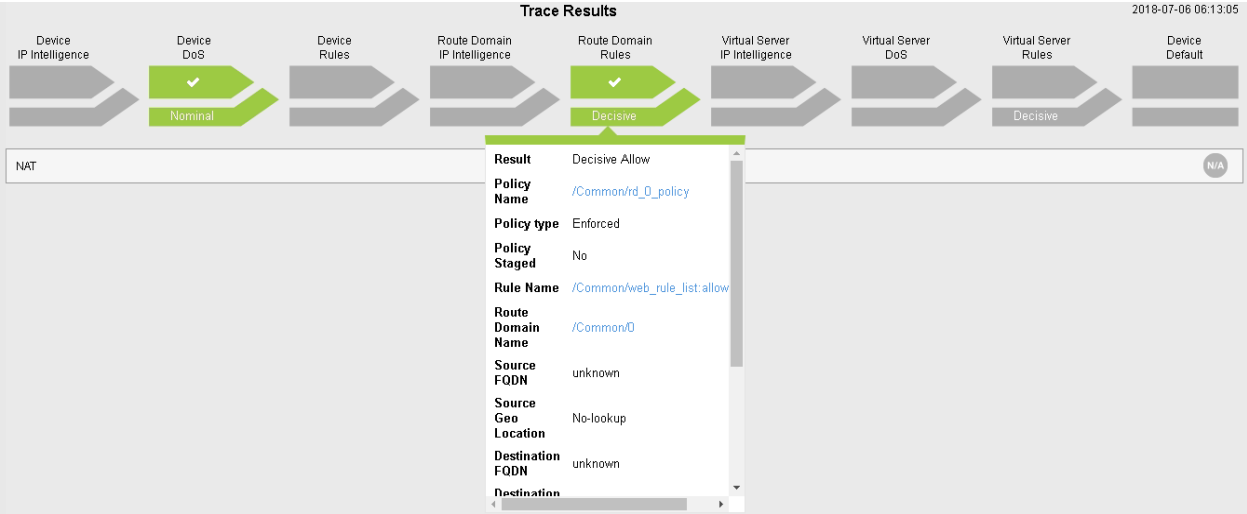
Create a packet test with the following parameters:

Protocol	TCP
TCP Flags	SYN
Source	IP - 1.2.3.4 Port – 9999 Vlan – external
TTL	255
Destination	IP – 10.1.20.11 Port - 80
Trace Options	Use Staged Policy – no Trigger Log - no

Click Run Trace to view the response. Your output should resmeble the allowed flow as shown below:



You can also click on the “Route Domain Rules” trace result and see which rule is permitting the traffic.

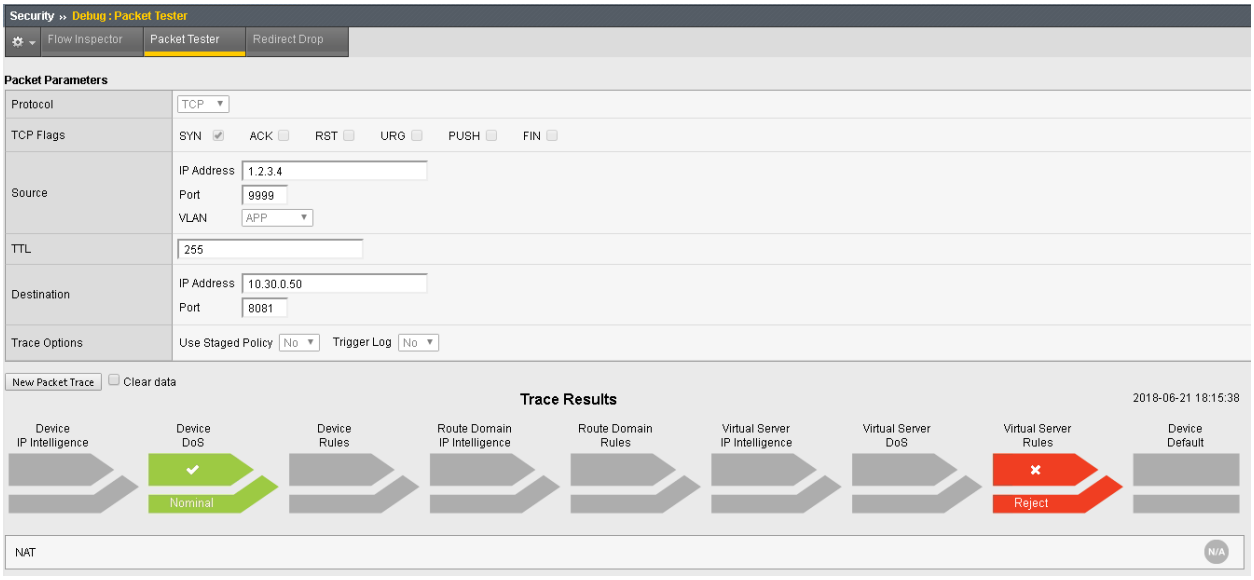


Click **New Packet Trace** (optionally do not clear the existing data – aka leave checked).

Create a packet test with the following parameters:

Protocol	TCP
TCP Flags	SYN
Source	IP - 1.2.3.4 Port – 9999 Vlan – Outside
TTL	255
Destination	IP – 10.1.20.11 Port - **8081**
Trace Options	Use Staged Policy – no Trigger Log - no

Click Run Trace to view the response. Your output should resemble the allowed flow as shown below:



Protocol	TCP
TCP Flags	SYN
Source	IP - 1.2.3.4 Port – 9999 Vlan – Outside
TTL	255
Destination	IP – 10.1.20.11 Port - **443**
Trace Options	Use Staged Policy – no Trigger Log - no

This traffic will be blocked by the default deny rule

This shows there is no rule associated with the route domain or a virtual server which would permit the traffic. As such, the traffic would be dropped/rejected.

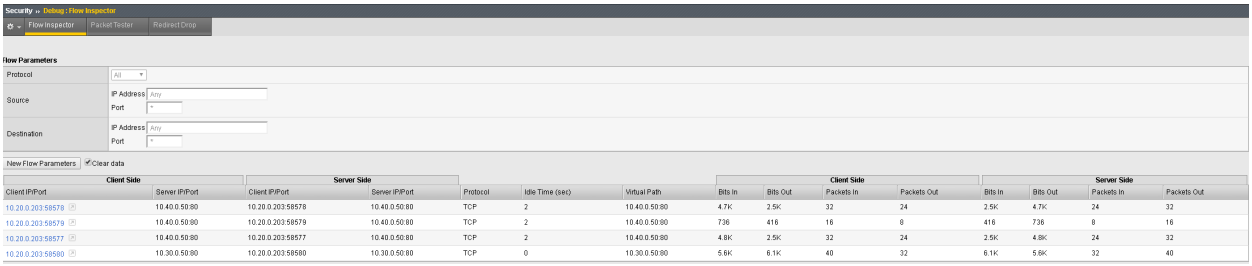
1.3.3 Advanced Firewall Manager (AFM) Flow Inspector

Create and View Flow Inspector Data

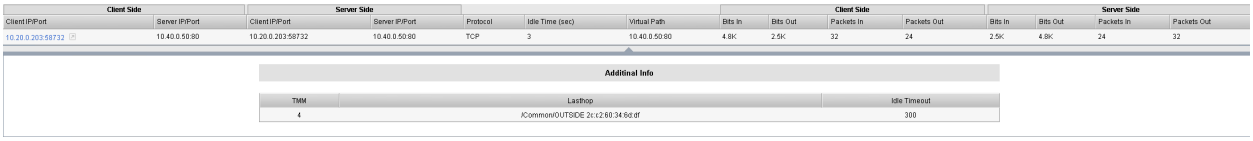
A new tool introduced in version 13 is the flow inspector. This tool is useful to view statistical information about existing flows within the flow table. To test the flow inspector, navigate to **Security > Debug > Flow Inspector.** Refresh the web page we've been using for testing (http://10.1.20.11) and click "Get Flows".

MK—SSH from jumphost to 10.1.20.11 (no login but session will show up in flows)

(mkurath—Opened incident C3173863 – I could not see any flow data)



Select a flow and click on the pop-out arrow for additional data.

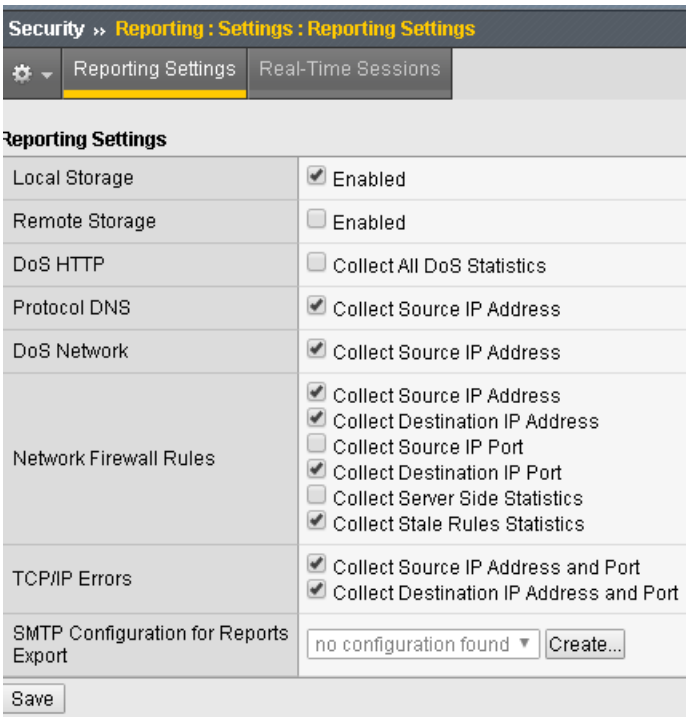


This will show the TMM this is tied to as well as the last hop and the idle timeout. This data is extremely valuable when troubleshooting application flows.

It is also worth noting you can click directly on the IP address of a flow to pre-populate the data in the packet tester for validating access and/or where the flow is permitted.

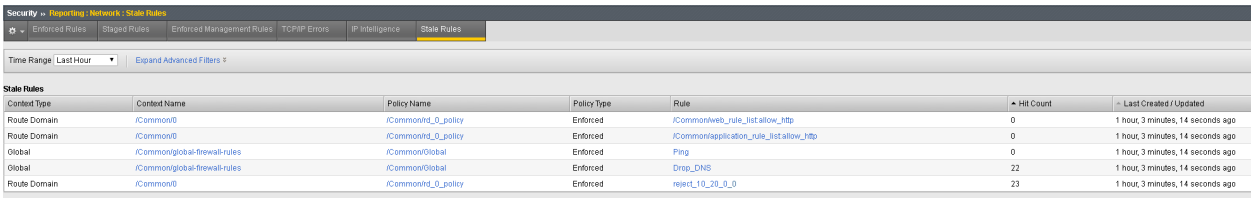
1.3.4 Stale Rule Report

AFM also can list out stale rules within the device its self. You must first enable the feature. To enable, navigate to **Security >Reporting > Settings > Report Settings.** You will then need to check "**Collect Stale Rules Statistics**" found under the Network Firewall Rules Section. Please be sure to click "Save" before proceeding.



Once enabled, navigate to **Security >Reporting > Network > Stale Rules.** Feel free to refresh the web page we've been testing with (http://10.1.20.11) to see data populate into the rules.

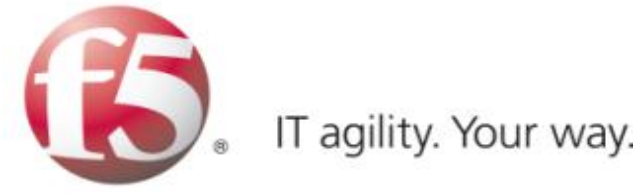
Note: It could take 60+ seconds for data to populate



This information is quite useful for keeping a rule base tidy and optimized.

Anyone can create a firewall rule, but who is the person that removes the unneccesary ones?

****Written for TMOS 15.1.0**



1.4 Lab 3 - AFM DDoS Lab

1.4.1 Lab Overview

During this lab, you will configure the BIG-IP system to detect and report on various network level Denial of Service events. You will then run simulated attacks against the BIG-IP and verify the mitigation, reporting and logging of these attacks.

1.4.2 Detecting and Preventing DNS DoS Attacks on a Virtual Server

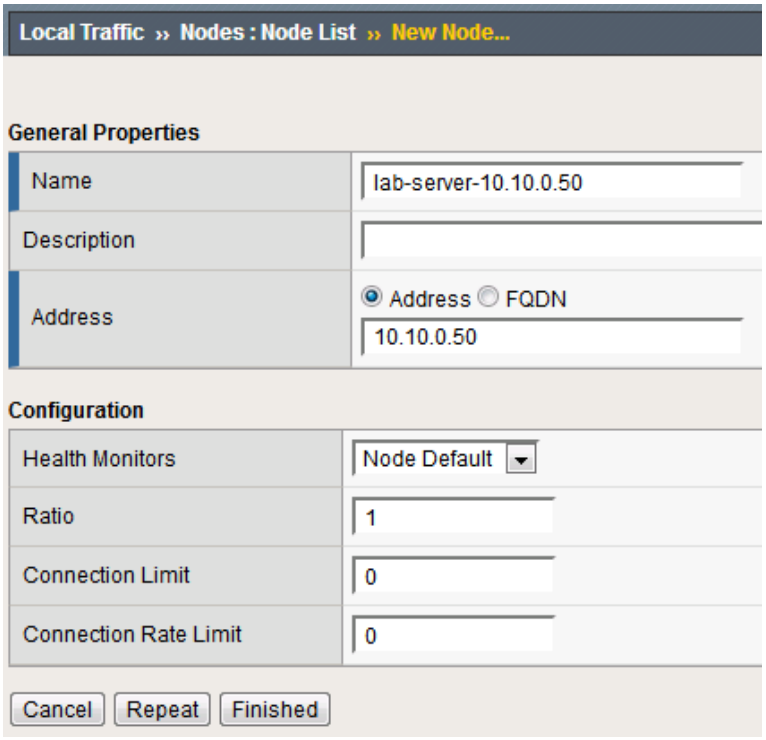
It is day two of your career at Initech, and you are under attack!! You walk into the office on day two only to learn your DNS servers are being attacked by Joanna who took out her flair frustrations on your DNS servers. Before you can protect the servers however, you must first tune and configure them appropriately. (The most challenging part of DoS based protection is tuning correctly).

In this section of the lab, we'll focus on creating DOS profiles that we can assign to virtual servers for protection. Let's get started!

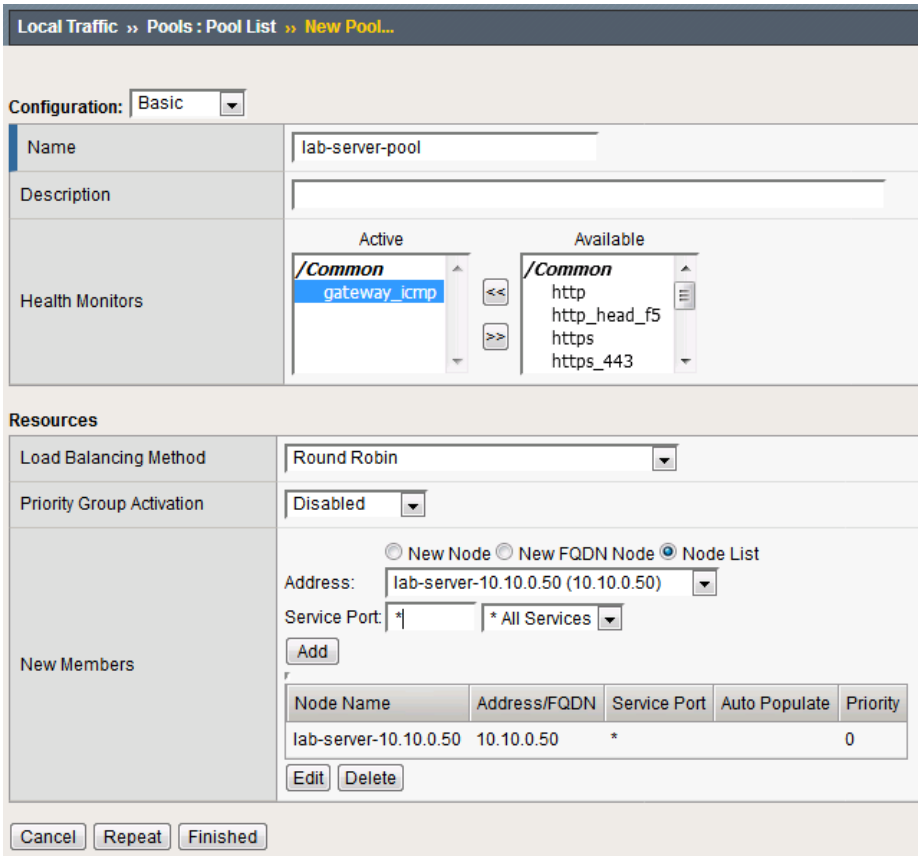
Base BIG-IP Configuration

In this lab, the VE has been configured with the basic system settings and the VLAN/self-IP configurations required for the BIG-IP to communicate and pass traffic on the network. We will now need to configure the BIG-IP to listen for traffic and pass it to the back-end server.

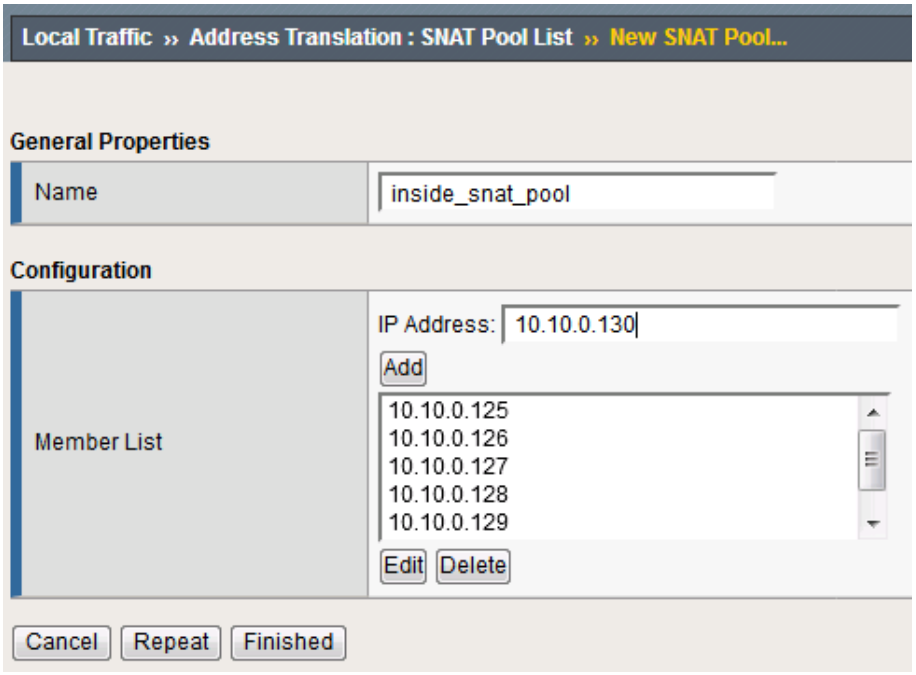
1. Launch the Chrome shortcut titled "BIG-IP UI" on the desktop of your lab jump server. For this lab you will be working on bigip1.dnstest.lab (http://192.168.1.100). The credentials for the BIG-IP are conveniently displayed in the login banner. Just in case: **admin** / **401elliottW!**
2. Navigate to **Local Traffic** > **Nodes** and create a new node with the following settings, leaving unspecified fields at their default value:
    - Name: lab-server-10.10.0.50
    - Address: 10.10.0.50



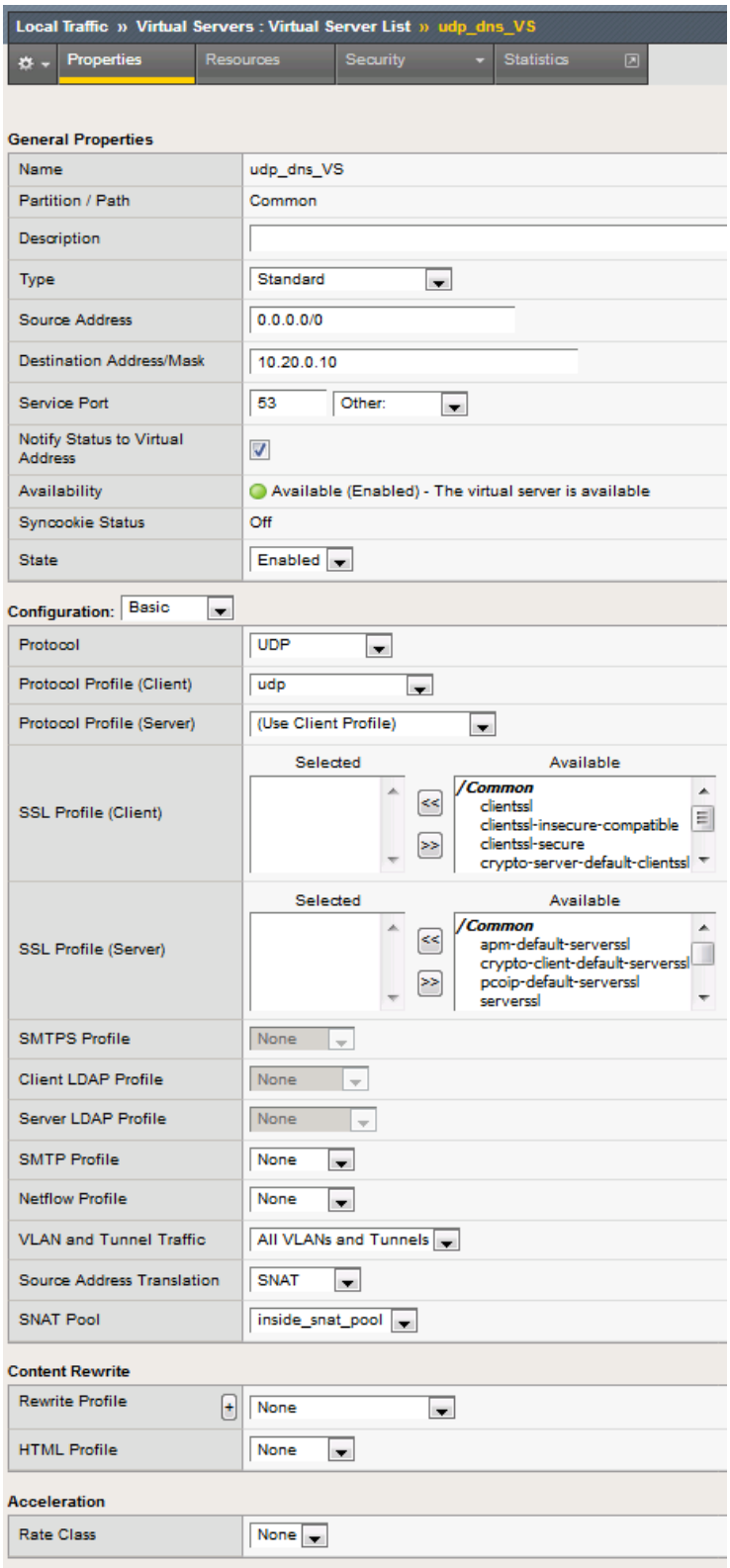
3. Click **Finished** to add the new node.
4. Navigate to **Local Traffic** > **Pools** and create a new pool with the following settings, leaving unspecified attributes at their default value:
   - Name: lab-server-pool
   - Health Monitors: gateway_icmp
   - **New Members: Node List**
     - Address: lab-server-10.10.0.50
     - Service Port: * (All Services)
     - Click **Add** to add the new member to the member list.



5. Click **Finished** to create the new pool.
6. Because the attack server will be sending a huge amount of traffic, we'll need a large SNAT pool. Navigate to **Local Traffic** > **Address Translation** > **SNAT Pool List** and create a new SNAT pool with the following attributes:
   - Name: inside_snat_pool
   - Member List (click Add after each IP):
     10.10.0.125, 10.10.0.126, 10.10.0.127, 10.10.0.128, 10.10.0.129, 10.10.0.130
   - Click Finished



7. Navigate to **Local Traffic** > **Virtual Servers** and create a new virtual server with the following settings, leaving unspecified fields at their default value:
   - Name: udp_dns_VS
   - Destination Address/Mask: 10.20.0.10
   - Service Port: 53 (other)
   - Protocol: UDP
   - Source Address Translation: SNAT
   - SNAT Pool: inside_snat_pool
   - Default Pool: lab-server-pool
8. Click Finished



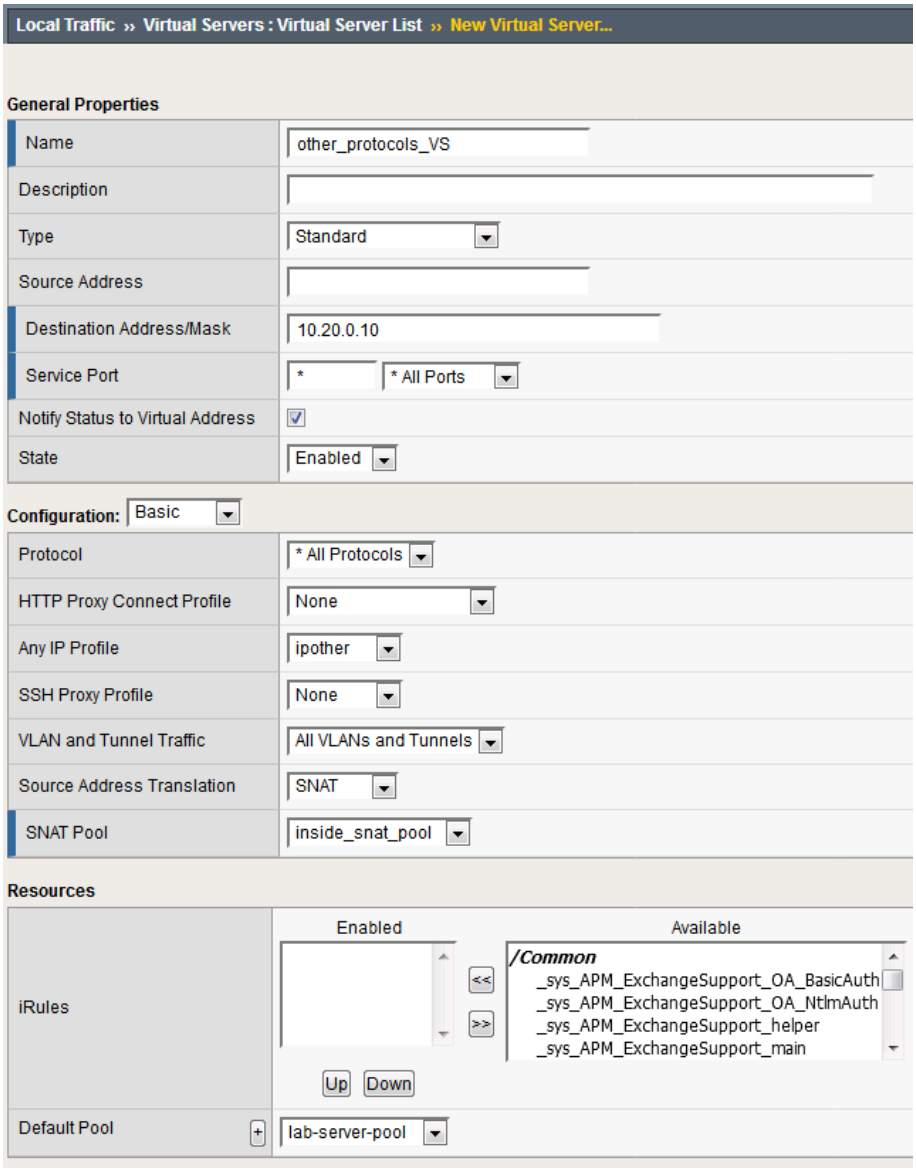
9. We'll now test the new DNS virtual server. SSH into the attack host by clicking the "Attack Host (Ubuntu)" icon on the jump host desktop.
10. Issue the `dig @10.20.0.10 www.example.com +short` command on the BASH CLI of the

attack host. You should see output similar to:

```
ubuntu@dnsclient:~$ dig @10.20.0.10 www.example.com +short
10.10.0.99
```

This verifies that DNS traffic is passing through the BIG-IP.

11. Return to the BIG-IP and navigate to **Local Traffic** > **Virtual Servers** and create a new virtual server with the following settings, leaving unspecified fields at their default value:
    - Name: other_protocols_VS
    - Destination Address/Mask: 10.20.0.10
    - Service Port: * (All Ports)
    - Protocol: * All Protocols
    - Any IP Profile: ipother
    - Source Address Translation: SNAT
    - SNAT Pool: inside_snat_pool
    - Default Pool: lab-server-pool
12. Click Finished

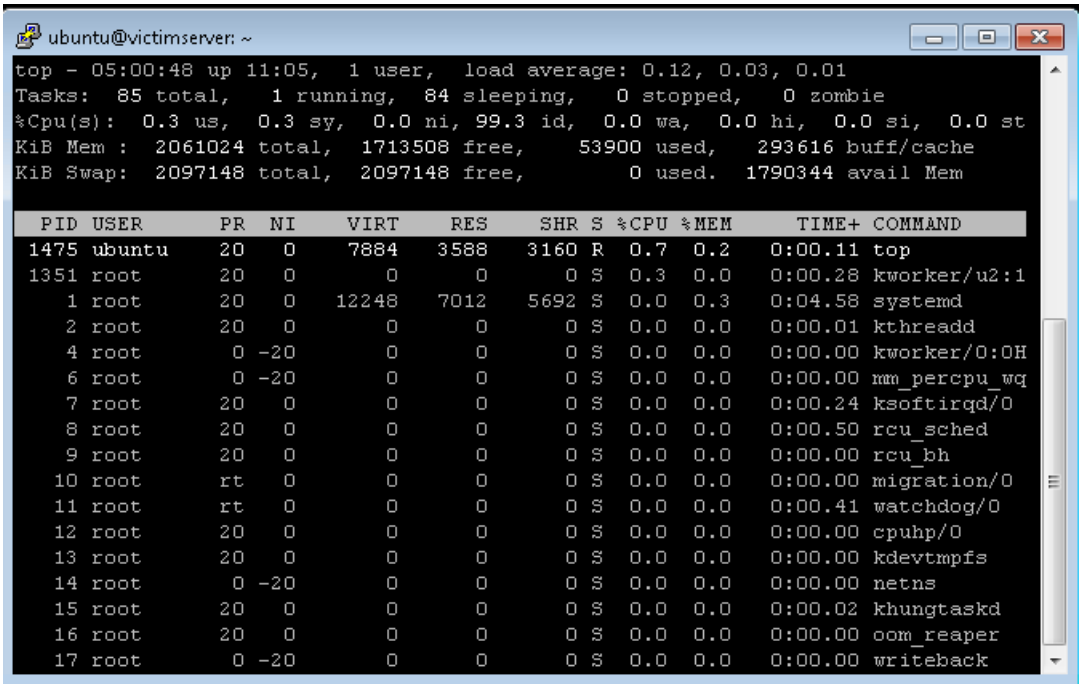


13. Return to the Attack Host SSH session and attempt to SSH to the server using SSH 10.20.0.10. Simply verify that you are prompted for credentials and press CTRL+C to cancel the session. This verifies that non-DNS traffic is now flowing through the BIG-IP.

Establishing a DNS server baseline

Before we can prevent Joanna from attacking our DNS server, again, we should establish a baseline for how many QPS our DNS server can handle. For this lab, let's find the magic number of QPS that causes 50% CPU utilization on the BIND process.

1. Connect to the Victim Server SSH session by double-clicking the **Victim Server (Ubuntu)** shortcut on the jump host desktop.
2. From the BASH prompt, enter **top** and press **Enter** to start the top utility.
3. You will see a list of running processes sorted by CPU utilization, like the output below:

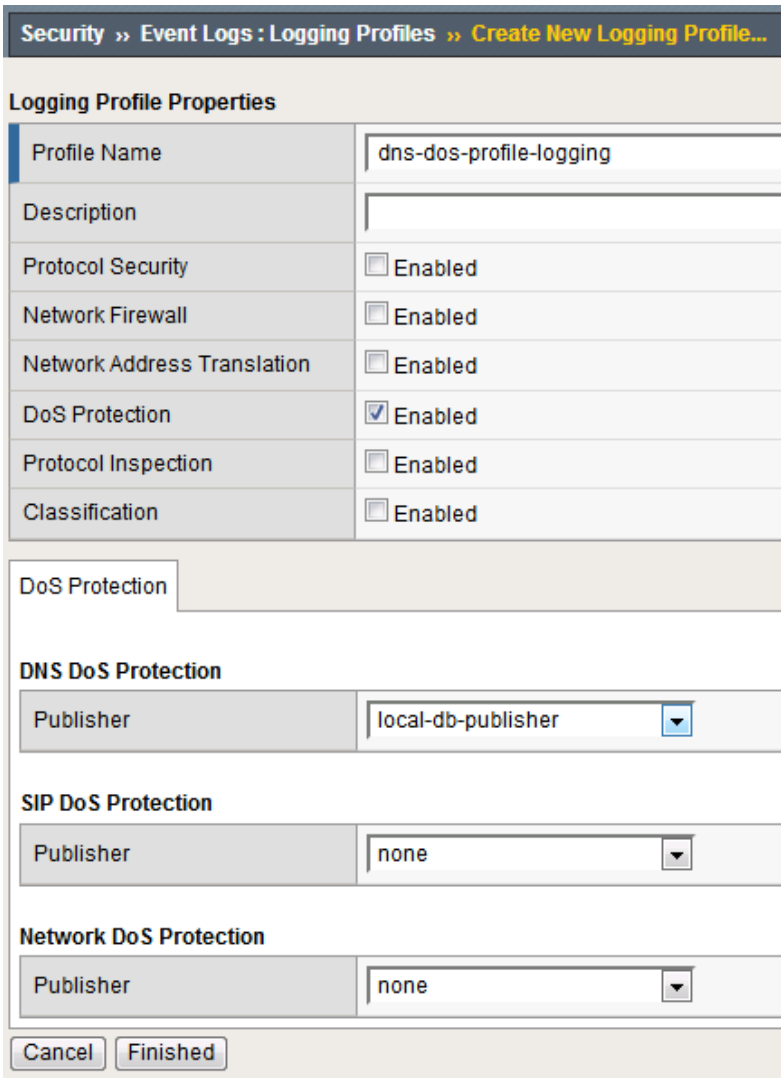


4. Connect to the Attack Host SSH session by double-clicking the **Attack Host (Ubuntu)** shortcut on the jump host desktop.
5. Start by sending 500 DNS QPS for 30 seconds to the host using the following syntax:
   `dnsperf -s 10.20.0.10 -d queryfile-example-current -c 20 -T 20 -l 30 -q 10000 -Q 500`
6. Observe CPU utilization over the 30 second window for the **named** process. If the CPU utilization is below 45%, increase the QPS by increasing the -Q value. If the CPU utilization is above 55%, decrease the QPS. This
7. Record the QPS required to achieve a sustained CPU utilization of approximately 50%. Consider this the QPS that the server can safely sustain for demonstration purposes.
8. Now, attack the DNS server with 10,000 QPS using the following syntax:
   `dnsperf -s 10.20.0.10 -d queryfile-example-current -c 20 -T 20 -l 30 -q 10000 -Q 10000`
9. You'll notice that the CPU utilization on the victim server skyrockets, as well as DNS query timeout errors appearing on the attack server's SSH session. This shows your DNS server is overwhelmed.

Configuring a DoS Logging Profile

We'll create a DoS logging profile so that we can see event logs in the BIG-IP UI during attack mitigation.

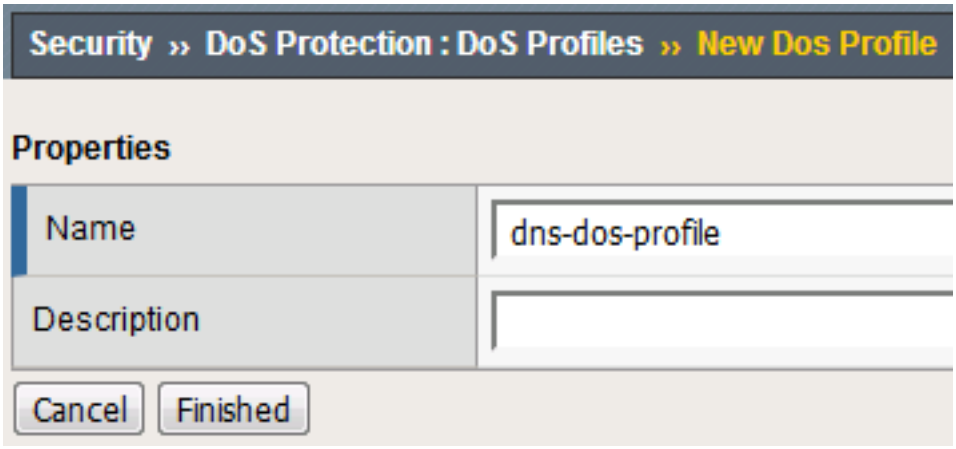
1. On the BIG-IP web UI, navigate to **Security** > **Event Logs** > **Logging Profiles** and create a new profile with the following values, leaving unspecified attributes at their default value:
   - Profile Name: dns-dos-profile-logging
   - DoS Protection: Enabled
   - DNS DoS Protection Publisher: local-db-publisher and click **Finish.**



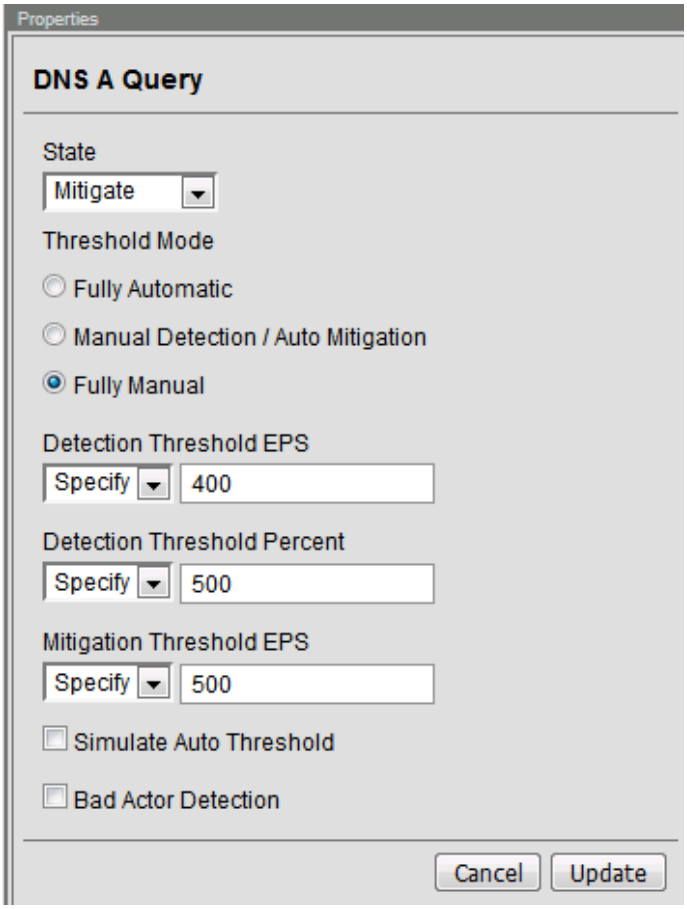
Configuring a DoS Profile

We will now create a DoS profile with manually configured thresholds to limit the attack's effect on our server.

1. Navigate to **Security** > **DoS Protection** > **DoS Profiles**
2. Create a new DoS profile with the name **dns-dos-profile**.
3. Click **Finished**.



4. The UI will return to the DoS Profiles list. Click the **dns-dos-profile** name.
5. Click the **Protocol Security** tab and select **DNS Security** from the drop-down.
6. Click the **DNS A Query** vector from the Attack Type list.
7. Modify the **DNS A Query** vector configuration to match the following values, leaving unspecified attributes with their default value:
   - State: Mitigate
   - Threshold Mode: Fully Manual
   - Detection Threshold EPS: (Set this at 80% of your safe QPS value)
   - Mitigation Threshold EPS: (Set this to your safe QPS value)



8. Make sure that you click **Update** to save your changes.

Attaching a DoS Profile

We will attach the DoS profile to the virtual server that we configured to manage DNS traffic.

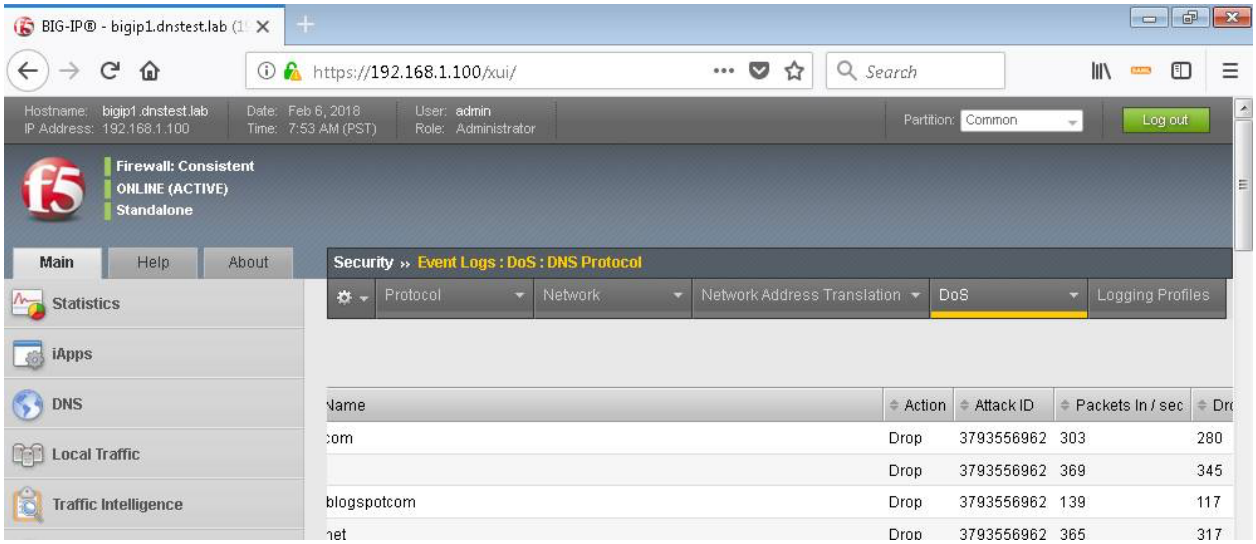
1. Navigate to **Local Traffic** > **Virtual Servers** > **Virtual Server List**.
2. Click on the **udp_dns_VS** name.
3. Click on the **Security** tab and select **Policies**.
4. In the **DoS Protection Profile** field, select **Enabled** and choose the **dns-dos-profile**.
5. In the **Log Profile**, select **Enabled** and move the **dns-dos-profile-logging** profile from **Available** to **Selected**.
6. Click **Update**.

Simulate a DNS DDoS Attack

1. Open the SSH session to the victim server and ensure the top utility is running.
2. Once again, attack your DNS server from the attack host using the following syntax:

```
dnsperf -s 10.20.0.10 -d queryfile-example-current -c 20 -T 20 -l 30 -q
10000 -Q 10000
```

3. On the server SSH session running the top utility, notice the CPU utilization on your server remains in a range that ensures the DNS server is not overwhelmed.
4. After the attack, navigate to **Security** > **Event Logs** > **DoS** > **DNS Protocol**. Observe the logs to see the mitigation actions taken by the BIG-IP. Be sure to scroll right. . .



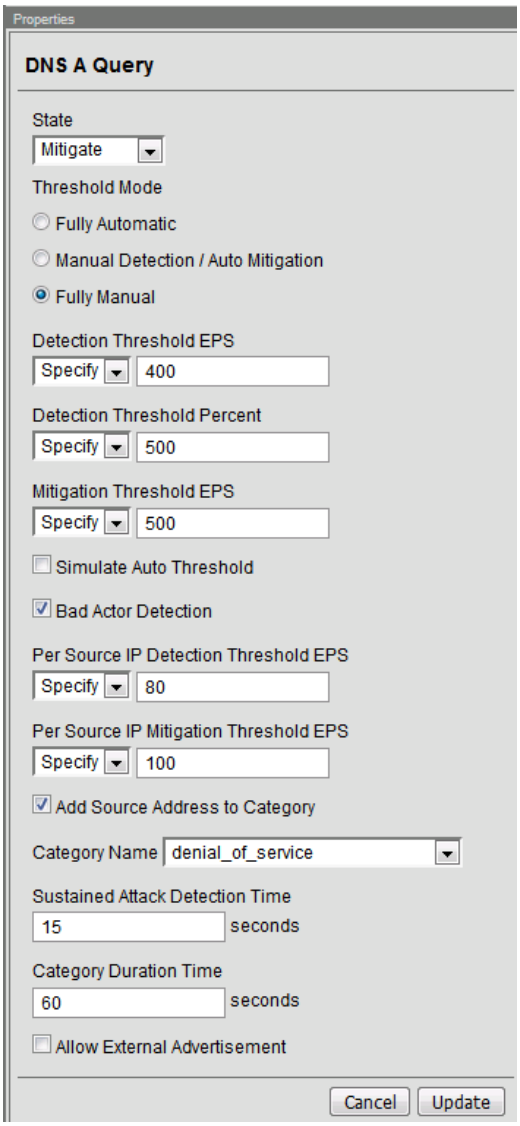
DNS DDoS Mitigations for Continued Service

At this point, you have successfully configured the BIG-IP to limit the amount of resource utilization on the BIG-IP, thus further frustrating Joanna on her flair rage. Unfortunately, even valid DNS requests can be caught in the mitigation we've configured. There are further steps that can be taken to mitigate Joanna's attack that will allow non-malicious DNS queries.

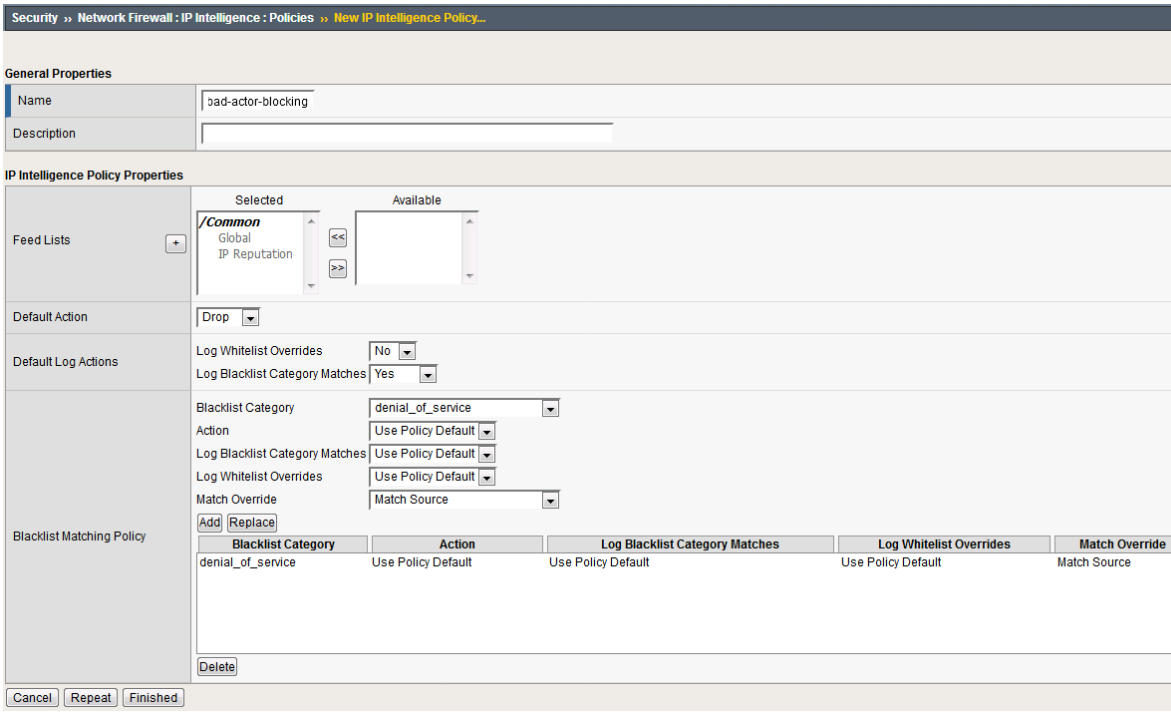
Bad Actor Detection

Bad actor detection and blacklisting allows us to completely block communications from malicious hosts at the BIG-IP, completely preventing those hosts from reaching the back-end servers. To demonstrate:

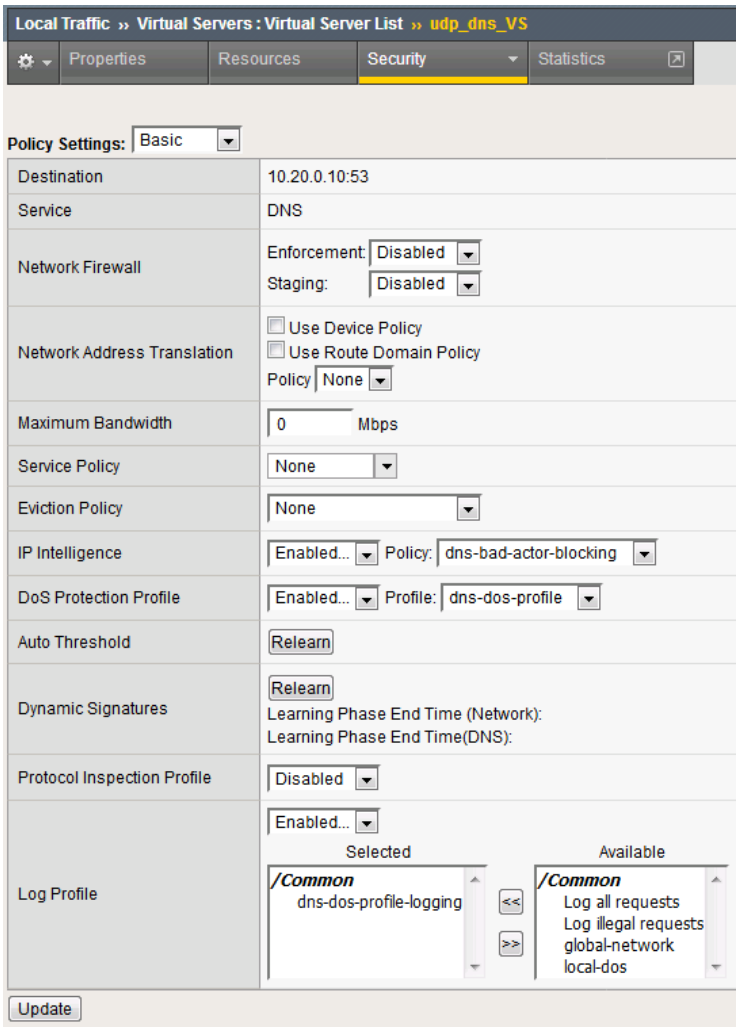
1. Navigate to **Security** > **DoS Protection** > **DoS Profiles**.
2. Click on the **dns-dos-profile** profile name.
3. Click on the **Protocol Security** tab then select **DNS Security**.
4. Click on the **DNS A Query** attack type name.
5. Modify the vector as follows:
    - Bad Actor Detection: Checked
    - Per Source IP Detection Threshold EPS: 80
    - Per Source IP Mitigation Threshold EPS: 100
    - Add Source Address to Category: Checked
    - Category Name: denial_of_service
    - Sustained Attack Detection Time: 15 seconds
    - Category Duration Time: 60 seconds



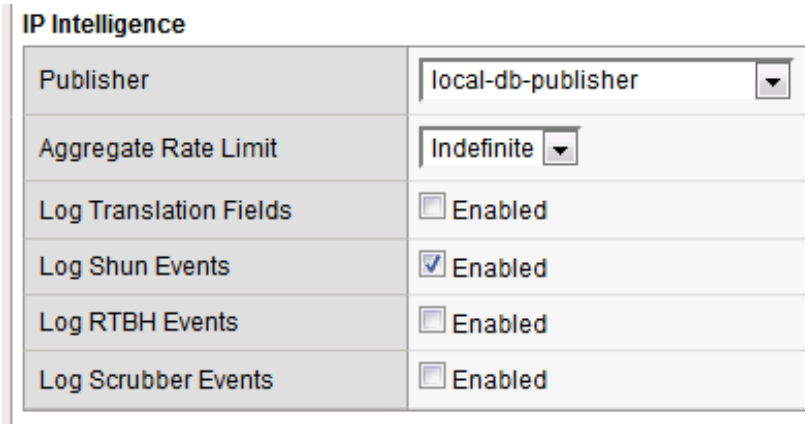
6. Make sure you click **Update** to save your changes.
7. Navigate to **Security** > **Network Firewall** > **IP Intelligence** > **Policies** and create a new IP Intelligence policy with the following values, leaving unspecified attributes at their default values:
    - Name: dns-bad-actor-blocking
    - **Default Log Actions section:**
        - Log Blacklist Category Matches: Yes
    - **Blacklist Matching Policy**
        - **Create a new blacklist matching policy:**
            - Blacklist Category: denial_of_service
            - Click **Add** to add the policy then click finished



8. Navigate to **Local Traffic** > **Virtual Servers** > **Virtual Server List**.
9. Click on the **udp_dns_VS** virtual server name.
10. Click on the **Security** tab and select **Policies**.
11. Enable **IP Intelligence** and choose the **dns-bad-actor-blocking** policy.

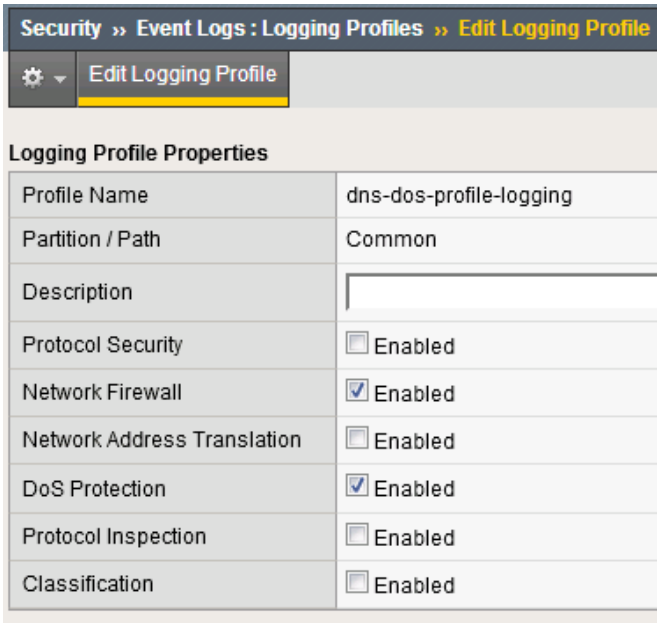


12. Make sure you click **Update** to save your changes.
13. Navigate to **Security** > **Event Logs** > **Logging Profiles**.
14. Click the **global-network** logging profile name.
15. Under the **Network Firewall** tab (next to Protocol Security), set the IP Intelligence Publisher to **local-db-publisher** and check **Log Shun Events**.

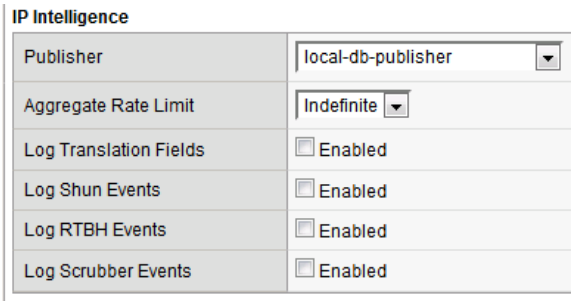


16. Click **Update** to save your changes.

17. Click the **dns-dos-profile-logging** logging profile name.
18. Check **Enabled** next to **Network Firewall**.



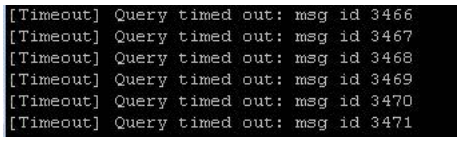
19. Under the **Network Firewall** tab, change the **IP Intelligence Publisher** to **local-db-publisher** and click **Update**.



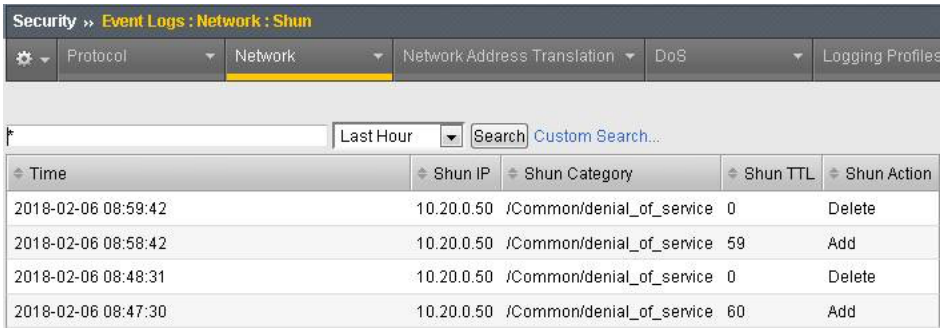
20. Bring into view the Victim Server SSH session running the top utility to monitor CPU utilization.
21. On the Attack Server host, launch the DNS attack once again using the following syntax:

```
dnsperf -s 10.20.0.10 -d queryfile-example-current -c 20 -T 20 -l 30 -q
10000 -Q 10000
```

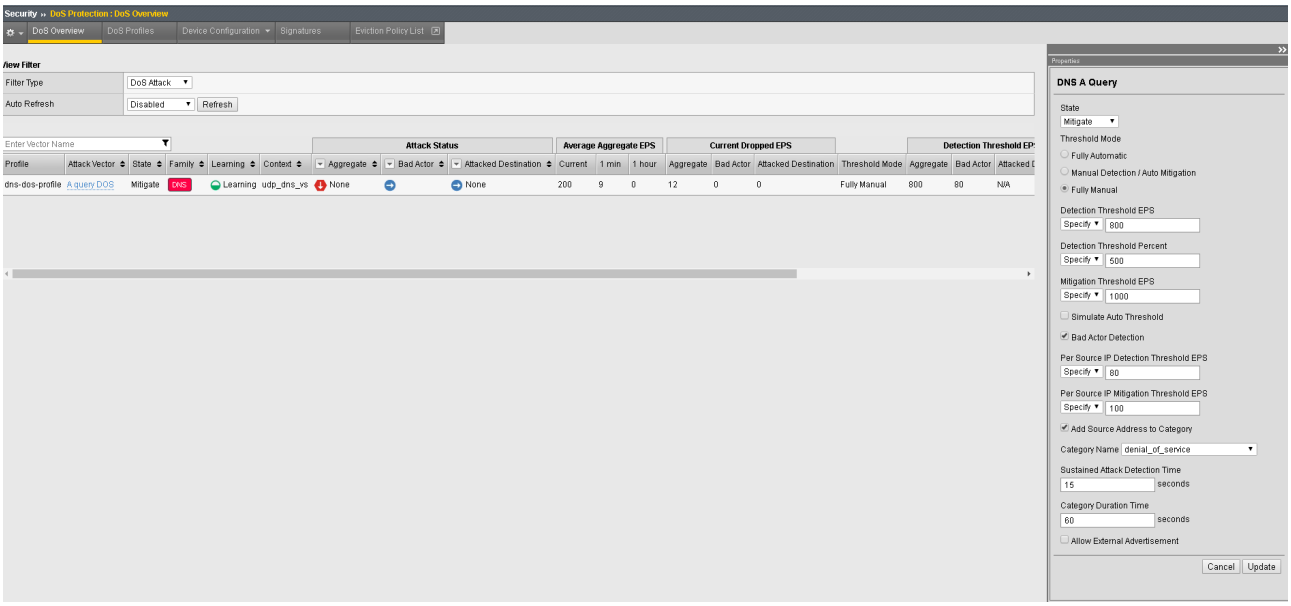
22. You'll notice CPU utilization on the BIG-IP begin to climb, but slowly drop. The attack host will show that queries are timing out as shown below. This is due to the BIG-IP blacklisting the bad actor.



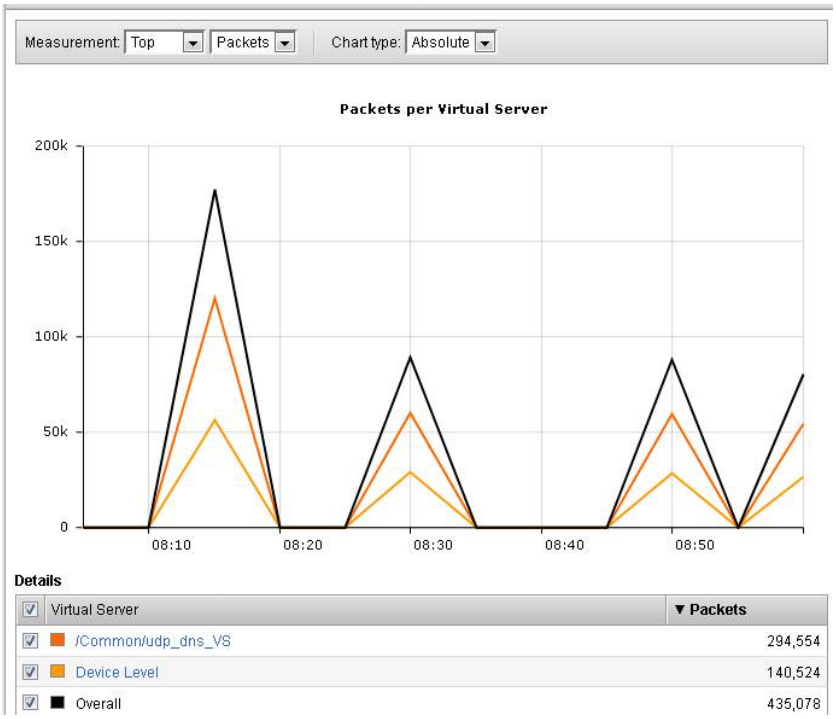
23. Navigate to **Security** > **Event Logs** > **Network** > **IP Intelligence**. Observe the bad actor blocking mitigation logs.
24. Navigate to **Security** > **Event Logs** > **Network** > **Shun**. This screen shows the bad actor being added to (and later deleted from) the shun category.



25. While the attack is running, navigate to **Security** > **DoS Protection**> **DoS Overview (you may need to refresh or set the auto refresh to 10 seconds).** You will notice from here you can see all the details of the active attacks. You can also modify an attack vector right from this screen by clicking on the attack vector and modifying the fly out.

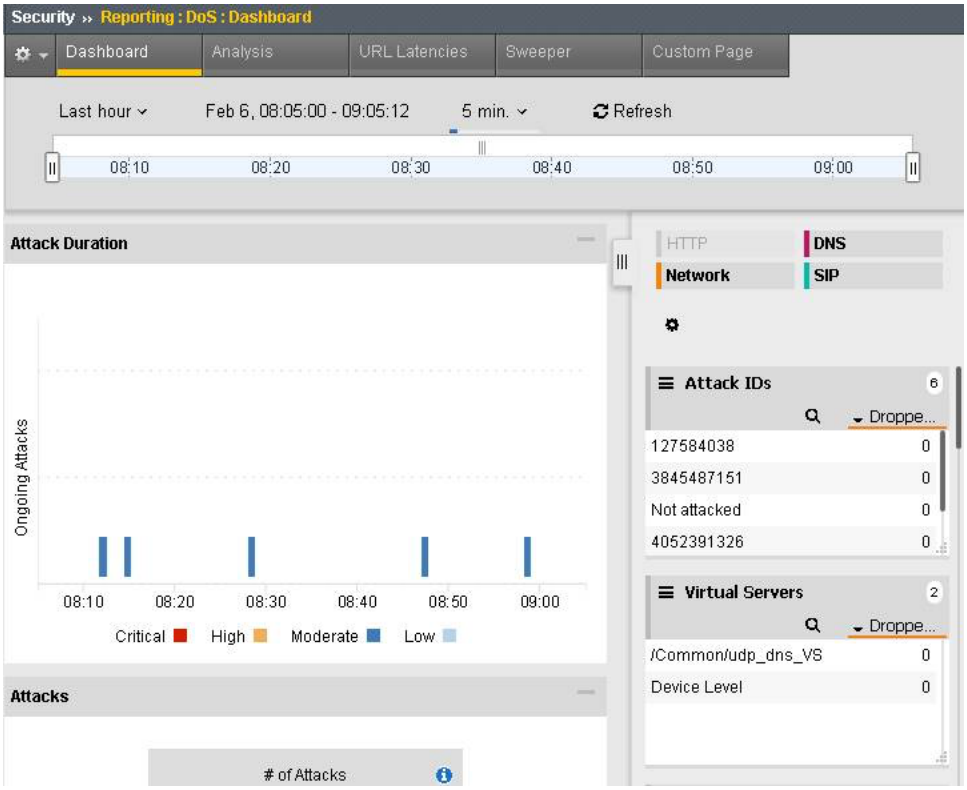


26. Navigate to **Security** > **Reporting** > **Protocol** > **DNS**. Change the **View By** drop-down to view various statistics around the DNS traffic and attacks.

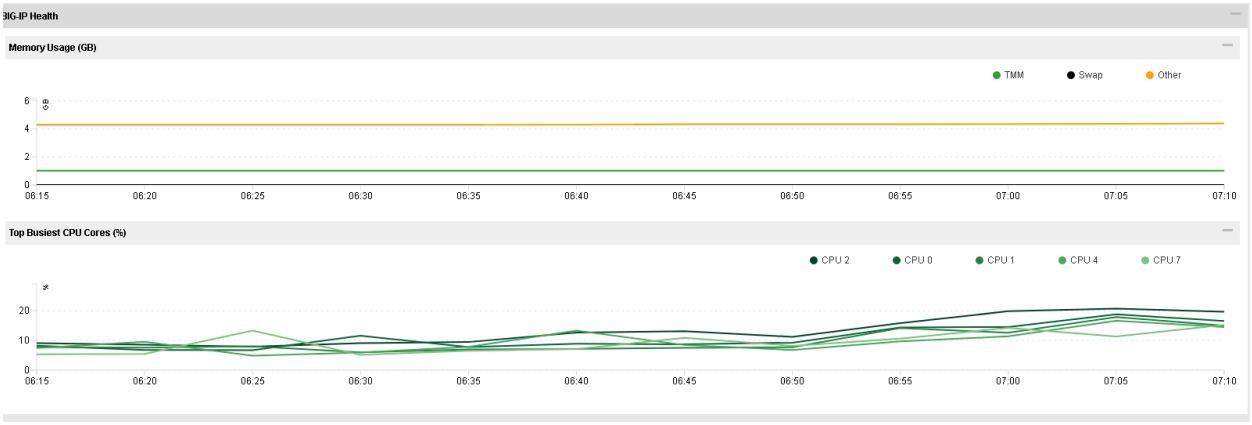


27. Navigate to **Security** > **Reporting** > **Network** > **IP Intelligence**. The default view may be blank. Change the **View By** drop-down to view various statistics around the IP Intelligence handling of the attack traffic.

28. Navigate to **Security** > **Reporting** > **DoS** > **Dashboard** to view an overview of the DoS attacks and timeline. You can select filters in the filter pane to highlight specific attacks.



29. Finally, navigate to **Security** > **Reporting** > **DoS** > **Analysis**. View detailed statistics around each attack.



Remote Triggered Black Holing

The BIG-IP supports the advertisement of bad actor(s) to upstream devices via BGP to block malicious traffic closer to the source. This is accomplished by publishing a blacklist to an external resource. This is not demonstrated in this lab.

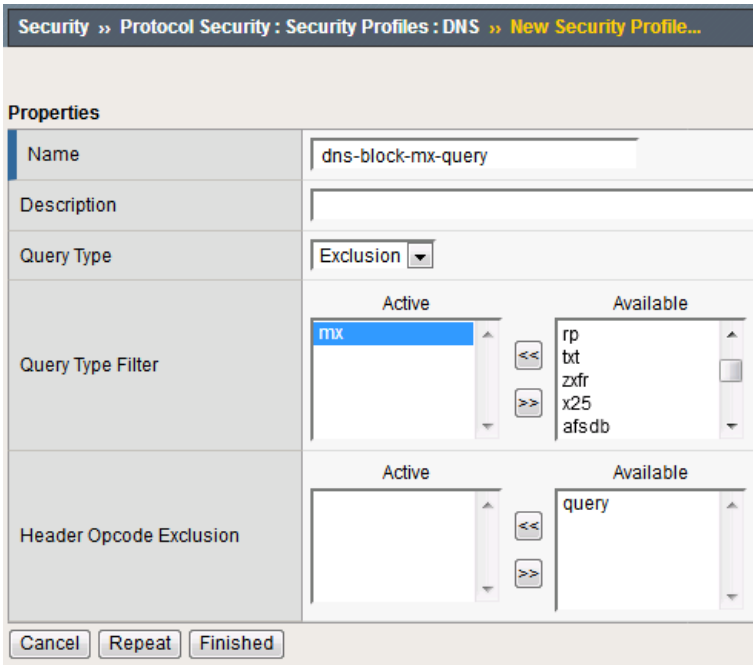
Silverline Mitigation

F5's Silverline service offers "always on" and "on demand" DDoS scrubbing that could assist in this scenario as well. This is not demonstrated in this lab.

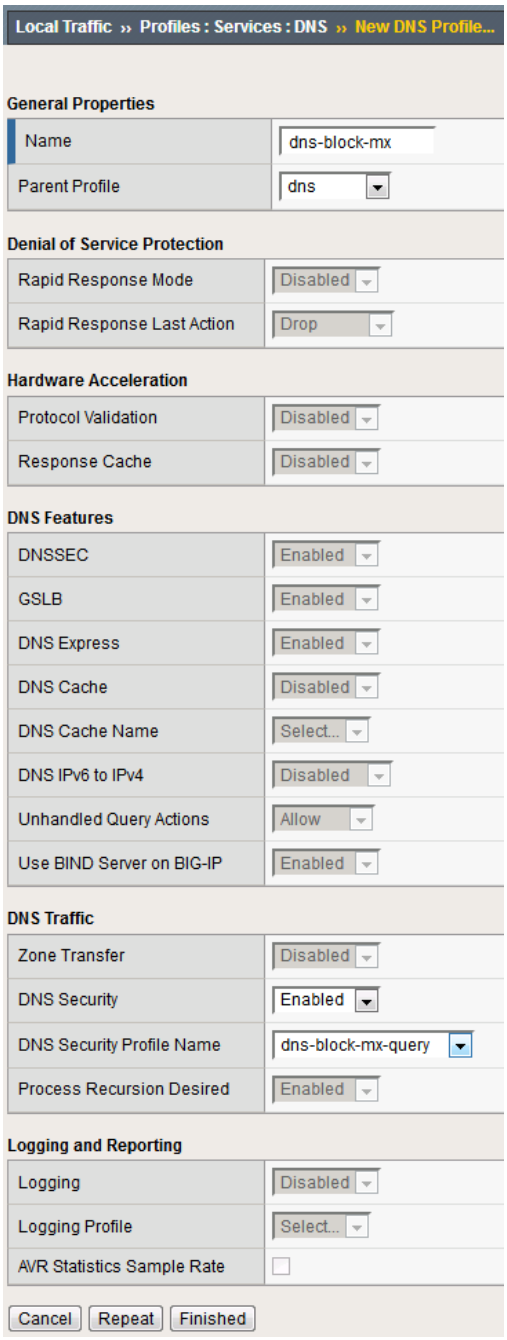
Filtering specific DNS operations

The BIG-IP offers the ability to filter DNS query types and header opcodes to act as a DNS firewall. To demonstrate, we will block MX queries from our DNS server.

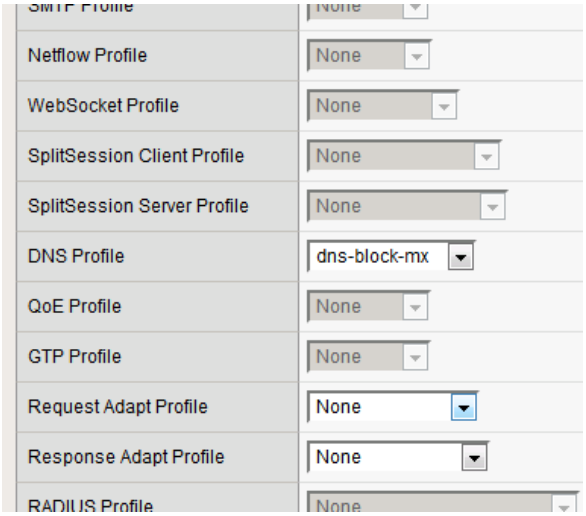
1. Open the SSH session to the Attack Host.
2. Perform an MX record lookup by issuing the following command:
   `dig @10.20.0.10 MX example.com`
3. The server doesn't have a record for this domain. This server doesn't have MX records, so those requests should be filtered
4. Navigate to **Security** > **Protocol Security** > **Security Profiles** > **DNS** and create a new DNS security profile with the following values, leaving unspecified attributes at their default value:
   - Name: dns-block-mx-query
   - Query Type Filter: move mx from Available to Active and click finished



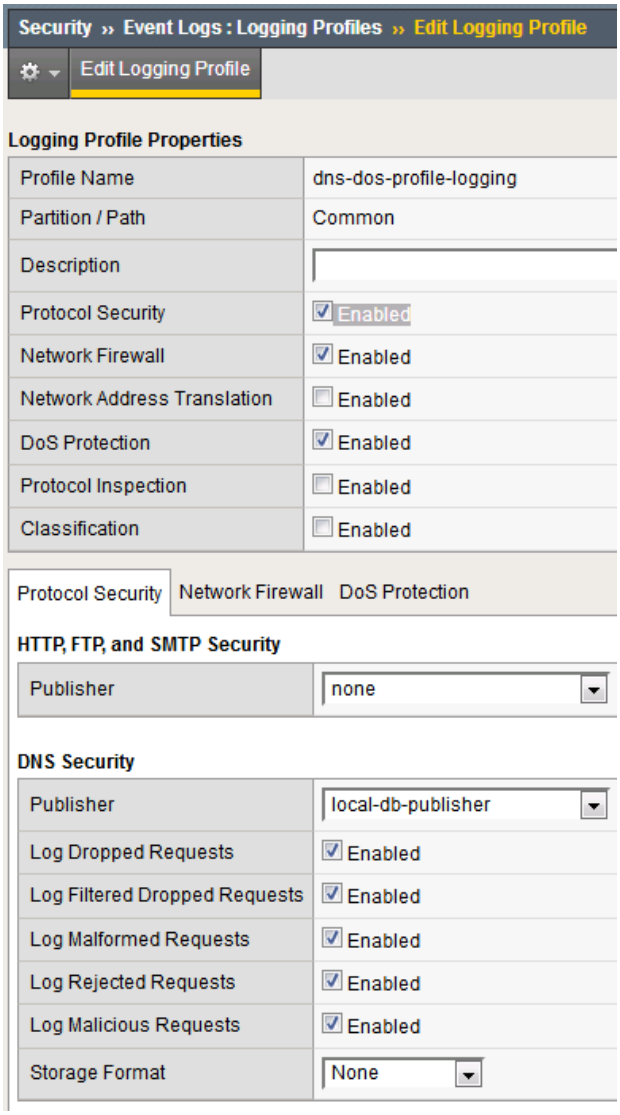
5. Navigate to **Local Traffic** > **Profiles** > **Services** > **DNS**. **NOTE:** if you are mousing over the services, DNS may not show up on the list. Select **Services** and then use the pulldown menu on services to select **DNS**.
6. Create a new DNS services profile with the following values, leaving unspecified values at their default values:
   - Name: dns-block-mx
   - **DNS Traffic**
     - DNS Security: Enabled
     - DNS Security Profile Name: dns-block-mx-query. Click finished



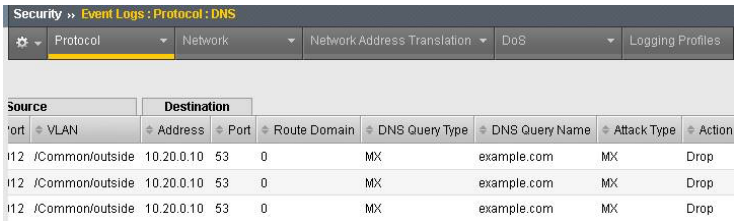
7. Navigate to **Local Traffic** > **Virtual Servers** > **Virtual Server List**.
8. Click on the **udp_dns_VS** virtual server name.
9. In the **Configuration** section, change the view to **Advanced**.
10. Set the **DNS Profile** to **dns-block-mx**.



11. Click **Update** to save your settings.
12. Navigate to **Security** > **Event Logs** > **Logging Profiles**.
13. Click on the **dns-dos-profile-logging** logging profile name.
14. Check **Enabled** next to **Protocol Security**.
15. In the **Protocol Security** tab, set the **DNS Security Publisher** to **local-db-publisher** and check all five of the request log types.



16. Make sure that you click **Update** to save your settings.
17. Return to the Attack Server SSH session and re-issue the MX query command:
    `dig @10.20.0.10 MX example.com`
18. The query hangs as the BIG-IP is blocking the MX lookup.
19. Navigate to **Security** > **Event Logs** > **Protocol** > **DNS**. Observe the MX query drops.



This concludes the DNS portion of the lab. On the Victim Server, stop the top utility by pressing **CTRL + C**. No mail for you Joanna!!

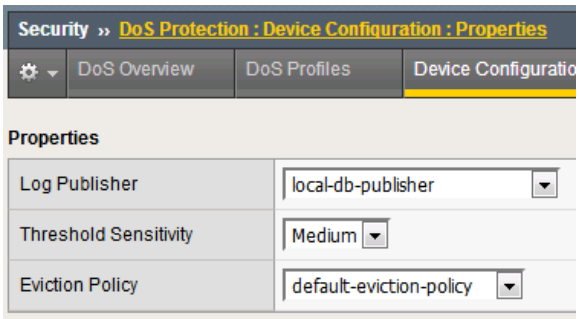
1.4.3 Advanced Firewall Manager (AFM) Detecting and Preventing System DoS and DDoS Attacks

In this part of the lab, you'll focus on creating system-wide policies that mitigate attacks across the entire BIG-IP instance.

Configure Logging

Configuring a logging destination will allow you to verify the BIG-IPs detection and mitigation of attacks, in addition to the built-in reporting.

1. In the BIG-IP web UI, navigate to **Security** > **DoS Protection** > **Device Configuration** > **Properties**.
2. Under **Log Pubisher**, select **local-db-publisher**.
3. Click the **Commit Changes to System** button.

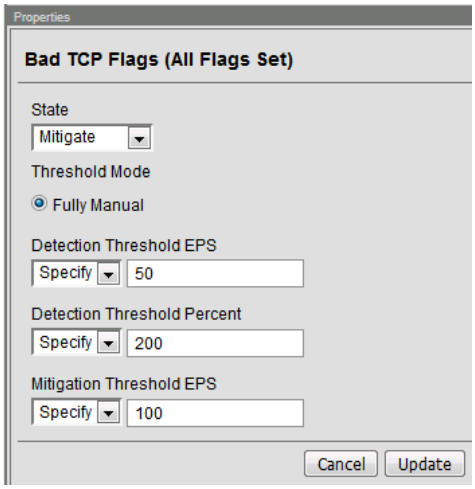


Simulating a Christmas Tree Packet Attack

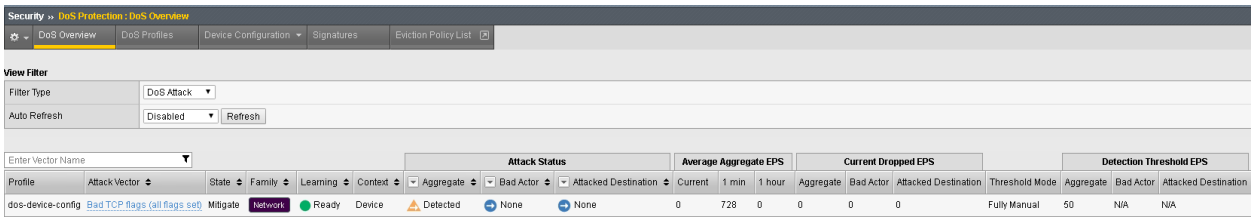
Joanna was feeling festive this morning. In this example, we'll set the BIG-IP to detect and mitigate Joanna's attack where all flags on a TCP packet are set. This is commonly referred to as a Christmas Tree Packet and is intended to increase processing on in-path network devices and end hosts to the target.

We'll use the hping utility to send 25,000 packets to our server, with random source IPs to simulate a DDoS attack where multiple hosts are attacking our server. We'll set the SYN, ACK, FIN, RST, URG, PUSH, Xmas and Ymas TCP flags.

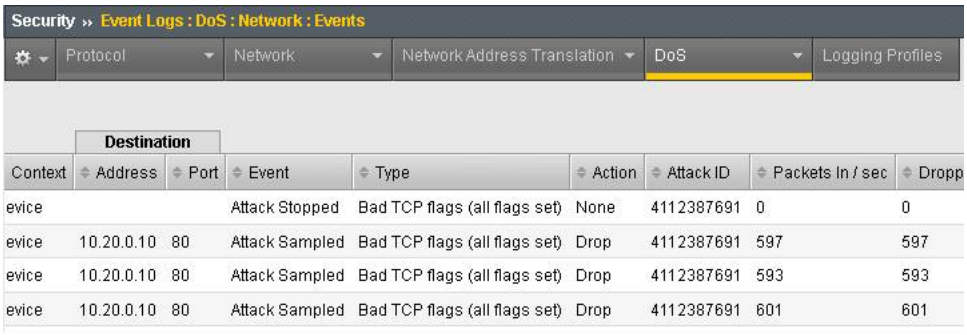
1. In the BIG-IP web UI, navigate to **Security** > **DoS Protection** > **Device Configuration** > **Network Security**.
2. Expand the **Bad-Header-TCP** category in the vectors list.
3. Click on the **Bad TCP Flags (All Flags Set)** vector name.
4. Configure the vector with the following parameters:
    - State: Mitigate
    - Threshold Mode: Fully Manual
    - Detection Threshold EPS: Specify 50
    - Detection Threshold Percent: Specify 200
    - Mitigation Threshold EPS: Specify 100



5. Click **Update** to save your changes.
6. Open the BIG-IP SSH session and scroll the ltm log in real time with the following command: tail -f /var/log/ltm
7. On the attack host, launch the attack by issuing the following command on the BASH prompt:
   ```
   sudo hping3 10.20.0.10 --flood --rand-source --destport 80 -c 25000 --syn --ack --fin --rst --push --urg --xmas --ymas
   ```
8. You'll see the BIG-IP ltm log show that the attack has been detected:
   ```
   Feb  6 09:36:09 bigip1 err tmm[10663]: 01010252:3: A Enforced Device DOS attack start was detected for vector Bad TCP flags (all flags set), Attack ID 411238769
   1.
   ```
9. After approximately 60 seconds, press **CTRL+C** to stop the attack.
   ```
   ubuntu@attackhost:~$ sudo hping3 10.20.0.10 --flood --rand-source --destport 80
   -c 25000 --syn --ack --fin --rst --push --urg --xmas --ymas
   HPING 10.20.0.10 (ens3 10.20.0.10): RSAFPUXY set, 40 headers + 0 data bytes
   hping in flood mode, no replies will be shown
   ^C
   --- 10.20.0.10 hping statistic ---
   361447 packets transmitted, 0 packets received, 100% packet loss
   round-trip min/avg/max = 0.0/0.0/0.0 ms
   ubuntu@attackhost:~$
   ```
10. Navigate to **Security** > **DoS Protection**> **DoS Overview (you may need to refresh or set the auto refresh to 10 seconds).** You'll notice from here you can see all the details of the active attacks. You can also modify an attack vector right from this screen by clicking on the attack vector and modifying the details in the fly out panel.



11. Return to the BIG-IP web UI. Navigate to **Security** > **Event Logs** > **DoS** > **Network** > **Events**. Observe the log entries showing the details surrounding the attack detection and mitigation.

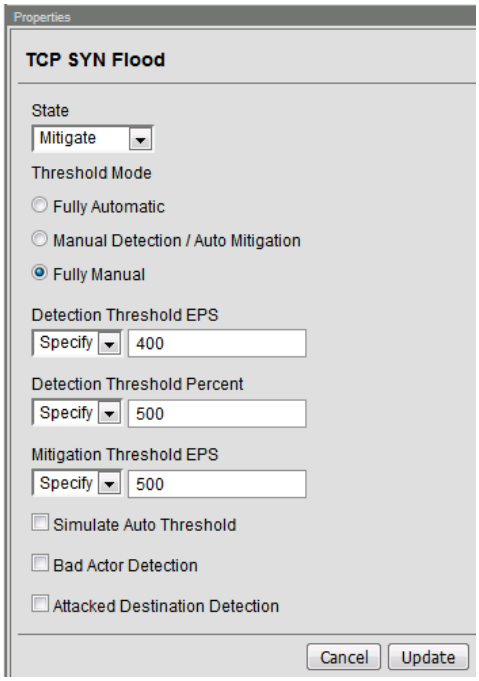


12. Navigate to **Security** > **Reporting** > **DoS** > **Analysis**. Single-click on the attack ID in the filter list to the right of the charts and observe the various statistics around the attack.

Simulating a TCP SYN DDoS Attack

In the last example, Joanna crafted a packet that is easily identified as malicious, as its invalid. We'll now simulate an attack with traffic that could be normal, acceptable traffic. The TCP SYN flood attack will attempt to DDoS a host by sending valid TCP traffic to a host from multiple source hosts.

1. In the BIG-IP web UI, go to **Security** > **DoS Protection** > **Device Configuration** > **Network Security**.
2. Expand the **Flood** category in the vectors list.
3. Click on **TCP Syn Flood** vector name.
4. Configure the vector with the following parameters:
    - State: Mitigate
    - Threshold Mode: Fully Manual
    - Detection Threshold EPS: 200
    - Detection Threshold Percent: 500
    - Mitigation Threshold EPS: 400



5. Click **Update** to save your changes.
6. Open the BIG-IP SSH session and scroll the ltm log in real time with the following command: `tail -f /var/log/ltm`
7. On the attack host, launch the attack by issuing the following command on the BASH prompt:
   `sudo hping3 10.20.0.10 --flood --rand-source --destport 80 --syn -d 120 -w 64`
8. After about 60 seconds, stop the flood attack by pressing **CTRL + C**.
9. Return to the BIG-IP web UI and navigate to **Security** > **Event Logs** > **DoS** > **Network** > **Events**. Observe the log entries showing the details surrounding the attack detection and mitigation.
10. Navigate to **Security** > **Reporting** > **DoS** > **Dashboard** to view an overview of the DoS attacks and timeline. You can select filters in the filter pane to highlight the specific attack.
11. Finally, navigate to **Security** > **Reporting** > **DoS** > **Analysis**. View detailed statistics around the attack.

Preventing Global DoS Sweep and Flood Attacks

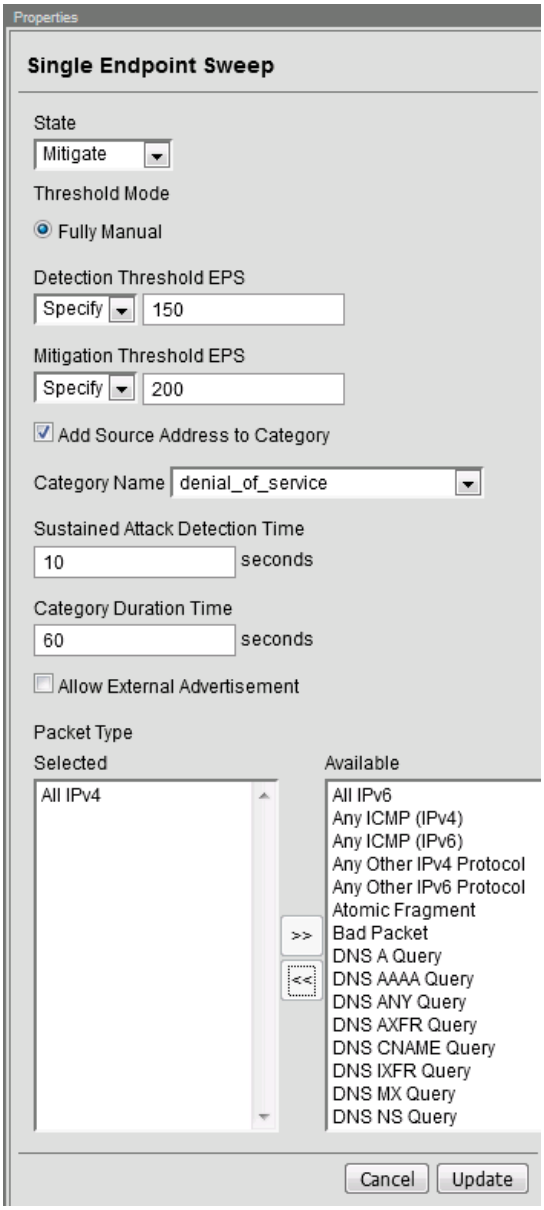
In the last section, the focus was on attacks originating from various hosts. In this section, we will focus on mitigating flood and sweep attacks from a single host.

Single Endpoint Sweep

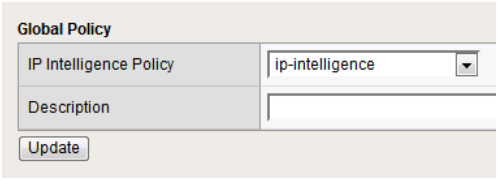
The single endpoint sweep is an attempt for an attacker to send traffic across a range of ports on the target server, typically to scan for open ports.

1. In the BIG-IP web UI, navigate to **Security** > **DoS Protection** > **Device Configuration** > **Network Security**.
2. Expand the **Single-Endpoint** category in the vectors list.

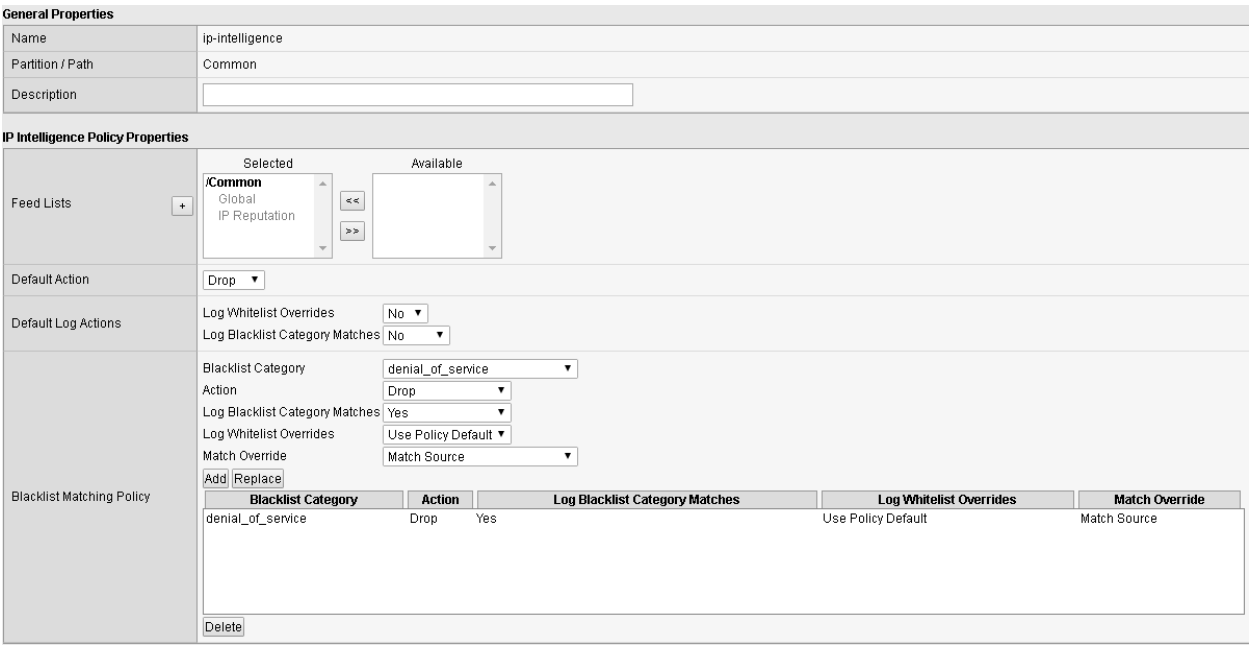
3. Click on **Single Endpoint Sweep** vector name.
4. Configure the vector with the following parameters:
    - State: Mitigate
    - Threshold Mode: Fully Manual
    - Detection Threshold EPS: 150
    - Mitigation Threshold EPS: 200
    - Add Source Address to Category: Checked
    - Category Name: denial_of_service
    - Sustained Attack Detection Time: 10 seconds
    - Category Duration Time: 60 seconds
    - Packet Type: Move All IPv4 to Selected



5. Click **Update** to save your changes.
6. Navigate to **Security** > **Network Firewall** > **IP Intelligence** > **Policies**.
7. In the **Global Policy** section, change the **IP Intelligence Policy** to **ip-intelligence**.



8. Click **Update**.
9. Click on the **ip-intelligence** policy in the policy list below.
10. Create a new Blacklist Matching Policy in the IP Intelligence Policy Properties section with the following attributes, leaving unspecified attributes with their default values:
    - Blacklist Category: denial-of-service
    - Action: drop
    - Log Blacklist Category Matches: Yes
11. Click **Add** to add the new Blacklist Matching Policy.



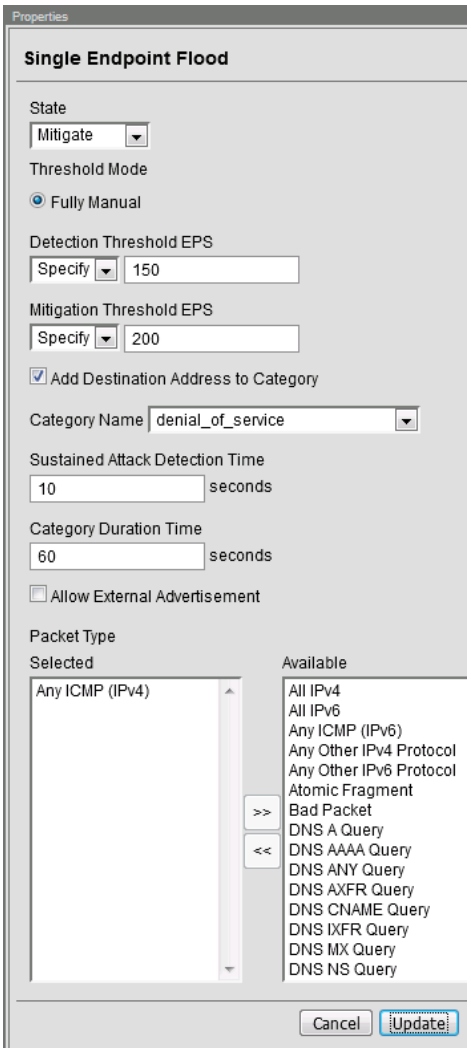
12. Click **Update** to save changes to the ip-intelligence policy.
13. Open the BIG-IP SSH session and scroll the ltm log in real time with the following command: `tail -f /var/log/ltm`
14. On the victim server, start a packet capture with an SSH filter by issuing sudo tcpdump -nn not port 22
15. On the attack host, launch the attack by issuing the following command on the BASH prompt:
    `sudo hping3 10.20.0.10 --flood --scan 1-65535 -d 128 -w 64 --syn`

16. You will see the scan find a few open ports on the server, and the server will show the inbound sweep traffic. However, you will notice that the traffic to the server stops after a short time (10 seconds, the configured sustained attack detection time.) Leave the test running.
17. After approximately 60 seconds, sweep traffic will return to the host. This is because the IP Intelligence categorization of the attack host has expired. After 10 seconds of traffic, the bad actor is again blacklisted for another 60 seconds.
18. Stop the sweep attack on the attack host by pressing **CTRL + C**.
19. Return to the BIG-IP web UI and navigate to **Security** > **Event Logs** > **DoS** > **Network** > **Events**. Observe the log entries showing the details surrounding the attack detection and mitigation.
20. Navigate to **Security** > **Event Logs** > **Network** > **IP Intelligence**. Observe the log entries showing the mitigation of the sweep attack via the ip-intelligence policy.
21. Navigate to **Security** > **Event Logs** > **Network** > **Shun**. Observe the log entries showing the blacklist adds and deletes.
22. Navigate to **Security** > **Reporting** > **Network** > **IP Intelligence**. Observe the statistics showing the sweep attack and mitigation. Change the **View By** drop-down to view the varying statistics.
23. Navigate to **Security** > **Reporting** > **DoS** > **Dashboard** to view an overview of the DoS attacks and timeline. You can select filters in the filter pane to highlight the specific attack.
24. Finally, navigate to **Security** > **Reporting** > **DoS** > **Analysis**. View detailed statistics around the attack.

Single Endpoint Flood

The single endpoint flood attack is an attempt for an attacker to send a flood of traffic to a host in hopes of overwhelming a service to a point of failure. In this example, we'll flood the target server with ICMP packets.

1. In the BIG-IP web UI, navigate to **Security** > **DoS Protection** > **Device Configuration** > **Network Security**.
2. Expand the **Single-Endpoint** category in the vectors list.
3. Click on **Single Endpoint Flood** vector name.
4. Configure the vector with the following parameters:
    - State: Mitigate
    - Threshold Mode: Fully Manual
    - Detection Threshold EPS: 150
    - Mitigation Threshold EPS: 200
    - Add Destination Address to Category: Checked
    - Category Name: denial_of_service
    - Sustained Attack Detection Time: 10 seconds
    - Category Duration Time: 60 seconds
    - Packet Type: Move Any ICMP (IPv4) to Selected



5. Click **Update** to save your changes.
6. Open the BIG-IP SSH session and scroll the ltm log in real time with the following command: `tail -f /var/log/ltm`
7. We'll run a packet capture on the victim server to gauge the incoming traffic. On the victim server, issue the following command: `sudo tcpdump -nn not port 22`
8. On the attack host, launch the attack by issuing the following command on the BASH prompt: `sudo hping3 10.20.0.10 --faster -c 25000 --icmp`
9. The attack host will begin flooding the victim server with ICMP packets. However, you will notice that the traffic to the server stops after a short time (10 seconds, the configured sustained attack detection time.)
10. After approximately 60 seconds, run the attack again. ICMP traffic will return to the host. This is because the IP Intelligence categorization of the attack host has expired.
11. Return to the BIG-IP web UI.
12. Navigate to **Security** > **Event Logs** > **DoS** > **Network** > **Events**. Observe the log entries showing the details surrounding the attack detection and mitigation.
13. Navigate to **Security** > **Event Logs** > **Network** > **IP Intelligence**. Observe the log entries showing

the mitigation of the sweep attack via the ip-intelligence policy.

14. Navigate to **Security** > **Reporting** > **Network** > **IP Intelligence**. Observe the statistics showing the sweep attack and mitigation.
15. Navigate to **Security** > **Reporting** > **DoS** > **Dashboard** to view an overview of the DoS attacks and timeline. You can select filters in the filter pane to highlight the specific attack.
16. Finally, navigate to **Security** > **Reporting** > **DoS** > **Analysis**. View detailed statistics around the attack.

This concludes the DoS/DDoS portion of the lab. You have successfully defeated Joanna, she has decided a career at Chotchkie's is more prosperous than nefarious internet activities, even with the new flair requirements. Well done!

Written for TMOS 13.1.0.1/BIG-IQ 6.0

1.5 Lab 4 - Device Management Workflows

1.5.1 Lab Overview

Day 3, you get a little curious and wonder why both BIG-IP's you've been working on say they're managed by BIG-IQ (look near the red f5 ball on the top left of both BIG-IP's). Unbelievable, all this time you've been configuring both devices independently when you could have been configuring them on a central management device.

Central Management Version - 6.0 was a major evolution of the BIG-IQ product line designed to become the primary source of centralized management for all physical and virtual F5 BIG-IP devices. BIG-IQ extends its offerings for security users, improving the user experience, and adding robustness and scale throughout the platform.

1.5.2 Base BIG-IQ Configuration

In this lab, the VE has been configured with the basic system settings and the VLAN/self-IP configurations required for the BIG-IQ to communicate and pass traffic on the network. Additionally, the Data Collection Device has already been added to BIG-IQ and the BIG-IP's have been imported and have been gathering health statistics. They have not however had their configurations imported.

1.5.3 New features

Statistics Dashboards

This is the real first step managing data statistics using a DCD (data collection device) evolving toward a true analytics platform. In this guide, we will explore setting up and establishing connectivity using master key to each DCD (data collection device).

- Enabling statistics for each functional area as part of the discovery process. This will allow BIG-IQ to proxy statistics gathered and organized from each BIG-IP device leveraging F5 Analytics iApp service (https://devcentral.f5.com/codeshare/f5-analytics-iapp).
- Configuration and tuning of statistic collections post discovery allowing the user to focus on data specific to their needs.
- Viewing and interaction with statistics dashboard, such as filtering views, differing time spans, selection and drilldown into dashboards for granular data trends and setting a refresh interval for collections.

Auto-scaling in a VMware cloud environment

You can now securely manage traffic to applications in a VMware cloud environment, specifying the parameters in a service scaling group to dynamically deploy and delete BIG-IP devices as needed. BIG-IQ manages the BIG-IP devices that are load balancing to the BIG-IP VE devices in the cloud, as well as to the BIG-IP devices' application servers.

Auto-scaling in an AWS environment

You can now securely manage traffic to applications in a VMware cloud environment, specifying the parameters in a service scaling group to dynamically deploy and delete BIG-IP devices as needed. You can manage the BIG-IP VE devices from a BIG-IQ system on-premises, or in the cloud. You have the option to use an F5 AWS Marketplace license, or your own BIG-IP license.

BIG-IQ VE deployment in MS Azure

You can now deploy a BIG-IQ VE in a MS Azure cloud environment.

Intuitive visibility for all managed applications

BIG-IQ now provides an overview of all managed applications with the option for a more detailed view of each application. Both the overview and detailed views provide information about the application's performance, Web Application Security status, and network statistics.

Easy application troubleshooting based on application traffic and security data

You can now enable enhanced analytics to view detailed application data in real-time, which allows you to isolate traffic characteristics that are affecting your application's performance and security status.

Real-time notifications for monitored devices and applications

You can now receive real time alerts and events for BIG-IP devices and their connected applications. These notifications are integrated into the BIG-IQ UI charts and allow you to pinpoint activities that are currently affecting your application.

Enhanced HTTP and Web Application Security visibility for all applications

You can use the HTTP and Web Application Security Dashboards to monitor all applications managed by BIG-IQ Centralized Management. These dashboards allow you to compare applications, pool members, and other aspects of traffic to your applications. In addition, the enhanced view includes real time events and alerts within the charts, and enhanced analytics data.

Added object and management support for DNS features

Creating, reading, updating, and deleting DNS GSLB objects, and listeners is now supported from the BIG-IQ user interface and the API.

Visibility into managed service scaling groups

An automatically scalable environment of BIG-IP VE devices can be defined to provide services to a set of applications. System administrators of BIG-IQ Centralized Management can monitor performance data for these BIG-IP VE devices.

Enhanced DNS visibility & configuration

BIG-IQ provides the ability to configure and have an enhanced view into DNS traffic, which now includes both peak traffic values and average traffic values over a selected period of time.

Application templates

Enhanced application/service templates that make deployments simple and repeatable.

Security policies and profiles available in applications

You can now add security policies and profiles to applications, including Web Application Security policies, Network Security firewall policies, DoS profiles, and logging profiles.

Automatically deploy policy learning

You can now enable automatic deployment of policy learning using Web Application Security.

Extended ASM/advanced WAF management that includes

- Auto-deploy policy learning
- Brute-force attack event monitoring
- Event correlation
- Manage DataSafe profiles
- Initial ASM and HTTP monitoring dashboards

Enhanced AFM Management

- AFM and DoS event visualization
- Multi device packet tester
- Enhanced debugging

APM enhancements

- Management capabilities for APM Federation through BIG-IQ (SAML, IdP and SP)
- Management capabilities for APM SSO configuration for Web Proxy Authentication Support Through BIG-IQ

Manage cookie protection

You can now manage cookie protection for BIG-IP devices using Web Application Security.

Monitoring dashboard for Web Application Security statistics

You can review Web Application Security policy statistics using a graphical dashboard.

Manage DataSafe profiles

You can now manage DataSafe profiles using Fraud Protection Security.

Enhanced support for NAT firewalls

You can now use the enhanced NAT firewall support in Network Security.

Subscriber support in firewall rules

You can now add subscriber IDs and groups to firewall rules in Network Security for BIG-IP devices that support them.

Firewall testing using packet flow reports

You can now create and view packet flow reports to test firewall configurations in Network Security.

Support for multiple BIG-IP devices with packet tester reports

You can now select multiple BIG-IP devices when generating packet tester reports in Network Security.

Renaming of firewall objects supported

You can now rename firewall objects, such as firewall policies in Network Security.

Enhanced support for DoS profiles, device DoS configurations, and scrubber profiles

You can now manage additional features of DoS profiles, device DoS configurations, and scrubber profiles that are found in BIG-IP version 13.1, such as new vectors, stress-based mitigation, DNS dynamic signatures, and VLAN support in scrubber profiles.

Copying device DoS configurations

You can now copy device DoS configurations from one BIG-IP device to multiple BIG-IP devices with the same version.

Viewing logs for DoS and firewall events in the user interface

You can now configure and view logging of DoS and firewall events, and for DoS events, see that information in a graphical format.

Additional details can be found in the full release notes:

https://support.f5.com/kb/en-us/products/big-iq-centralized-mgmt/releasenotes/product/relnote-big-iq-central-mgmt-6-0-0.html

BIG-IP Versions AskF5 SOL with this info:

https://support.f5.com/kb/en-us/solutions/public/14000/500/sol14592.html

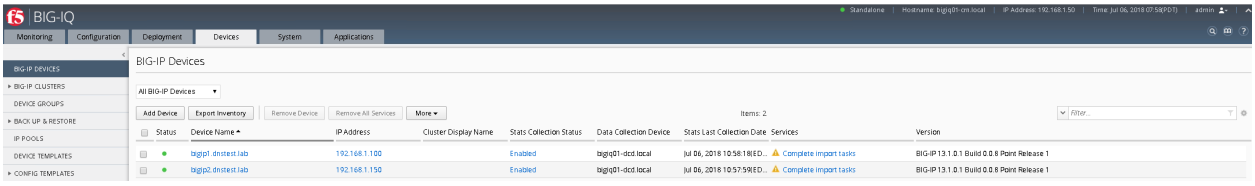
1.5.4 Changes to BIG-IQ User Interface

The user interface in the 6.0 release navigation has changed to a more UI tab-based framework.

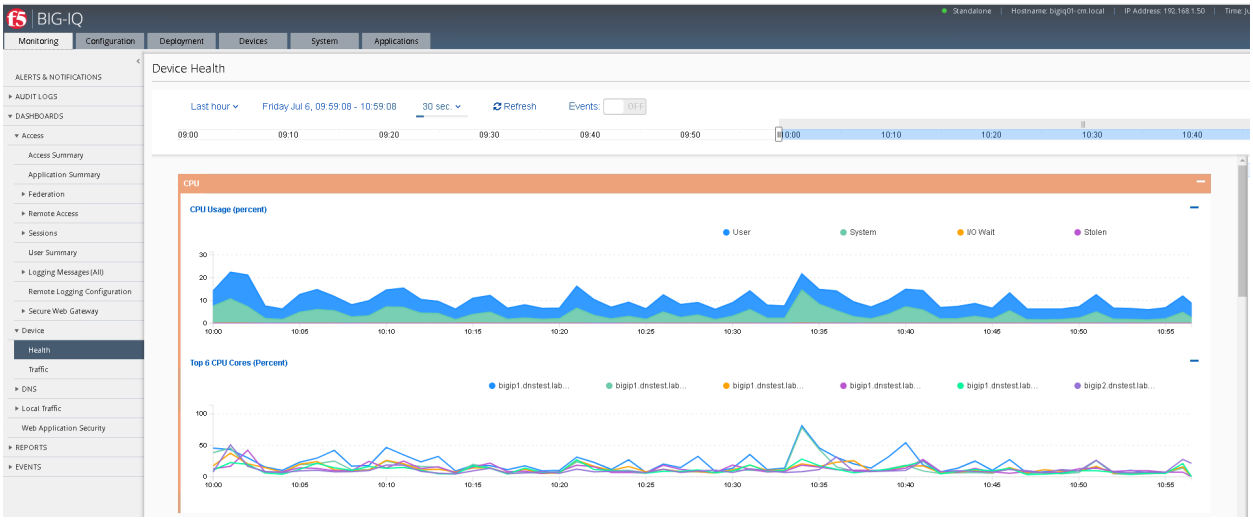
In this section, we will go through the main features of the user interface. Feel free to log into the BIG-IQ (https://192.168.1.50) username: admin password: 401elliottW! device to explore some of these features in the lab.

After you log into BIG-IQ, you will notice:

- A navigation tab model at the top of the screen to display each high level functional area.
- A tree based menu on the left-hand side of the screen to display low-level functional area for each tab.
- A large object browsing and editing area on the right-hand side of the screen.



- Let us look a little deeper at the different options available in the bar at the top of the page.

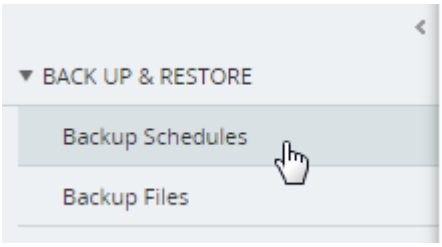


- At the top, each tab describes a high-level functional area for BIG-IQ central management:
- Monitoring –Visibility in dashboard format to monitor performance and isolate fault area.
- Configuration – Provides configuration editors for each module area.
- Deployment – Provides operational functions around deployment for each module area.
- Devices – Lifecycle management around discovery, licensing and software install / upgrade.
- System – Management and monitoring of BIG-IQ functionality.
- Applications – Build, deploy, monitor service catalog-based applications centrally.

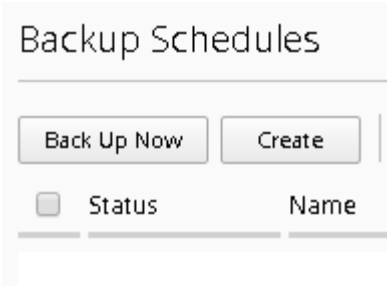
1.5.5 Workflow 1: Creating a Backup Schedule

BIG-IQ is capable of centrally backing up and restoring all the BIG-IP devices it manages. To create a simple backup schedule, follow the following steps.

1. Click on the **Back Up & Restore** submenu in the Devices header.
2. Expand the **Back Up and Restore** menu item found on the left and click on **Backup Schedules**

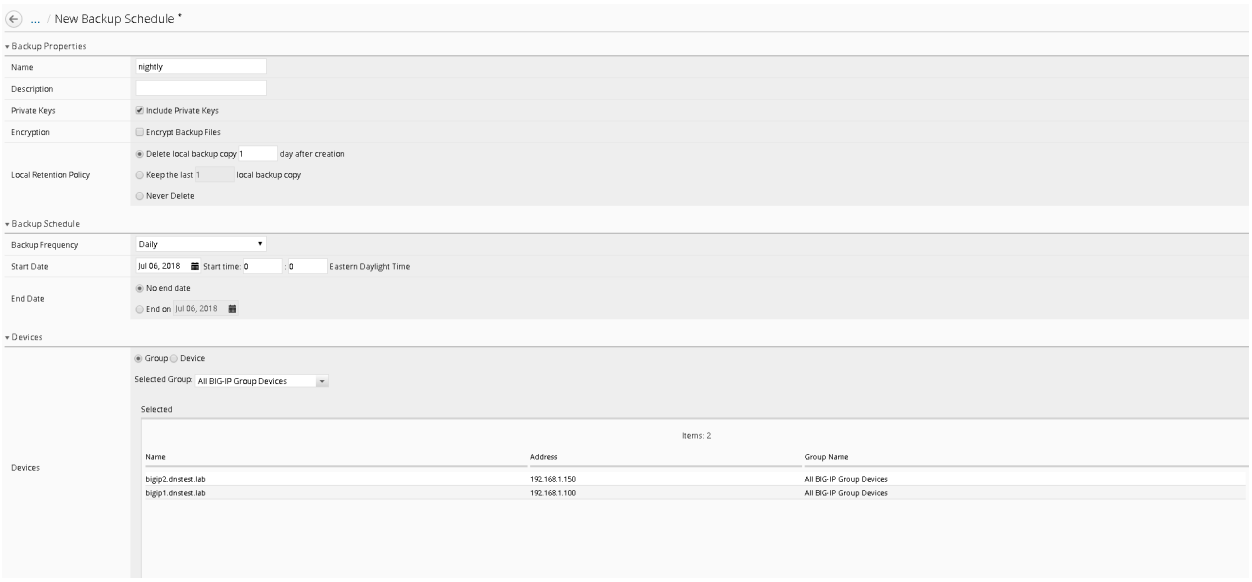


3. Click the **Create** button



4. Fill out the Backup Schedule using the following settings:
   - **Name:** Nightly
   - **Local Retention Policy:** Delete local backup copy 1 day after creation
   - **Backup Frequency:** Daily
   - **Start Time:** 00:00 Eastern Daylight Time
   - **Devices: Groups (radio button):** All BIG-IP Group Devices

   Your screen should look similar to the one below.



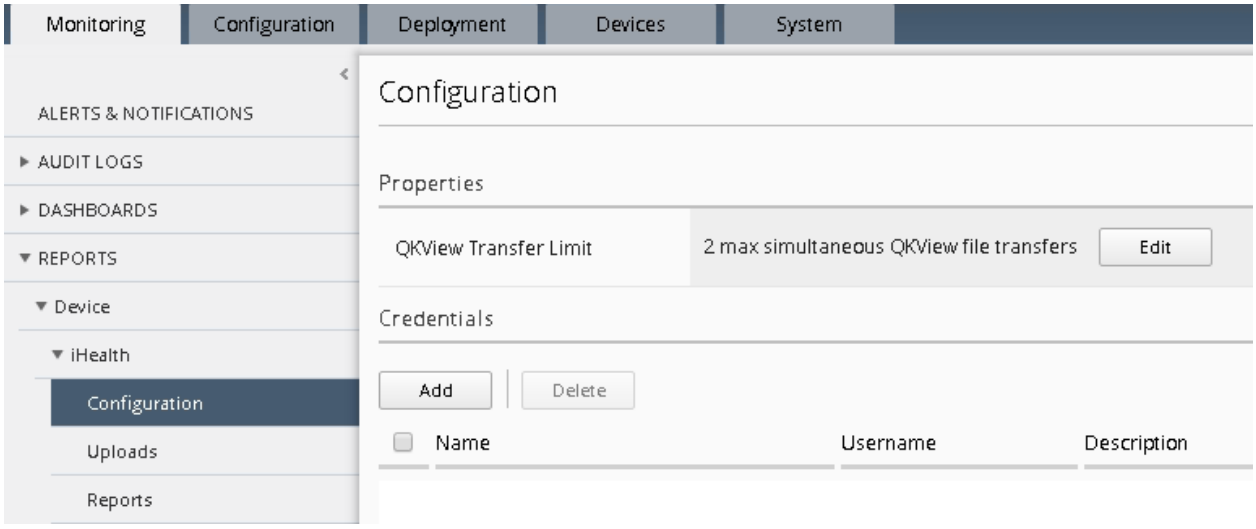
5. Click **Save & Close** to save the scheduled backup job.
6. Optionally feel free to select the newly created schedule and select “Run Schedule Now” to immediately backup the devices.
   - Add a Name for the Back Up
   - Click **Start**
   - When completed the backups will be listed under the Backup Files section

1.5.6 Workflow 2: Uploading QKviews to iHealth for a support case

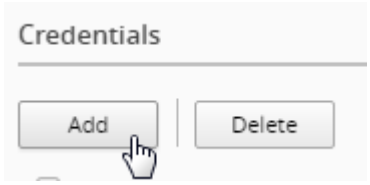
BIG-IQ can now push qkviews from managed devices to ihealth.f5.com and provide a link to the report of heuristic hits based on the qkview. These qkview uploads can be performed ad-hoc or as part of a

F5 support case. If a support case is specified in the upload job, the qkview(s) will automatically be associated/linked to the support case. In addition to the link to the report, the qkview data is accessible at ihealth.f5.com to take advantage of other iHealth features like the upgrade advisor.

1. Navigate to **Monitoring Reports Device iHealth Configuration**

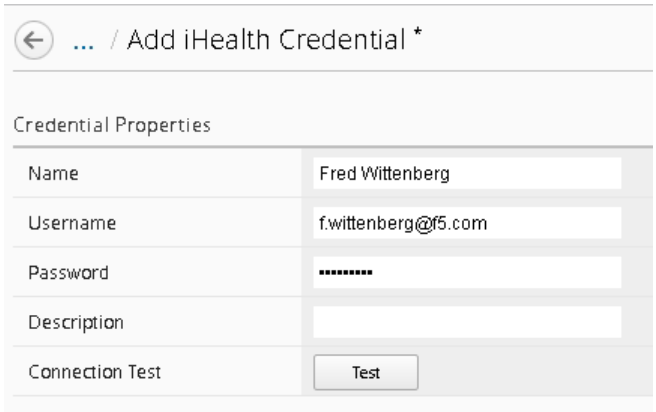


2. Add Credentials to be used for the qkview upload and report retrieval. Click the Add button under Credentials.



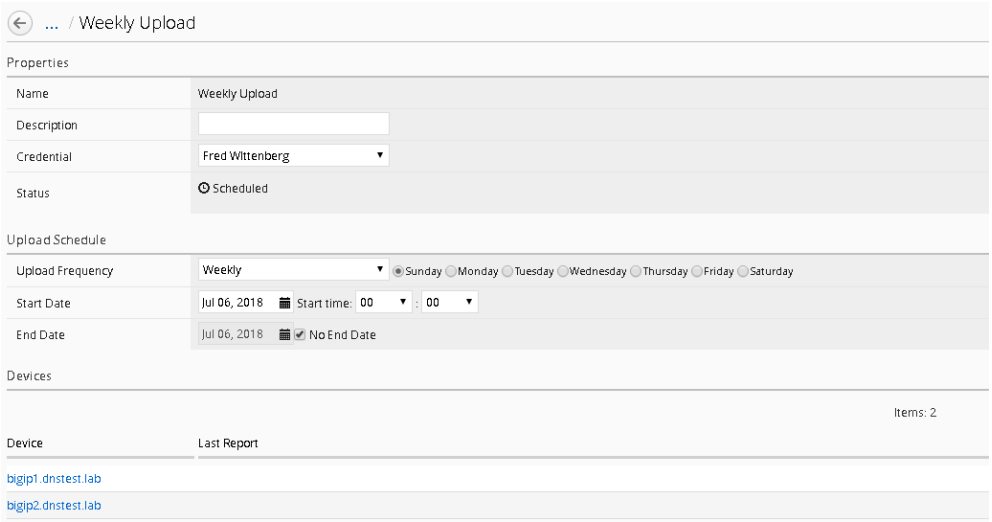
Warning: If you do not have credentials, please raise your hand and speak to an instructor

3. Fill in the credentials that you used to access https://ihealth.f5.com:
   - Name: Give the credentials a name to be referenced in BIG-IQ
   - Username: <Username you use to access iHealth.f5.com>
   - Password: <Password you use to access iHealth.f5.com>



4. Click the Test button to validate that your credentials work.
5. Click the Save & Close button in the lower right.
6. Click the QKview Upload Schedules button in the BIG-IP iHealth menu.

   Monitoring > **Reports** > **Device** > **iHealth** > **QKView Upload Schedule**
7. Click Create with the following values
   - Name – Weekly Upload
   - Description – Nightly QKView Upload
   - Credential – (use what was created in step 3)
   - Upload Frequecny – Weekly (Select Sunday)
   - Start Time – Select todays date at 00:00
   - End Date – No End date should be checked
   - Select both devices
   - Click the right arrow to move to the “Selected” Area
   - Click Save & Close.

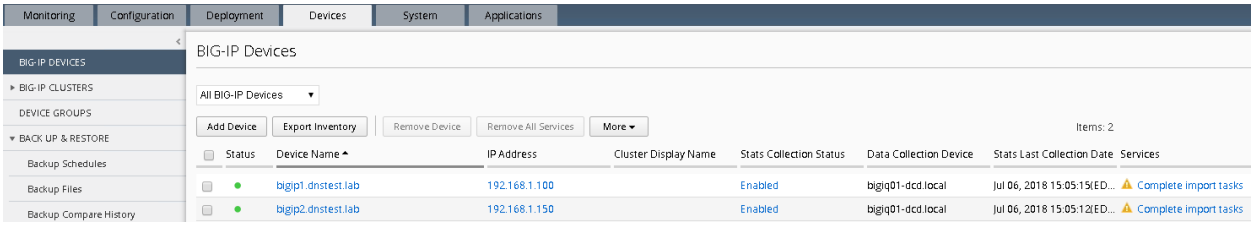


You will now have a fresh set of QKView in iHealth every Sunday morning. This is extremely useful for when new cases are opened, one less step you’ll need for support to engage quicker.

1.5.7 Workflow 3: Device Import

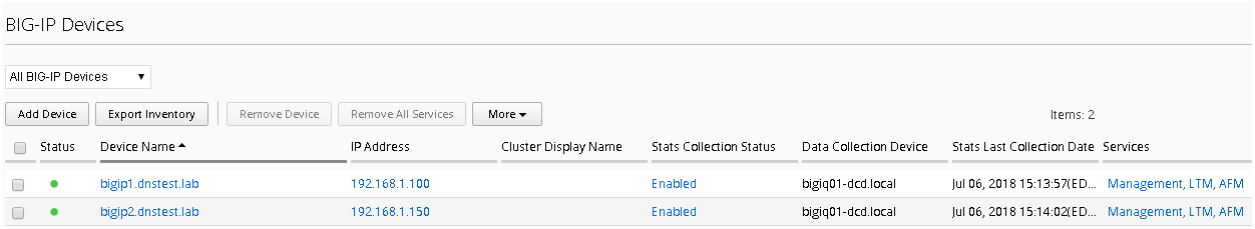
BIG-IQ is capable of centrally managing multiple products, for this lab we will only manage LTM and AFM. To import the device configurations, follow the steps below

1. Navigate to the Devices tab and click on **BIG-IP Devices** (left panel)



2. You'll notice both devices have not completed the import tasks, to remedy this simply click on the "Complete Import Tasks" Link
3. First Re-discover the LTM service
4. Then Discover the AFM service
5. Once Re-discovery has completed, import both the LTM and AFM services
6. Repeat this same procedure for both devices, once completed your screen will show the following.

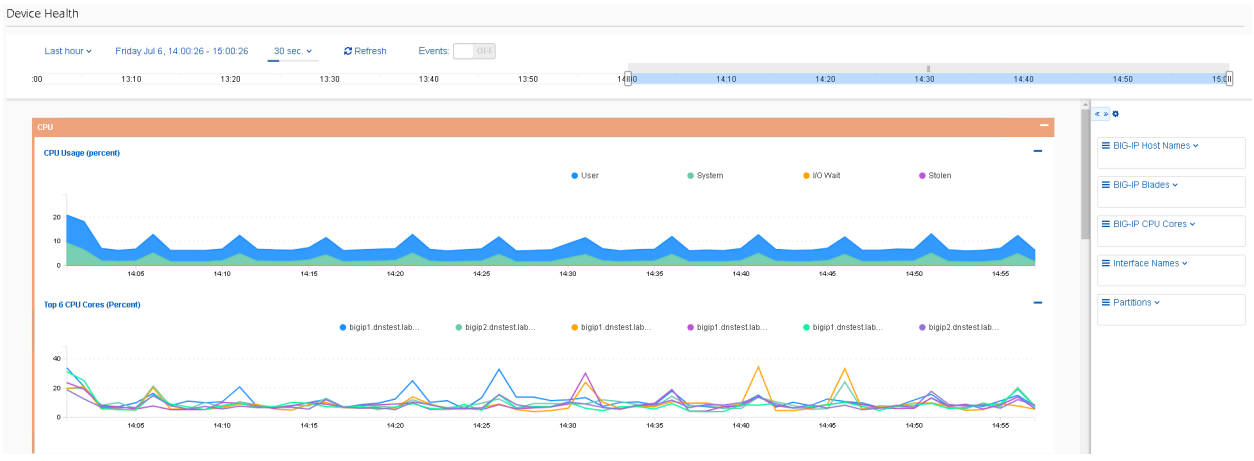
Note: For any conflicts you may encounter – leave BIG-IQ selected resolution



1.5.8 BIG-IQ Statistics Dashboards

Workflow 1: Reviewing the data in the dashboards

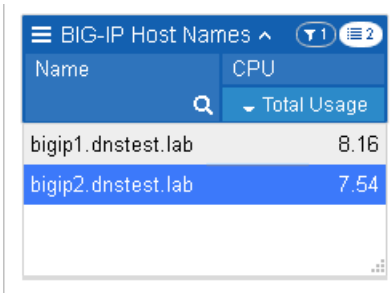
Navigate to **Monitoring Dashboards Device Health**



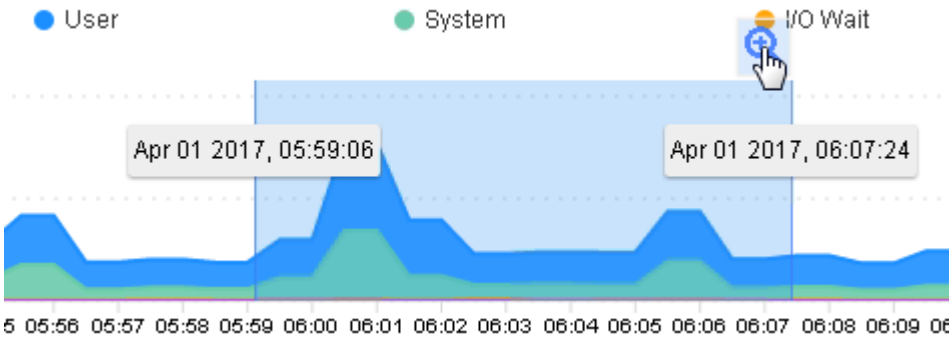
1.5.9 Workflow 2: Interacting with the data in the dashboards

- You can narrow the scope of what is graphed by selecting a object or objects from the selection panels on the right. For example, if you only want to see data from BIG-IP01, you can click on it to

filter the data.

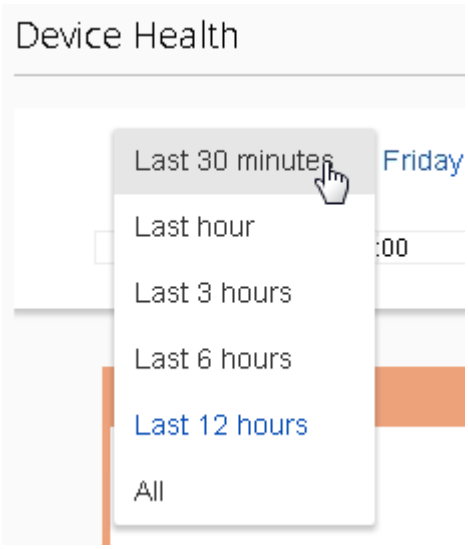


- You can create complex filters by making additional selections in other panels
- You can zoom in on a time, by selecting a section of a graph or moving the slider at the top of the page



or

- All the graphs update to the selected time.
- You can change how far in the data you want to look back by using the selection in the upper left (note you may need to let some time elapse before this option becomes available)



Written for TMOS 13.1.0.1/BIG-IQ 6.0

IT agility. Your way.

1.6 Lab 5 - Network Security (AFM) Management Workflows

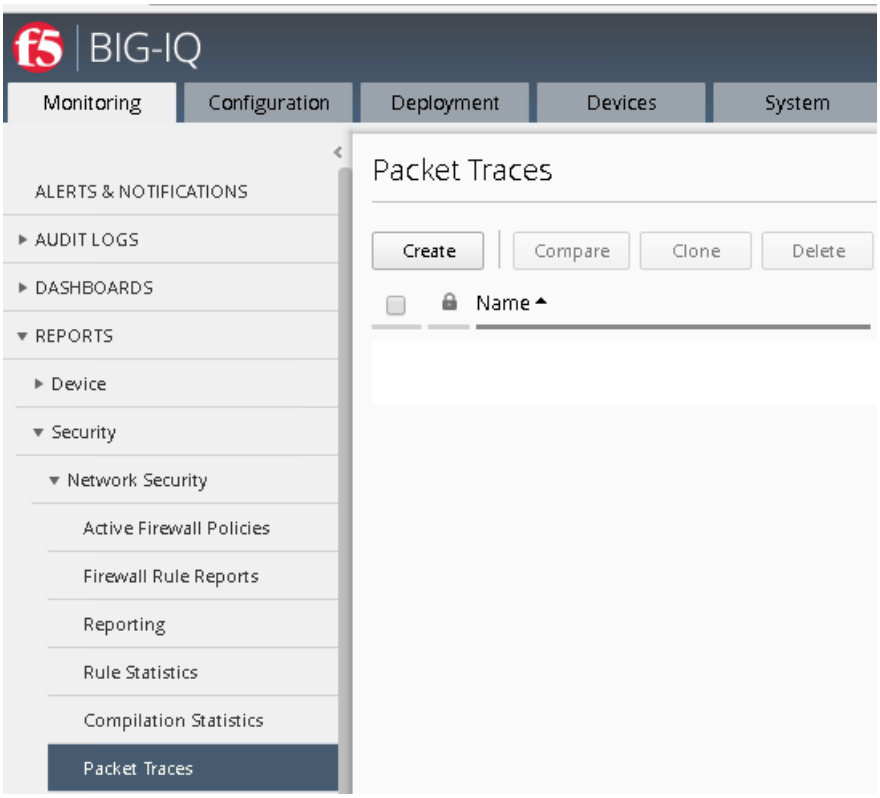
1.6.1 Network Security (AFM) Management Workflows

Workflow 1: Managing AFM from BIG-IQ

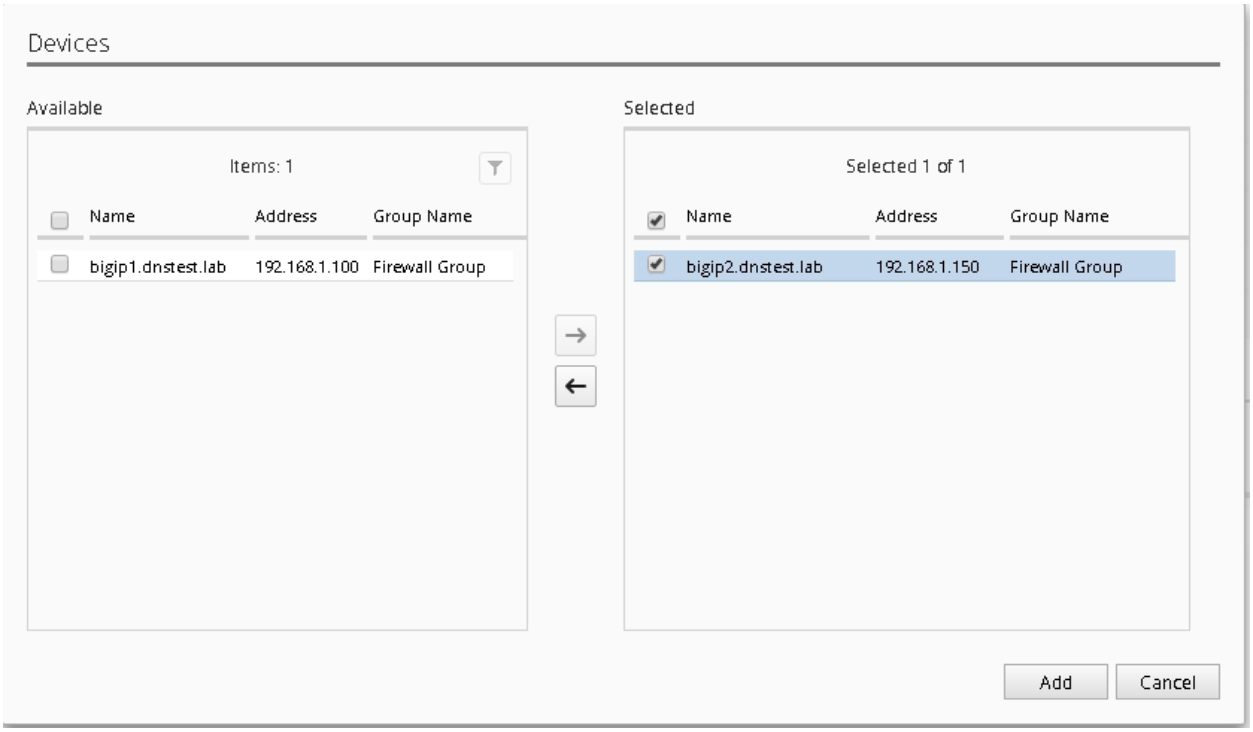
Day 4, it turns out no one thought about managing the new web and application servers, as such SSH is blocked to both devices. Let's first validate this by using the packet tester tool within BIG-IQ, note this is the same tool within BIG-IP with one major exception. Within BIG-IQ you can trace a packet through **more than one firewall**. This is very useful if you have multiple AFM devices in a packets path, now you can test the flow end to end from one central location.

Task 1 – Packet Tracer

1. Navigate to **Monitoring > Reports > Security > Network Security > Packet Traces**

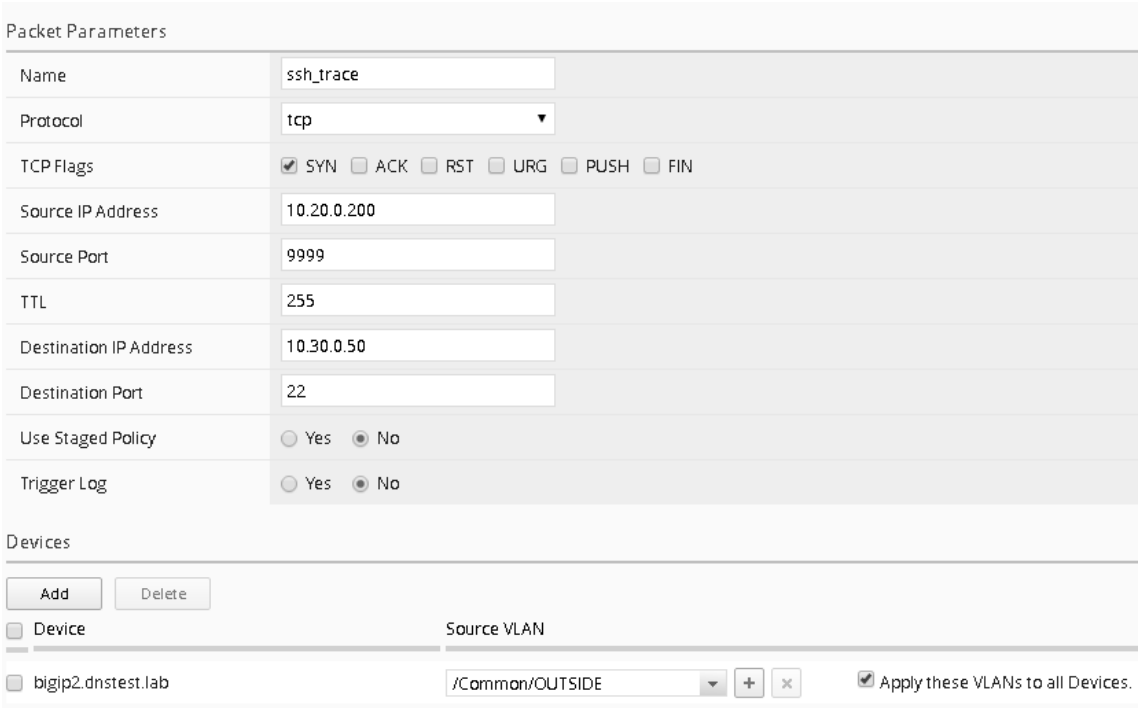


2. Click on the “Create” button from the top menu.
3. Complete the following information
    - Name – ssh_trace
    - Protocol – tcp
    - TCP Flags – Syn
    - Source IP Address – 10.20.0.200
    - Source Port – 9999
    - Destination IP Address – 10.30.0.50
    - Destination Port – 22
    - Use Staged Policy – No
    - Trigger Log – No
4. Under the Devices section click “Add” (notice you’ll see all the devices with AFM provision listed), for our lab however; just add **bigip2.dnstest.lab**



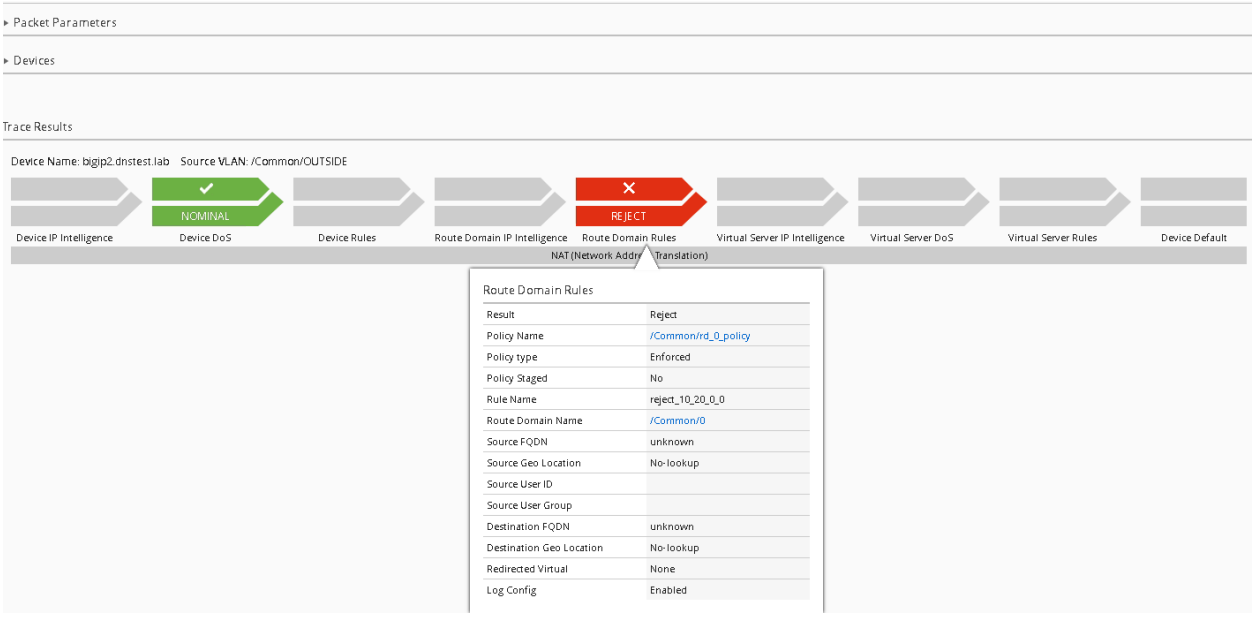
5. Select the “/Common/OUTSIDE” Vlan as the Source VLAN from the dropdown.

   When completed your screen should look like the screen shot below:



6. Click “Run Trace”

   You can see from the trace results; the traffic is indeed being denied



Another nice feature of Packet Trace within BIG-IQ is the ability to clone a trace, when you complete the next two tasks, we'll return to the packet tracer tool to re-run the results using the clone option. Additionally, the traces are saved and can be reviewed later, this can be very helpful in long troubleshooting situations where application teams are asking for results after changes are made to policies.

Follow the steps below to allow SSH access to both devices using BIG-IQ as a central management tool.

Task 2 – Modify Rule Lists

1. Navigate to the **Configuration > Security > Network Security > Rule Lists**
2. Notice the previously created rule lists have been imported into BIG-IQ
3. Click on the "**application_rule_list**"
4. Click **Create Rule** button.
5. Click on the pencil (edit rule) of the newly created rule listed with **Id** of **2.**
6. Create a new rule with the below information. Be prepared to scroll to find all the options

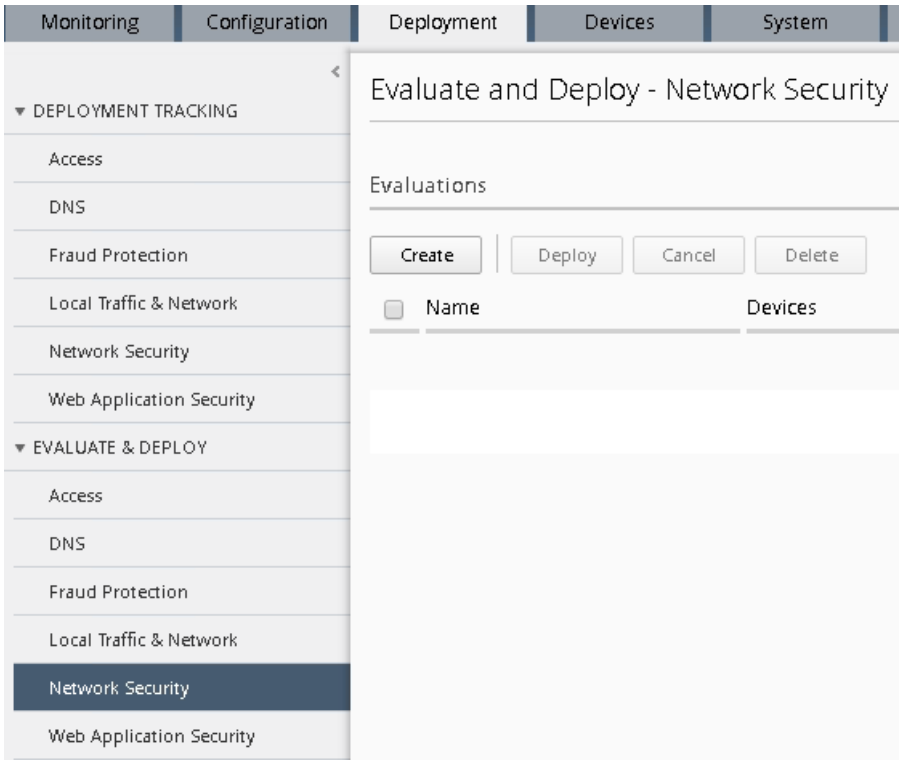
Name	allow_ssh
Source Address	**10.20.0.200**
Source Port	**any**
Source VLAN	**any**
Destination Address	**10.30.0.50**
Destination Port	**22**
Action	**Accept-Decisively**
Protocol	**TCP**
State	**enabled**
Log	**True (checked)**

7. Click **Save & Close** when finished.
8. Repeat the same procedure for the web_rule_list, be sure to change the destination to 10.30.0.50, all other setting remains the same.

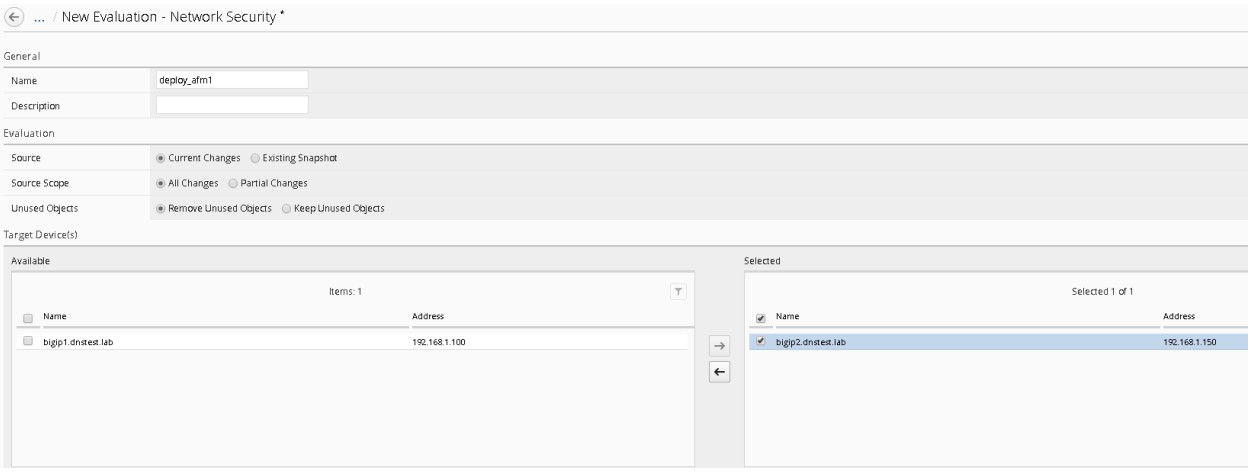
Task 3 – Deploy the Firewall Policy and related configuration objects

Now that the desired firewall configuration has been created on the BIG-IQ, you need to deploy it to the BIG-IP. In this task, you create the deployment, verify it, and deploy it.

1. From the top navigation bar, click on **Deployment** (tab).
2. Click on the **EVALUATE & DEPLOY** section on the left to expand it.
3. Click on **Network Security** in the expansion.



4. Click on the top **Create** button under the **Evaluations** section.
5. Give your evaluation a name (ex: **deploy_afm1**).
6. Evaluation **Source** should be **Current Changes** (default).
7. Source Scope should be **All Changes** (default)
8. Remove Unused Objects should be **Remove Unused Objects** (default)
9. Target Device(s) should be **Device**.
10. Select **bigip2.dnstest.lab** from the list of Available devices and move it to Selected area.



11. Click the **Create** button at the bottom right of the page.

    You should be redirected to the main **Evaluate and Deploy** page.

    This will start the evaluation process in which BIG-IQ compares its working configuration to the configuration active on each BIG-IP. This can take a few moments to complete.

    The **Status** section should be dynamically updating. . . (What states do you see?)

    Once the status shows **Evaluation Complete** you can view the evaluation results.

    Note: Before selecting to deploy, feel free to select the differences indicated to see the proposed deployment changes. This is your check before making changes on a BIG-IP.

    ---

12. Click the number listed under **Differences – Firewall**.
13. Scroll through the list of changes to be deployed.
14. Click on a few to review in more detail.

    What differences do you see from the **Deployed on BIG-IP** section and on **BIG-IQ**?

    Do you see the new rules you created in BIG-IQ? Ya should. . .

15. Click **Cancel**.

    Deploy your changes by checking the box next to your evaluation **deploy_afm1**.

16. With the box checked, click the **Deploy** button.

    Your evaluation should move to the **Deployments** section.

    After deploying, the status should change to **Deployment Complete**.

    - This will take a moment to complete. Once completed, log in to the BIG-IP and verify that the changes have been deployed to the AFM configuration.

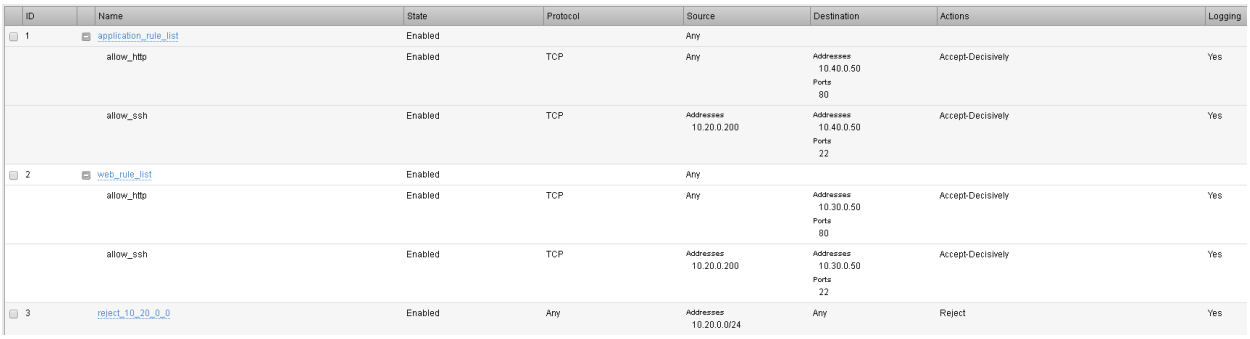
Congratulations, you just deployed your first AFM policy via BIG-IQ!

Review the configuration deployed to the BIG-IP units.

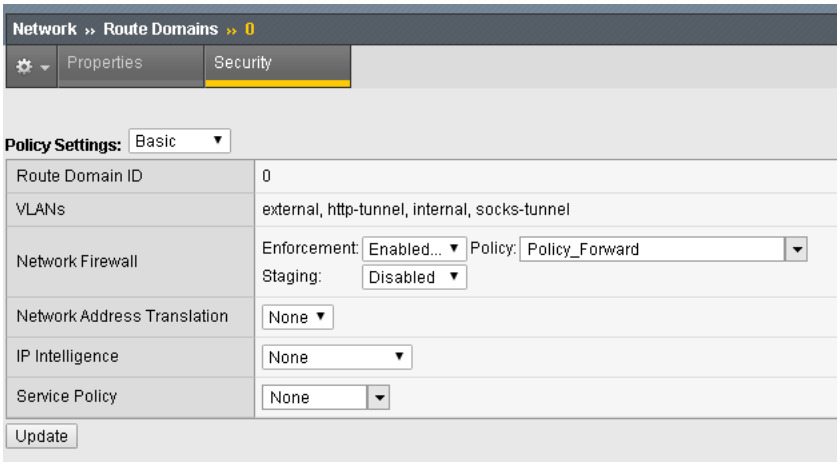
On **bigip2.dnstest.lab**: (https://192.168.1.150)

1. Navigate to Security > Network Firewall > Policies.
2. Click on rd_0_policy and expand the rule lists

   Are the two rules you created in BIG-IQ listed for this newly deployed firewall policy?



ID	Name	State	Protocol	Source	Destination	Actions	Logging
1	application_rule_list	Enabled		Any			
	allow_http	Enabled	TCP	Any	Addresses 10.40.0.50 Ports 80	Accept-Decisively	Yes
	allow_ssh	Enabled	TCP	Addresses 10.20.0.200	Addresses 10.40.0.50 Ports 22	Accept-Decisively	Yes
2	web_rule_list	Enabled		Any			
	allow_http	Enabled	TCP	Any	Addresses 10.30.0.50 Ports 80	Accept-Decisively	Yes
	allow_ssh	Enabled	TCP	Addresses 10.20.0.200	Addresses 10.30.0.50 Ports 22	Accept-Decisively	Yes
3	reject_10_20_0_0	Enabled	Any	Addresses 10.20.0.0/24	Any	Reject	Yes



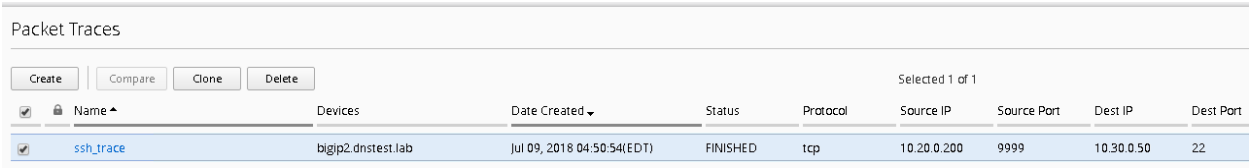
Test Access:

1. Open a new Web browser and access http://10.30.0.50
2. Open Putty and access 10.30.0.50

Task 4 – Packet Tracer (continued)

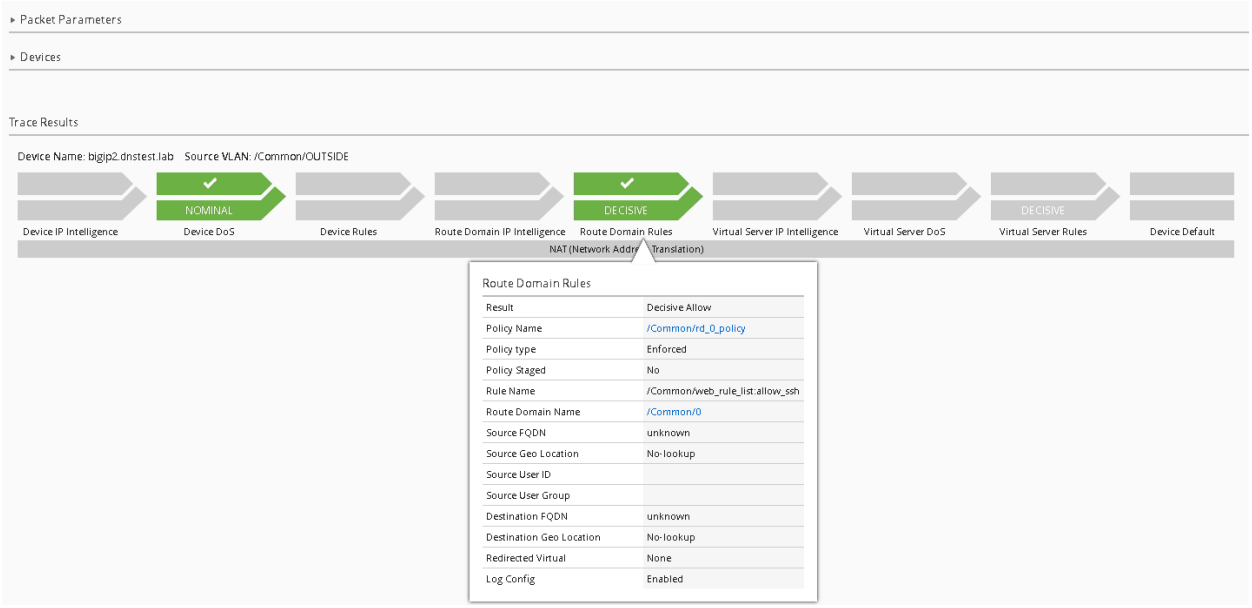
#. Navigate to the Monitoring tab Reports Security Network Security Packet Tracers

1. Highlight the previous trace (ssh_trace) and click on the “Clone” button



You’ll notice all the previously entered values are pre-populated, you now can make any changes if necessary (maybe the application team realized the source port of the flow is not random).

2. Click “Run Trace”



SUCCESS!!

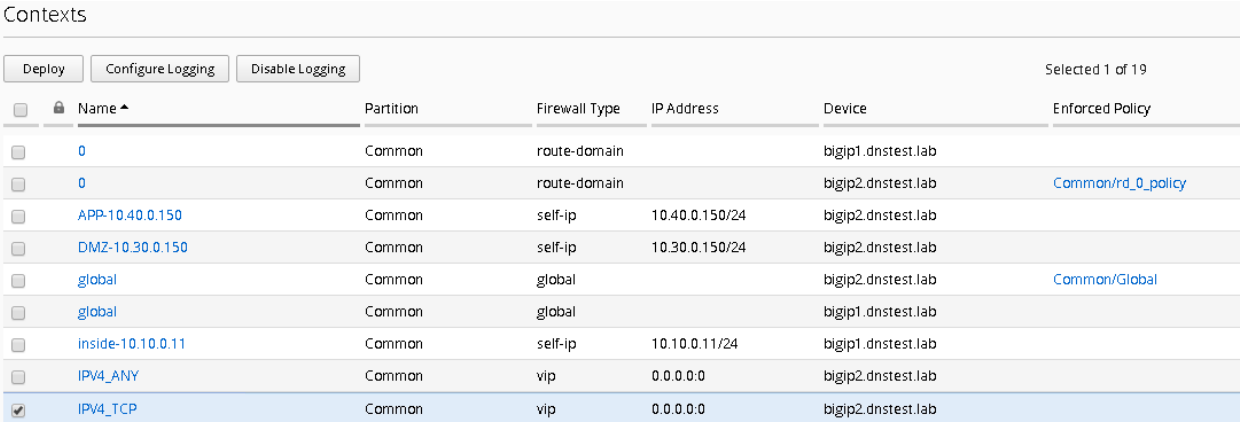
The history within the tool makes Root Cause Analysis (RCA) reports very easy, this allows the security team to show a denied flow and subsequent permitted flow.

1.6.2 Workflow 2: Configure Network Security and DoS Event Logging

Task 1 – Configure Network Security and DoS Event Logging

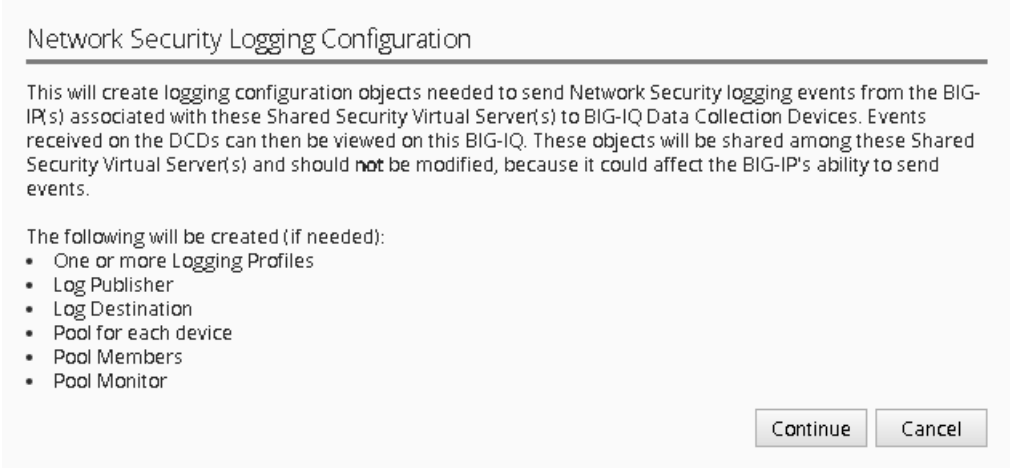
You enable Network Security event logging using the virtual servers displayed in the context list

1. Navigate to the Configuration Security Network Security Contexts
2. Check the box next to the IPV4_TCP VIP
3. Select “Configure Logging” from the top buttons



Name	Partition	Firewall Type	IP Address	Device	Enforced Policy
0	Common	route-domain		bigip1.dnstest.lab	
0	Common	route-domain		bigip2.dnstest.lab	Common/rd_0_policy
APP-10.40.0.150	Common	self-ip	10.40.0.150/24	bigip2.dnstest.lab	
DMZ-10.30.0.150	Common	self-ip	10.30.0.150/24	bigip2.dnstest.lab	
global	Common	global		bigip2.dnstest.lab	Common/Global
global	Common	global		bigip1.dnstest.lab	
inside-10.10.0.11	Common	self-ip	10.10.0.11/24	bigip1.dnstest.lab	
IPV4_ANY	Common	vip	0.0.0.0:0	bigip2.dnstest.lab	
IPV4_TCP	Common	vip	0.0.0.0:0	bigip2.dnstest.lab	

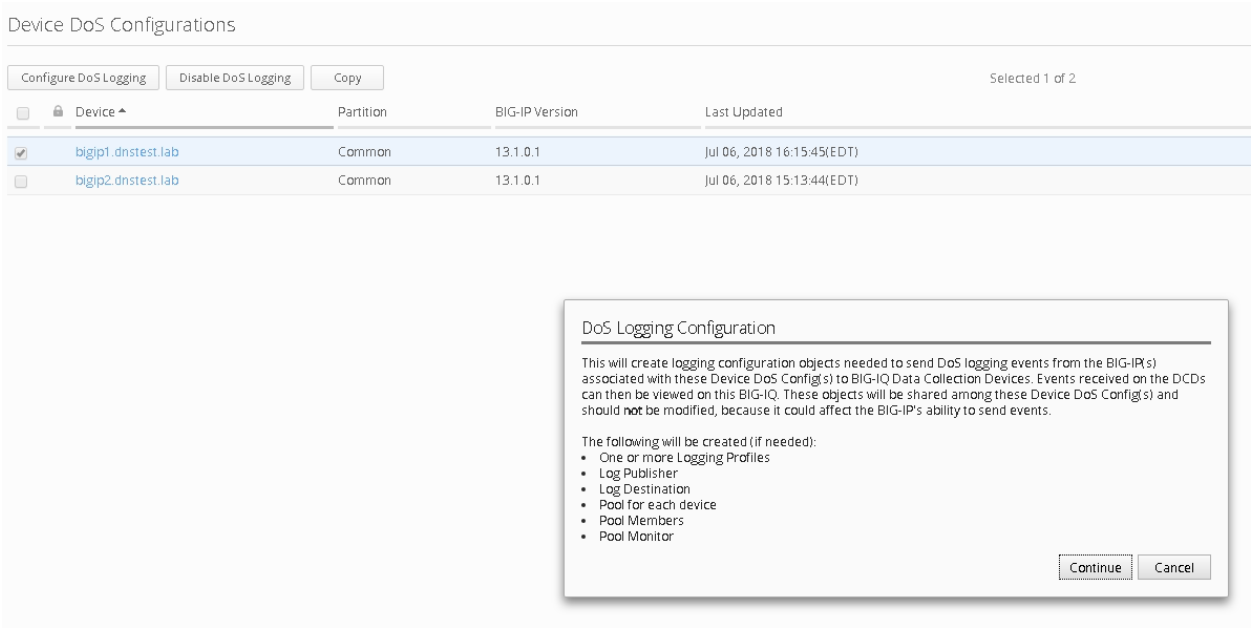
4. You will receive a configuration message alerting you to the changes about to be made to the device, click Continue



This will now configure a logging profile, associated pools, monitors and all necessary configuration to send logs to the Data Collection Device (DCD).

In the spirit of central management, we're also going to configure the DoS event logging, so we only must perform one deployment on both devices.

1. Navigate to Configuration Security Shared Security DoS Protection Device DoS Configurations
2. Highlight bigip1.dnstest.lab and click the "Configure DoS Logging" button from the top.



3. Once again you will receive a configuration message, click continue
4. Once completed navigate to the Deployments tab

   As most of the configuration is "LTM" related you will first need to deploy the LTM configuration.
5. Navigate to Evaluate & Deploy
6. Select Local Traffic & Network Traffic
7. Create an evaluation named "logging_configuration", leave all other defaults and select both devices, once finished, create the evaluation.

Feel free to examine the changes in the evaluation, when satisfied deploy the changes.

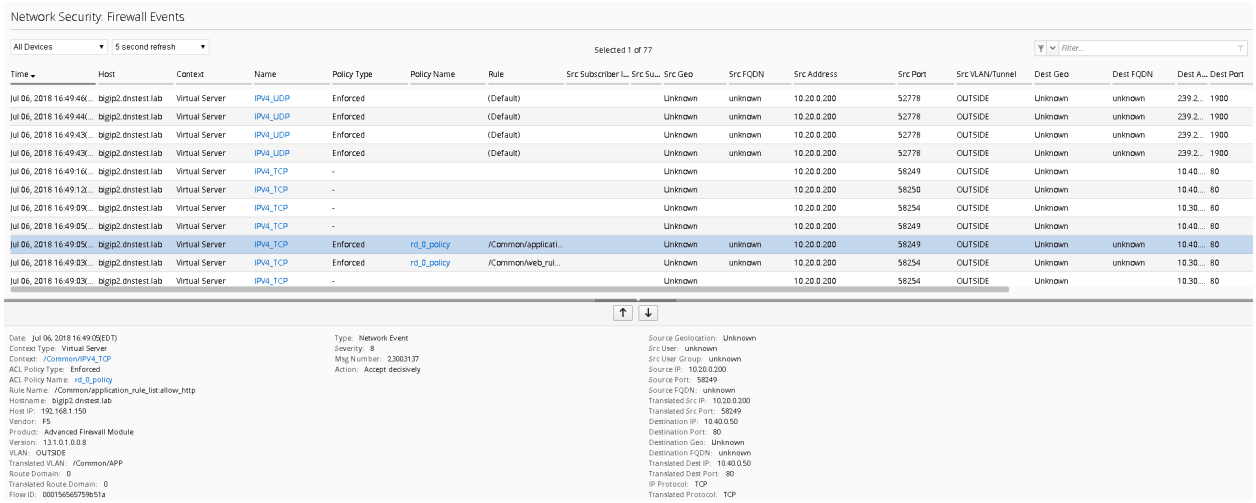
8. Once the LTM configuration is deployed, you'll need to also deploy the Network Security portion of the changes.

   Navigate to Deployment Evaluate & Deploy Network Security.

Again, create an evaluation and subsequent deployment for both devices.

Task 2 – Evaluate Network Firewall Events

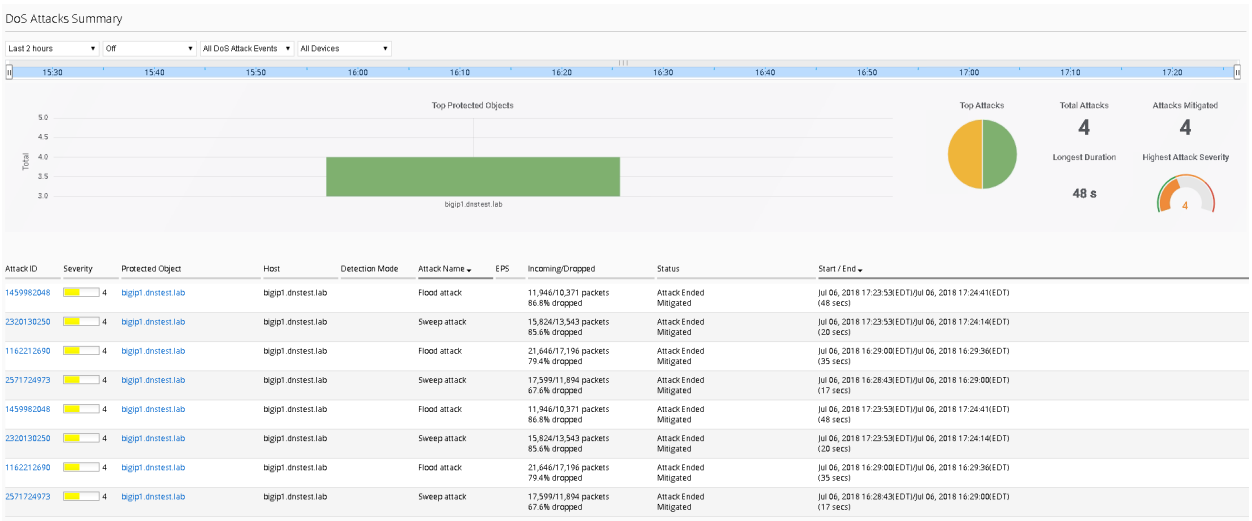
1. Browse to http://10.30.0.50 once again (or refresh in your tabs).
2. Within BIG-IQ, navigate to Monitoring Network Security Firewall
3. Click on a line item for enriched information in the window below as shown



Feel free to view other logs to see the data presented.

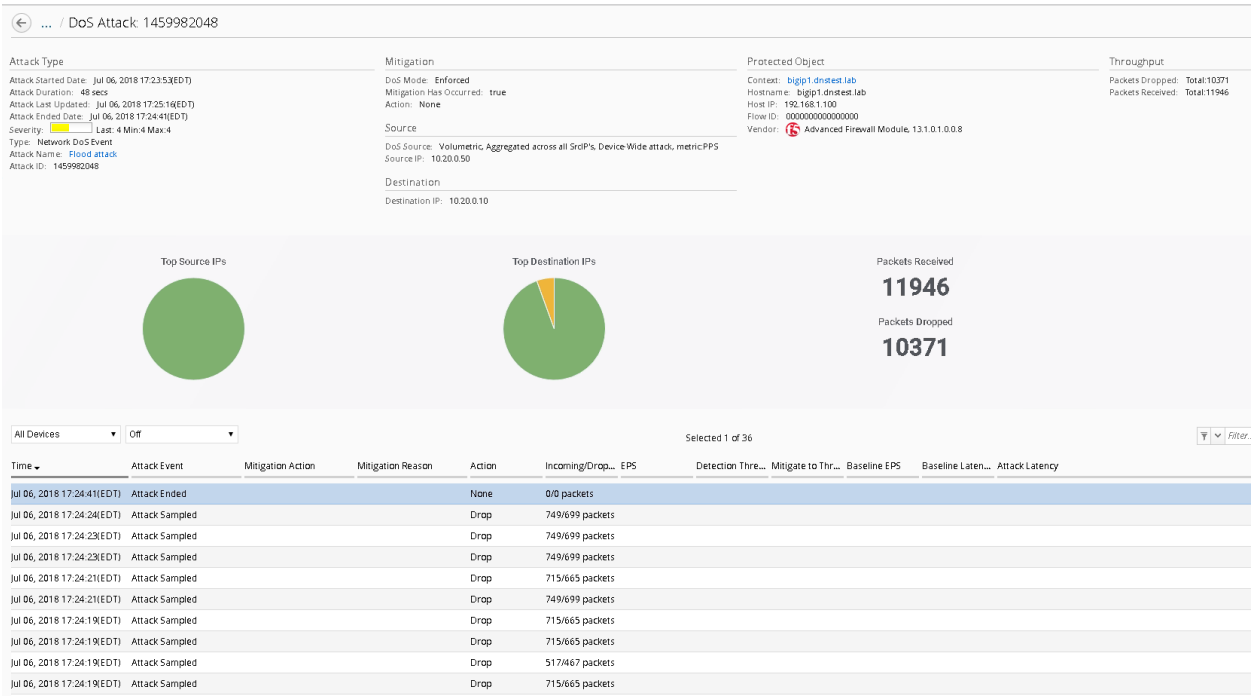
Task 3 – Evaluate DoS Events

1. Open a few separate windows to the attack host. We will launch a few attacks at once to see the value of consolidated reporting within BIG-IQ (there is a text document on the jumbox desktop which contains all of the attack commands).
2. Launch a few attacks at once and navigate to Monitoring Events –DoS DoS Summary



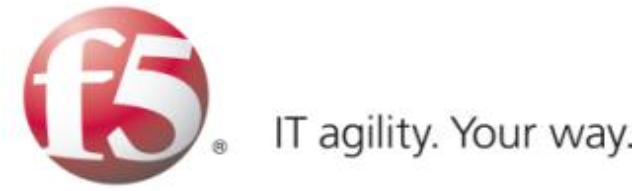
3. From here you have a consolidated view of all your devices and attacks.

   Click on one of the attack ID's for enriched information about the attack



This concludes the lab. You have had quite the eventful first week at Initech! You have successfully allowed communication to a new webserver, you tuned and defended against several DoS attacks, you then configured BIG-IQ for central device management and monitoring and lastly, you're now managing AFM within BIG-IQ. I think you deserve Friday off!!

Written for TMOS 13.1.0.1/BIG-IQ 6.0



1.7 Lab 6 - iControl REST API

1.7.1 Lab 6 Overview

It's Friday, you've made it through week one, but its not over yet. After another meeting with the Bob's they've decided they want to explore the SecOps world and configure devices through the REST API. Before we proceed let's learn a little about what REST is and how to interact with the F5 API, also known as iControl.

1.7.2 About Representational State Transfer

Representational State Transfer (REST) describes an architectural style of web services where clients and servers exchange representations of resources. The REST model defines a resource as a source of information and defines a representation as the data that describes the state of a resource. REST web services use the HTTP protocol to communicate between a client and a server, specifically by means of the POST, GET, PUT, and DELETE methods to create, read, update, and delete elements or collections. In general terms, REST queries resources for the configuration objects of a BIG-IP® system, and creates, deletes, or modifies the representations of those configuration objects. The iControl® REST implementation follows the REST model by:

- Using REST as a resource-based interface, and creating API methods based on nouns.
  - Employing a stateless protocol and MIME data types, as well as taking advantage of the authentication mechanisms and caching built into the HTTP protocol.
- Supporting the JSON format for document encoding.
  - Representing the hierarchy of resources and collections with a Uniform Resource Identifier (URI) structure.
  - Returning HTTP response codes to indicate success or failure of an operation.
- Including links in resource references to accommodate discovery.

1.7.3 About URI format

The iControl® REST API enables the management of a BIG-IP® device by using web service requests. A principle of the REST architecture describes the identification of a resource by means of a Uniform Resource Identifier (URI). You can specify a URI with a web service request to create, read, update, or delete some component or module of a BIG-IP system configuration. In the context of REST architecture, the system configuration is the representation of a resource. A URI identifies the name of a web resource; in this case, the URI also represents the tree structure of modules and components in TMSH.

In iControl REST, the URI structure for all requests includes the string /mgmt/tm/ to identify the namespace for traffic management. Any identifiers that follow the endpoint are resource collections.

Tip: Use the default administrative account, admin, for requests to iControl REST. Once you are familiar with the API, you can create user accounts for iControl REST users with various permissions.

https://management-ip/mgmt/tm/module

The URI in the previous example designates all of the TMSH subordinate modules and components in the specified module. iControl REST refers to this entity as an organizing collection. An organizing collection contains links to other resources. The management-ip component of the URI is the fully qualified domain name (FQDN) or IP address of a BIG-IP device.

Important: iControl REST only supports secure access through HTTPS, so you must include credentials with each REST call. Use the same credentials you use for the BIG-IP device manager interface.

For example, use the following URI to access all the components and subordinate modules in the LTM module:

https://management-ip/mgmt/tm/ltm

The URI in the following example designates all of the subordinate modules and components in the specified sub-module. iControl REST refers to this entity as a collection; a collection contains resources.

https://management-ip/mgmt/tm/module/sub-module

The URI in the following example designates the details of the specified component. The Traffic Management Shell (TMSH) Reference documents the hierarchy of modules and components, and identifies details of each component. iControl REST refers to this entity as a resource. A resource may contain links to sub-collections.

https://management-ip/mgmt/tm/module/{[}sub-module{]}/component

1.7.4 About reserved ASCII characters

To accommodate the BIG-IP® configuration objects that use characters, which are not part of the unreserved ASCII character set, use a percent sign (%) and two hexadecimal digits to represent them in a URI. The unreserved character set consists of: [A - Z] [a - z] [0 - 9] dash (-), underscore (_), period (.), and tilde (~).

You must encode any characters that are not part of the unreserved character set for inclusion in a URI scheme. For example, an IP address in a non-default route domain that contains a percent sign to indicate an address in a specific route domain, such as 192.168.25.90%3, should be encoded to replace the %character with %25.

1.7.5 About REST resource identifiers

A URI is the representation of a resource that consists of a protocol, an address, and a path structure to identify a resource and optional query parameters. Because the representation of folder and partition names in TMSH often includes a forward slash (/), URI encoding of folder and partition names must use a different character to represent a forward slash in iControl®

To accommodate the forward slash in a resource name, iControl REST maps the forward slash to a tilde (~) character. When a resource name includes a forward slash (/) in its name, substitute a tilde (~) for the forward slash in the path. For example, a resource name, such as /Common/plist1, should be modified to the format shown here:

https://management-ip/mgmt/tm/security/firewall/port-list/~Common~plist1

1.7.6 About Postman – REST Client

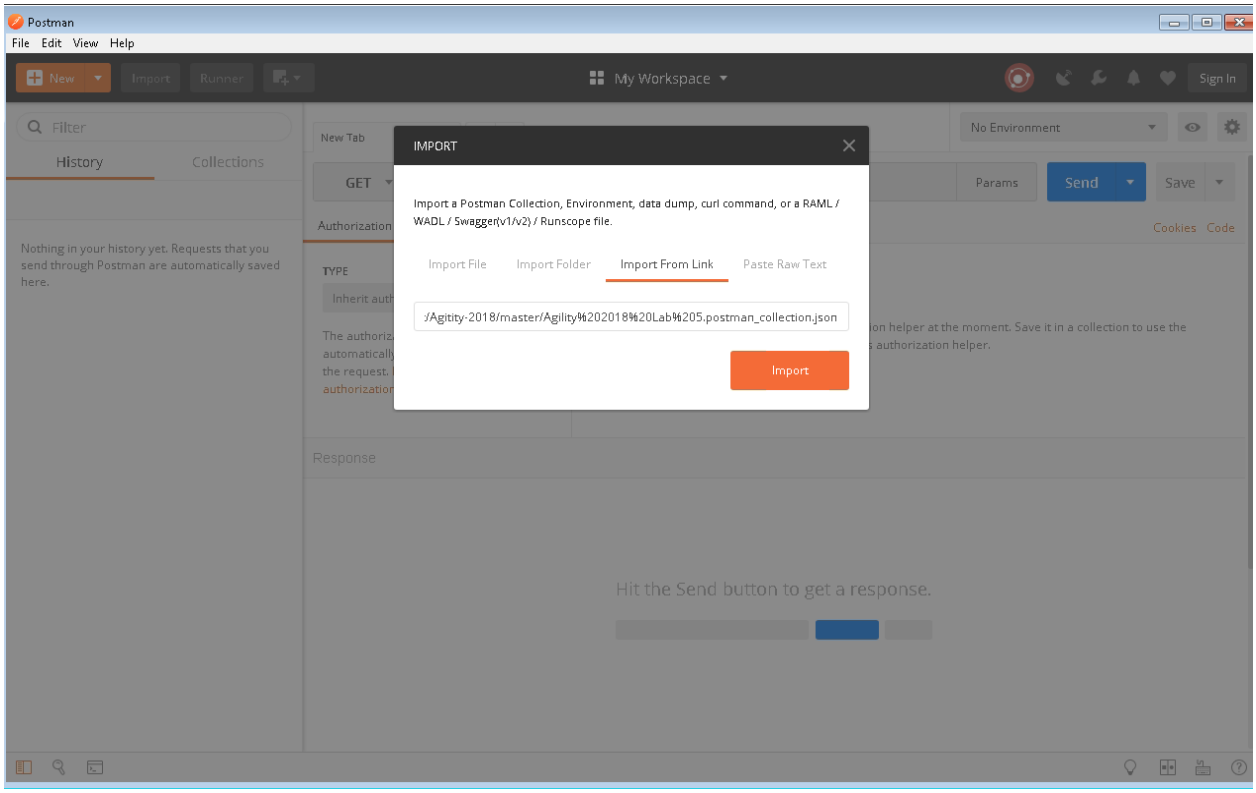
Postman helps you be more efficient while working with APIs. Postman is a scratch-your-own-itch project. The need for it arose while one of the developers was creating an API for his project. After looking around for a number of tools, nothing felt just right. The primary features added initially were a history of sent requests and collections. You can find Postman here - www.getpostman.com.

1.7.7 Simulating and defeating a Christmas Tree Packet Attack

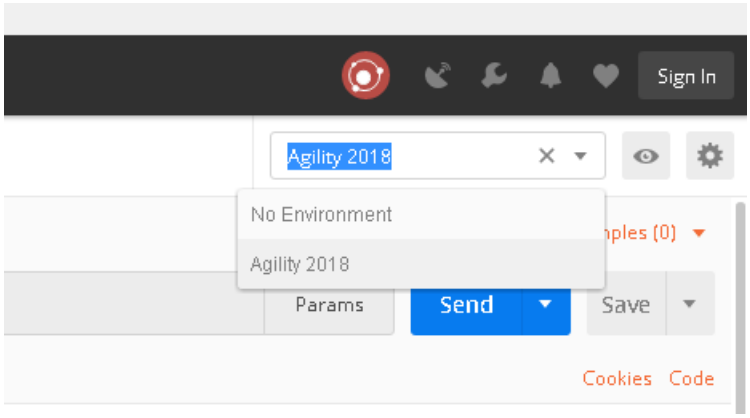
Now that we understand what REST is let's use it to defeat Joanna one last time. Joanna was feeling festive for her final attack. In this example, we'll set the BIG-IP to detect and mitigate Joanna's attack where all flags on a TCP packet are set. This is commonly referred to as a Christmas tree packet and is intended to increase processing on in-path network devices and end hosts to the target.

To interact with the REST API, we'll be using POSTMan. We'll then use the hping utility to send 25,000 packets to our server, with random source IPs to simulate a DDoS attack where multiple hosts are attacking our server. We'll set the SYN, ACK, FIN, RST, URG, PUSH, Xmas and Ymas TCP flags.

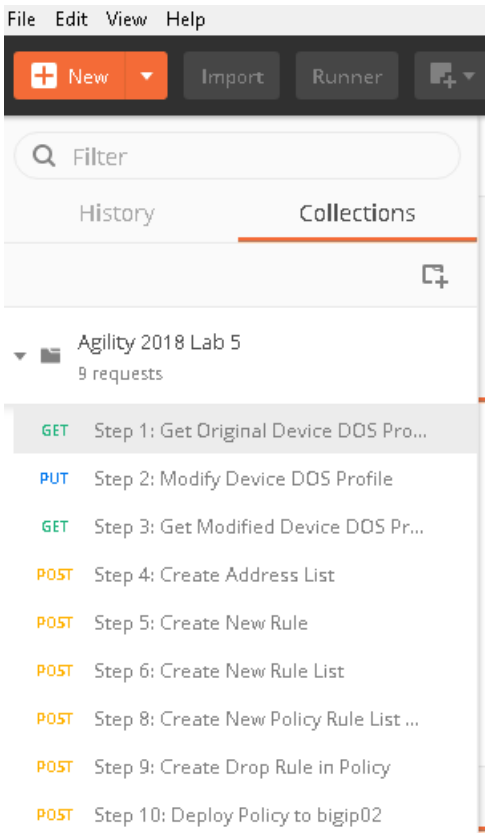
1. POSTMan is installed as an application and can be accessed from the desktop of the Jumpbox
2. Once you launch POSTMan You'll then want to import the API calls for the lab as well as the environment variables
    - There is a notepad on the desktop labeled "Postman Links"
    - Within POSTman and click on the "Import" link near the top and then select "Import from Link"
    - Copy and paste the **collection** link from within the notepad and select "Import"
    - Copy and paste the **environment** link from within the notepad and select "Import"



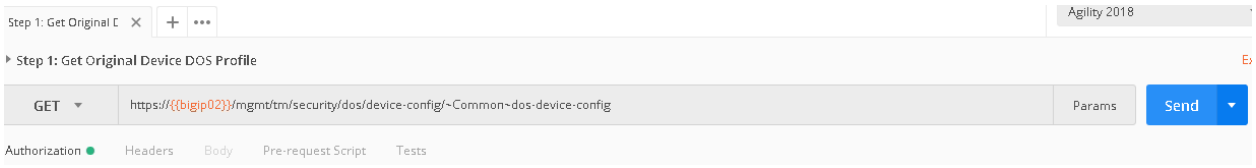
3. Before proceeding verify the Agility 2018 environment is selected from the drop down in the top right of POSTman



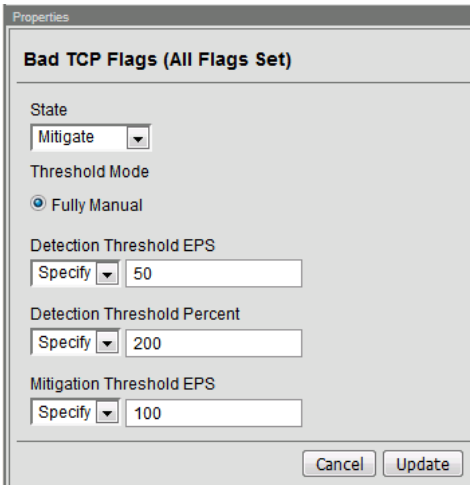
4. In the bigip01.dnstest.lab (https://192.168.1.100) web UI, navigate to Security > DoS Protection > Device Configuration > Network Security.
5. Expand the **Bad-Header-TCP** category in the vectors list.
6. Click on the **Bad TCP Flags (All Flags Set)** vector name and take note of the current settings
7. Within POSTman open the collection “Agility 2018 Lab 5”



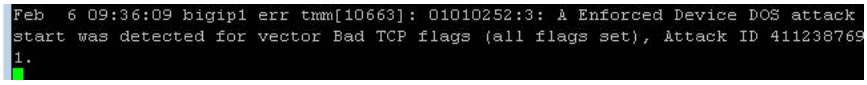
8. Run step 1 by clicking on the send button to the right



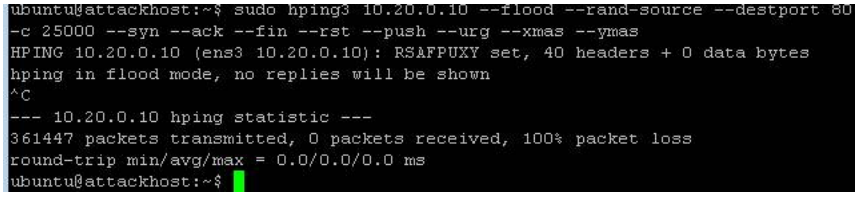
9. The output from the GET request can be reviewed, this is showing you all the device-dos configuration options and settings. Search for "bad-tcp-flags-all-set" by clicking 'ctrl +f'. Note the values as they are currently configured. We are now going to modify the Bad TCP Flags (All Flags Set) attack vector. To do so run step 2 of the collection by highlighting the collection and click "Send".
10. You can now execute step 3 in the collection and verify the changes, you can also verify the changes in the BIG-IP web UI.



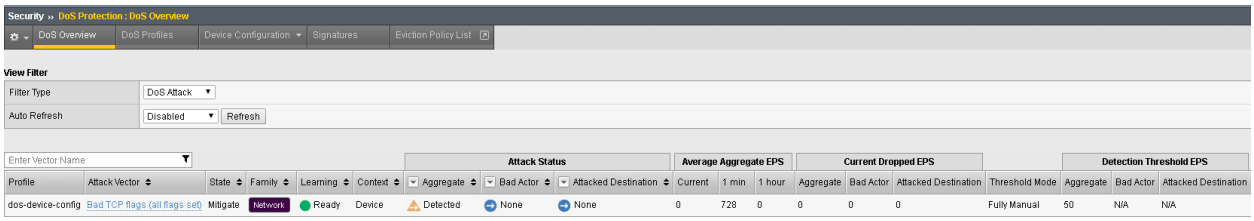
11. Open the BIG-IP SSH session and scroll the ltm log in real time with the following command: tail -f /var/log/ltm
12. On the attack host, launch the attack by issuing the following command on the BASH prompt:
    `sudo hping3 10.20.0.10 --flood --rand-source --destport 80 -c 25000 --syn --ack --fin --rst --push --urg --xmas --ymas`
13. You'll see the BIG-IP ltm log show that the attack has been detected:



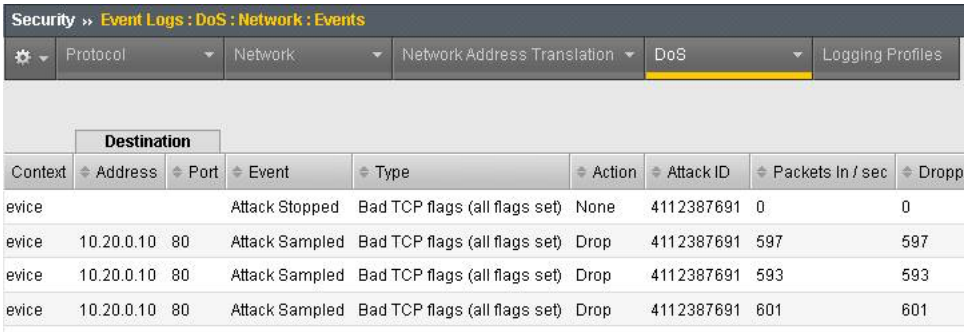
14. After approximately 60 seconds, press **CTRL+C** to stop the attack.



15. Navigate to **Security** > **DoS Protection**> **DoS Overview (you may need to refresh or set the auto refresh to 10 seconds).** You'll notice from here you can see all the details of the active attacks. You can also modify an attack vector right from this screen by clicking on the attack vector and modifying the fly out.



16. Return to the BIG-IP web UI. Navigate to **Security** > **Event Logs** > **DoS** > **Network** > **Events**. Observe the log entries showing the details surrounding the attack detection and mitigation.



17. Navigate to **Security** > **Reporting** > **DoS** > **Analysis**. Single-click on the attack ID in the filter list to the right of the charts and observe the various statistics around the attack.

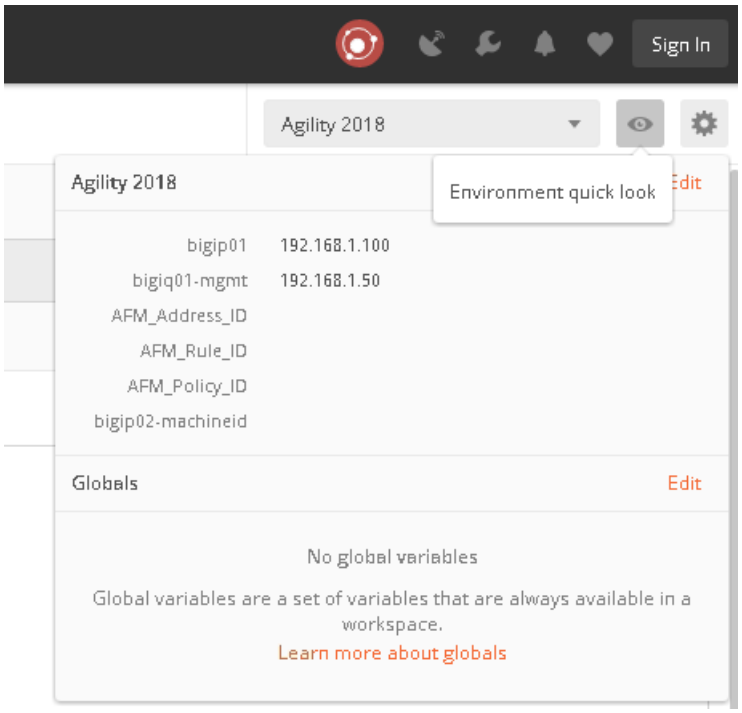
18. The same attacks can also be seen in BIG-IQ as demonstrated in the previous lab.

Congratulations, you have successfully defeated Joanna's festive attack using only the REST API to configure the device!

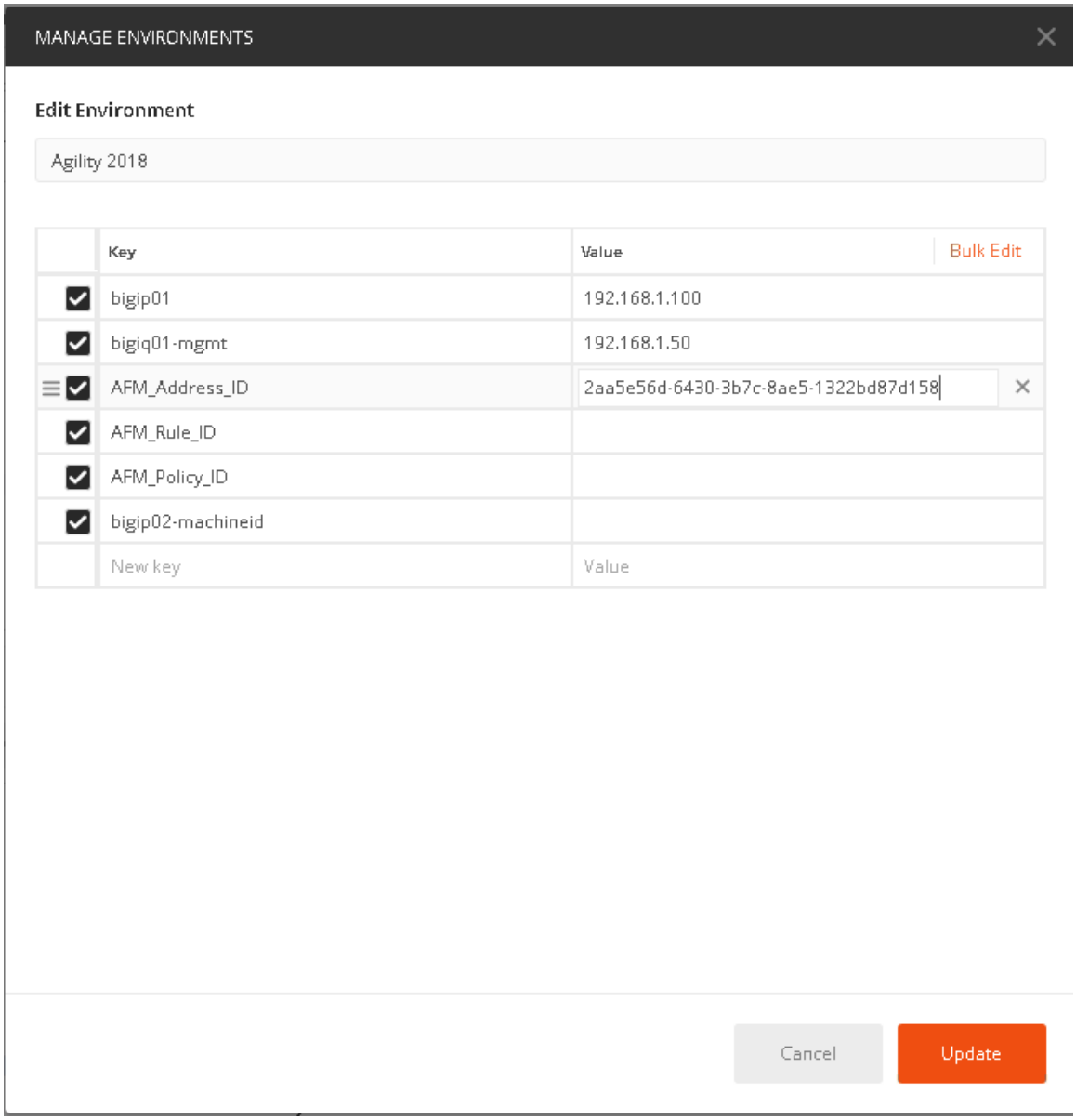
Since it's the end of the week and Joanna is using the same IP address continually, lets block her IP address and her subnet using BIG-IQ. We'll use the REST API to accomplish this as well, as BIG-IQ also has an available REST API.

1. Using POSTman run step 4, this will create an address-list within BIG-IQ, the advantage to address-lists is they allow you to group similar objects into a group. In this instance we're going to create an address-list named API_Naughty_Address_List with a host and a network. Once you run the command you'll receive output below. You will need to copy the value returned in the 'ID" field as shown below:

2. Take the copied text and paste it into the environment variable for AFM_Adddress_ID. The variables are accessed by clicking on the "eye" icon next to where you selected the Agility 2018 Environment:



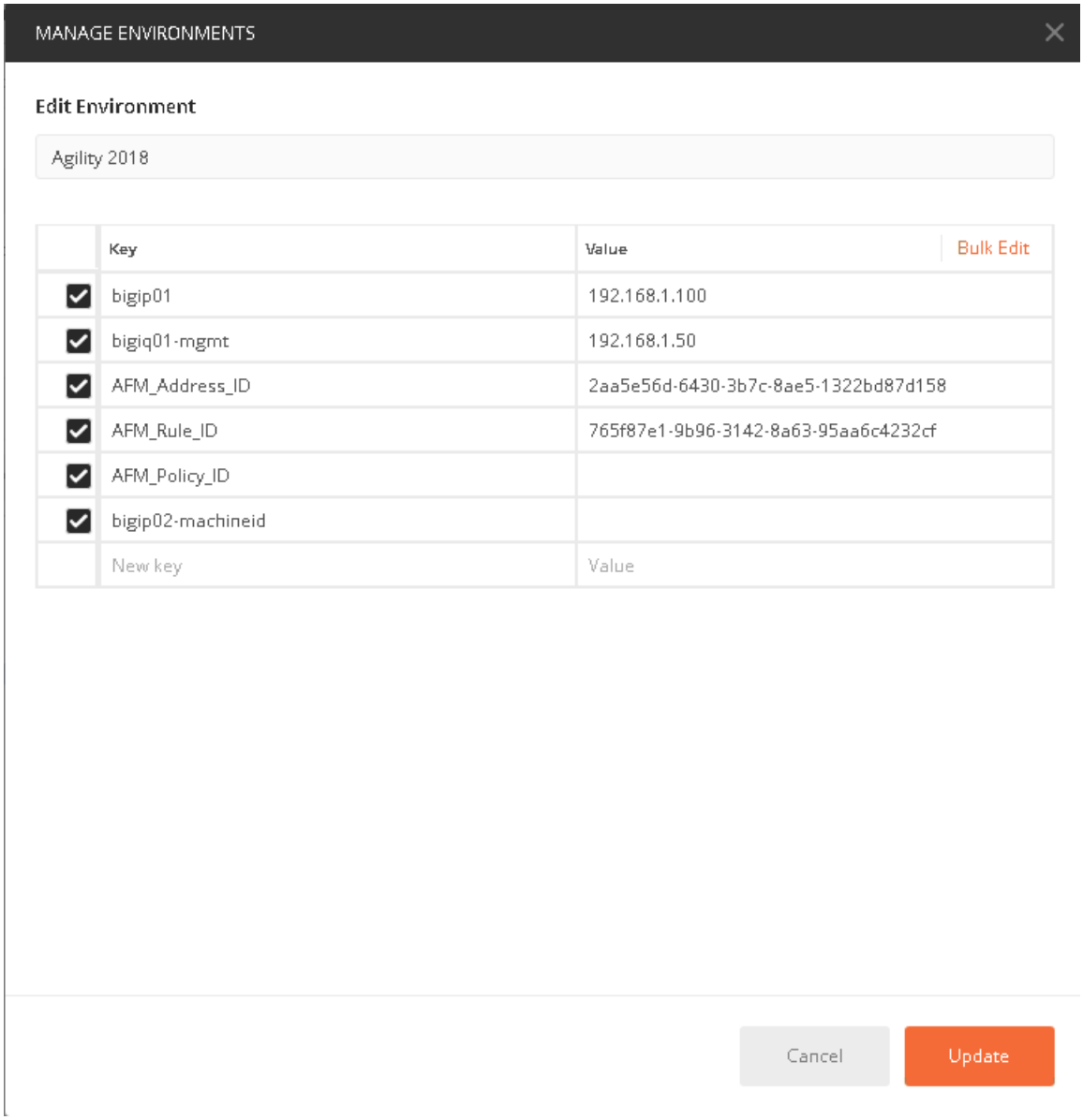
3. Click edit and enter the value returned in step 1, when completed click update



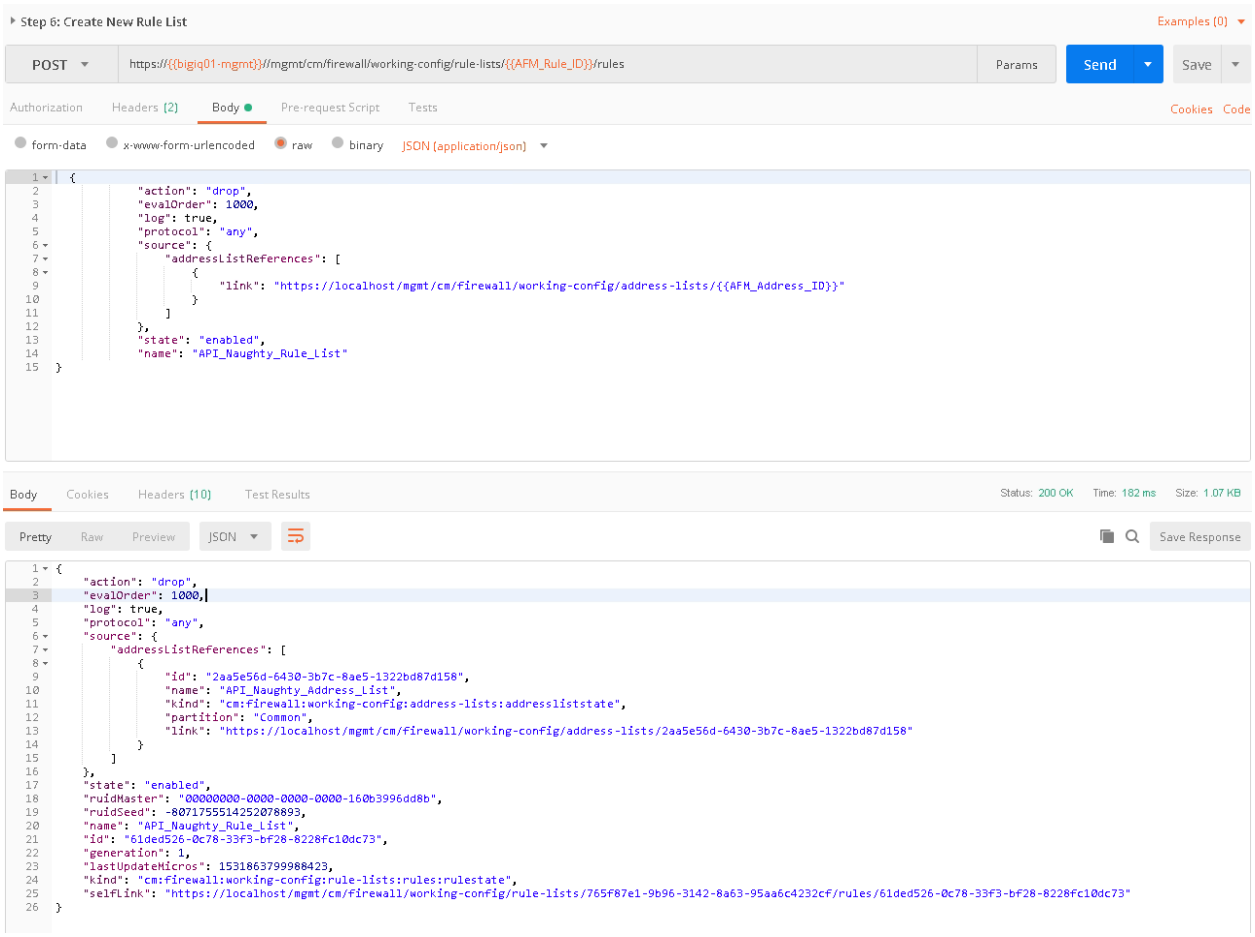
4. We will now create a rule list name first, to accomplish this send the call found in step 5. You will need to also capture the “ID” in this step as well. This value will be updated in the AFM_Rule_ID field


```
{
    "rulesCollectionReference": {
        "link": "https://localhost/mgmt/cm/firewall/working-config/rule-lists/765f87e1-9b96-3142-8a63-95aa6c4232cf/rules",
        "isSubcollection": true
    },
    "partition": "Common",
    "name": "API_Naughty_Rule_List",
    "id": "765f87e1-9b96-3142-8a63-95aa6c4232cf",
    "generation": 1,
    "lastUpdateMicros": 1531863428459874,
    "kind": "cm:firewall:working-config:rule-lists:ruleliststate",
    "selfLink": "https://localhost/mgmt/cm/firewall/working-config/rule-lists/765f87e1-9b96-3142-8a63-95aa6c4232cf"
}
```


5. Take the copied text and paste it into the environment variable for AFM_Rule_ID



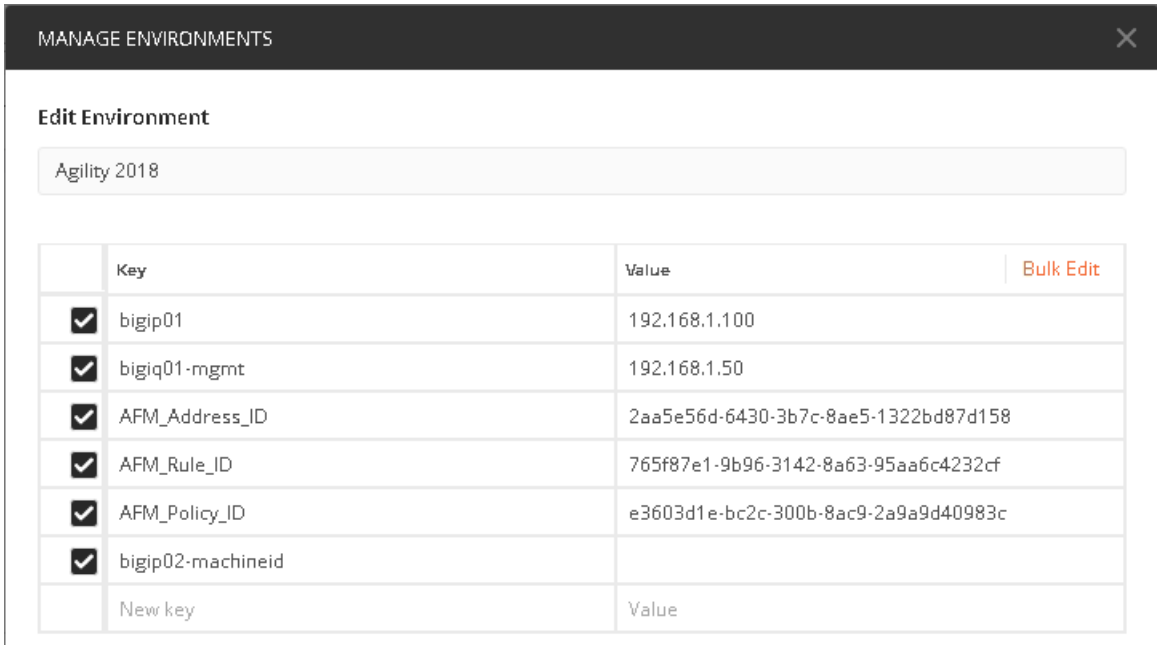
6. At this stage we have created an address-list with objects and saved the ID, we have also created a rule name and saved the ID. The next step is to add an actual rule to the newly created rule named “Naughty_Rule_List”. Before you send the call-in step 6, take a moment to examine the body of the request. You’ll notice in the URI we’re referencing the variable of AFM_Rule_ID and in the body of the JSON request we’re linking the AFM_Address_ID to the rule. Once sent you’ll receive confirmation similar to the below output.



7. Since this is an existing environment, we're going to first need to obtain the policy ID before we can assign the value to this variable. To obtain the policy ID of the existing policy we created in lab 1 and imported in the prior lab, run step 7.

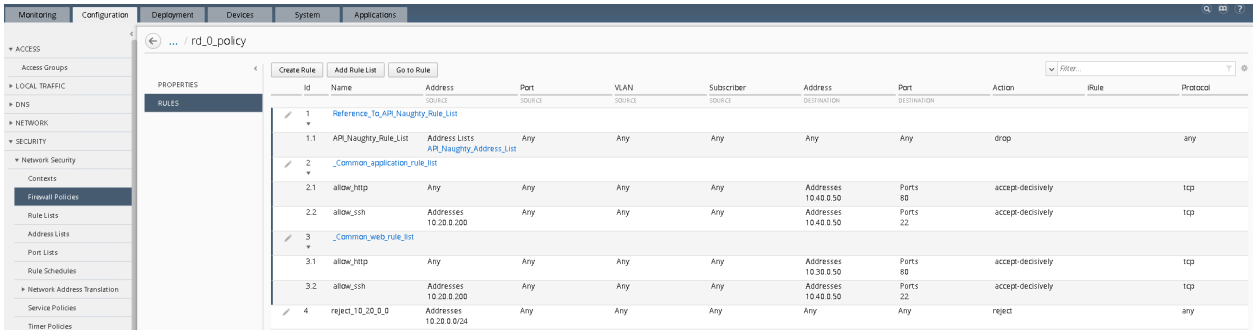
8. You will notice there are two policies, Global and rd_0_policy, we'll need to copy the ID for the

rd_0_policy which is located directly under its name and paste it into the variable for AFM_Policy_ID.



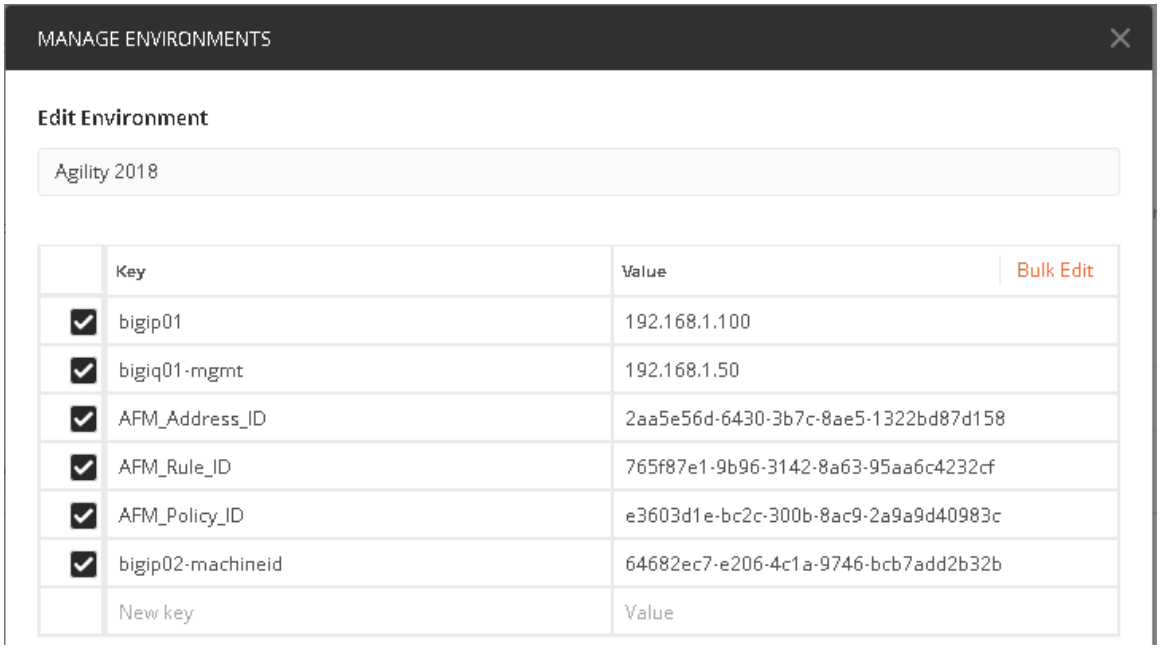
9. Finally run step 8 to add the new rule list to the existing policy, when completed you'll receive output similar as seen below.

10. Before we deploy the policy. Log into the BIG-IQ web UI (https://192.168.1.50) and navigate to Configuration Security Network Security Firewall Policies. Click on the link for the rd_0_policy, expand all the rules to verify your new API created rule list is first in the list and all objects are created as expected.

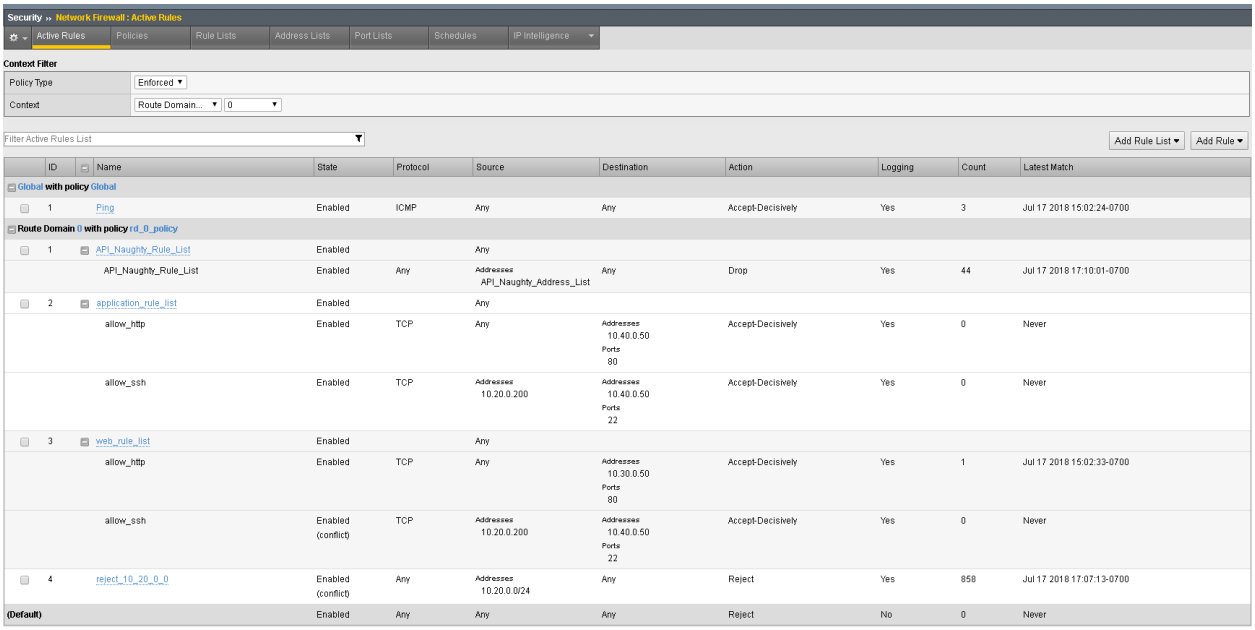


11. The final step is to deploy the policy to the BIG-IP. Before we can do this, we have one last variable we'll need to acquire, the machine ID of bigip02.dnslab.test. To obtain the machine ID run the call in step 9, once the call is run, you will look for the machineId key and copy the value to the environment

variable bigip02-machined as shown below and click update.

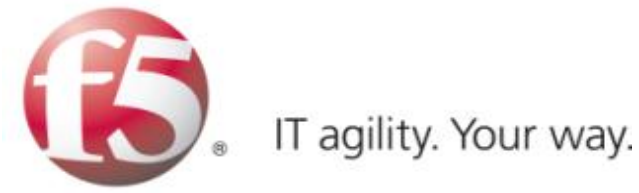


12. Finally, you will run step 10, this will initiate a deployment on BIG-IQ to deploy the changes to BIG-IP. Within BIG-IQ navigate to Deployment Evaluate & Deploy Network Security. At the bottom in the deployments section you'll notice an API Policy Deploy task. Feel free to click on the task to investigate the changes. Once the policy has deployed, log into the web UI of bigip02.dnstest.lab and navigate to Security network Firewall Active Rules. Change the context to Route Domain and select 0. Expand all of the rules to verify the rules have been deployed as expected. Your final screen should look something like the screen capture below.



Lastly, in your web browser, verify you can no longer access the web pages http://10.30.0.50 and http://10.40.0.50 as well as no longer being able to SSH to any of the devices.

Written for TMOS 13.1.0.1/BIG-IQ 6.0



2

Advanced Multi-Layer Firewall Protection

Firewall 320 – Advanced Multi-Layer Firewall Protection

Participant Hands-on Lab Guide

Last Updated: January 2 2020

Welcome to the F5 Agility 2018 Multilayer Firewall Implementations setup and hands-on exercise series.

The purpose of the Lab Setup and Configuration Guide is to walk you through the setup of F5 BIGIP to protect applications at multiple layers of the OSI stack hence providing Application Security Control. This in effect allows F5 BIG-IP to be multiple firewalls within a single platform.

Assumptions/Prerequisites: You have attended the AFM 101 lab sessions either this year or in previous years. Additionally this lab guide assumes that you understand LTM/TMOS basics and are comfortable with the process of creating Nodes, Pools, Virtual Servers, Profiles and Setting up logging and reporting.

There are three modules detailed in this document.

Module 1: F5 Multi-layer Firewall

Module 2: F5 Dynamic Firewall Rules With iRules LX

Module 3: AFM Protocol Inspection IPS

Lab Requirements:

- Remote Desktop Protocol (RDP) client utility
  - Windows: Built-in
  - Mac (Microsoft Client): https://itunes.apple.com/us/app/microsoft-remote-desktop/id715768417?mt=12
  - Mac (Open Source Client): http://sourceforge.net/projects/cord/files/cord/0.5.7/CoRD_0.5.7.zip/download
  - Unix/Linux (Source – Requires Compiling): http://www.rdesktop.org/

Note: You may use your webbrowser for console access if necessary but screen sizing may be affected.

Note: IP Filtering locks down connectivity to to the remote labs. If you are required to VPN into your corporate office to get Internet access, please determine your external IP address via https://www.whatismyip.com and provide an instructor with that information for your pod.

- Connectivity to the facility provided Internet service
- Unique destination IP address for RDP to your lab

2.1 Module 1: F5 Multi-layer Firewall

This module has seven labs in configuring an Advanced Multi-layer firewall applicable to many data center environments.

In this module, you will build a perimeter firewall with advanced Layer 7 security mitigations.

Estimated completion time: 1 hour

Objective:

- Inspect multiple internal pools and virtual servers for different applications within your data center. e.g. www, API, /downloads
- Inspect external hosted virtual server that allows the same IP address to be shared with multiple SSL enabled applications.
- Inspect and understand LTM policy to direct traffic to appropriate virtual server
- Configure local logging; test
- Create a network firewall policy to protect the internal application virtual servers; test
- Configure the external virtual server to tranform traffic coming through CDN networks so that firewall policies can be applied to specific clients; test
- Modify the network firewall policy to block based on XFF; test
- Apply Layer 7 responses (403 Denied) for CDN clients to firewall drop rules
- Configure HTTP protocol security; test
- Configure SSL Visibility to external security devices e.g. IDS; test

Labs 1 & 2 highlight the flexibility of leveraging an application proxy such as the BIG-IP for your perimeter security utilizing common traffic management techniques and some additional features unique to the BIG-IP as an Application Delivery Controller.

Labs 3 & 4 Breaks out applying differing security policies to the multi-tiered application deployment.

Lab 5 Highlights the flexibility of the Multi-Layered Firewall to solve common problems for hosting providers.

Lab 6 Applies Layer 7 protocol validation and security for HTTP to the existing applications.

Lab 7 Provides a solution for sending decrypted traffic to other security devices.

Warning: IP addresses in screenshots are examples only. Please read the step-by-step lab instructions to ensure that you use the correct IP addresses.

2.1.1 Lab 1: Pre-configured pools and virtual servers

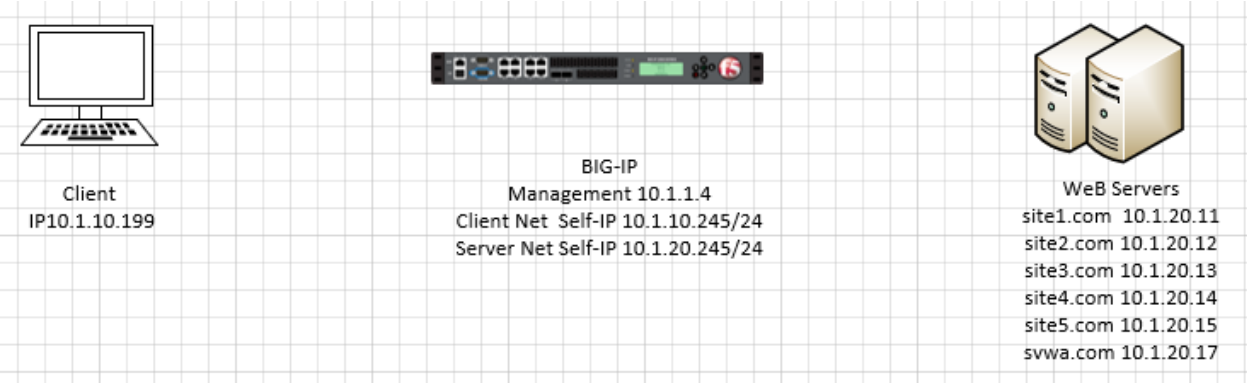
A virtual server is used by BIG-IP to identify specific types of traffic. Other objects such as profiles, policies, pools and iRules are applied to the virtual server to add features and functionality. In the context of security, since BIG-IP is a default-deny device, a virtual server is necessary to accept specific types of traffic.

The pool is a logical group of hosts that is applied to and will receive traffic from a virtual server.

On your personal device

Look at the supplemental login instructions for:

- External Hostnames
- External IP addressing diagram
- Login IDs and Passwords are subject to change as well.



Note: Use the Chrome Browser to Connect to BIG-IP01— https://10.1.1.4 Credentials are displayed in the login screen

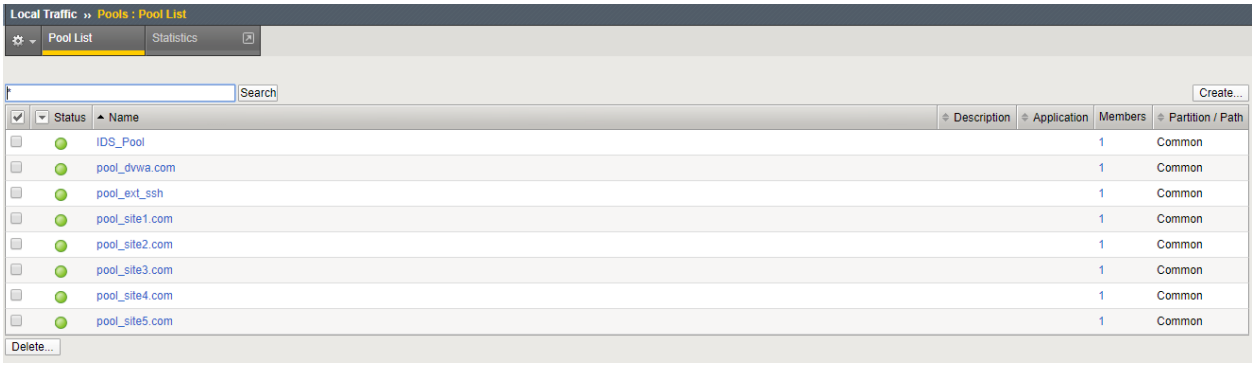
Inspect Application Pools

On BIG-IP

Verify the following pools using the following tabel of pool information.

Navigation: Local Traffic > Pools > Pool List

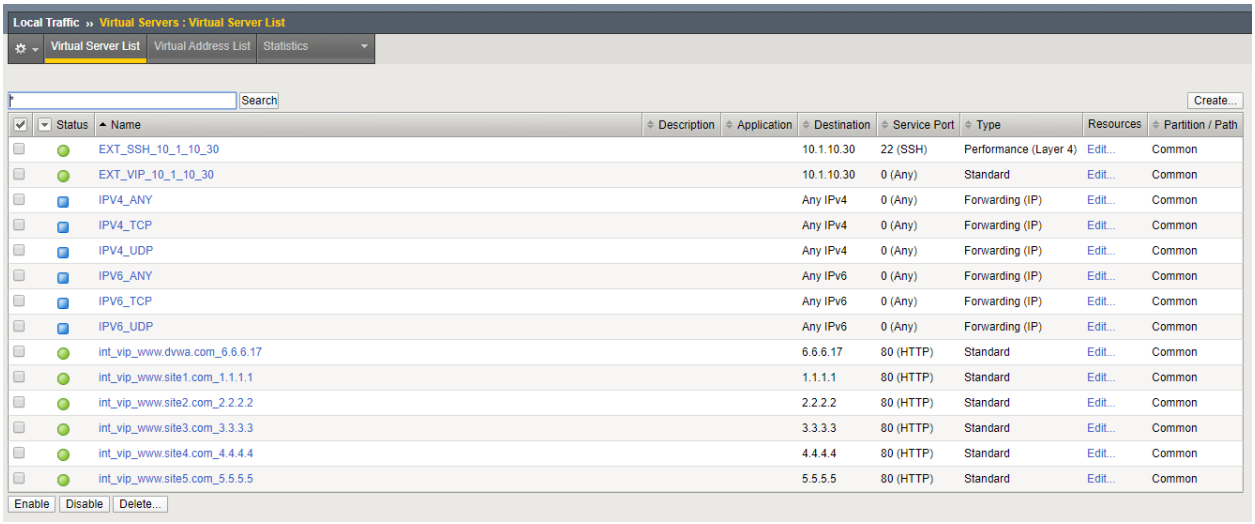
Name	Health Monitor	Members	Service Port
pool_www.site1.com	thttp	10.1.20.11	80
pool_www.site2.com	http	10.1.20.12	80
pool_www.site3.com	http	10.1.20.13	80
pool_www.site4.com	http	10.1.20.14	80
pool_www.site5.com	http	10.1.20.15	80
pool_www.dvwa.com	tcp_half_open	10.1.20.17	80



Inspect Application Virtual Servers

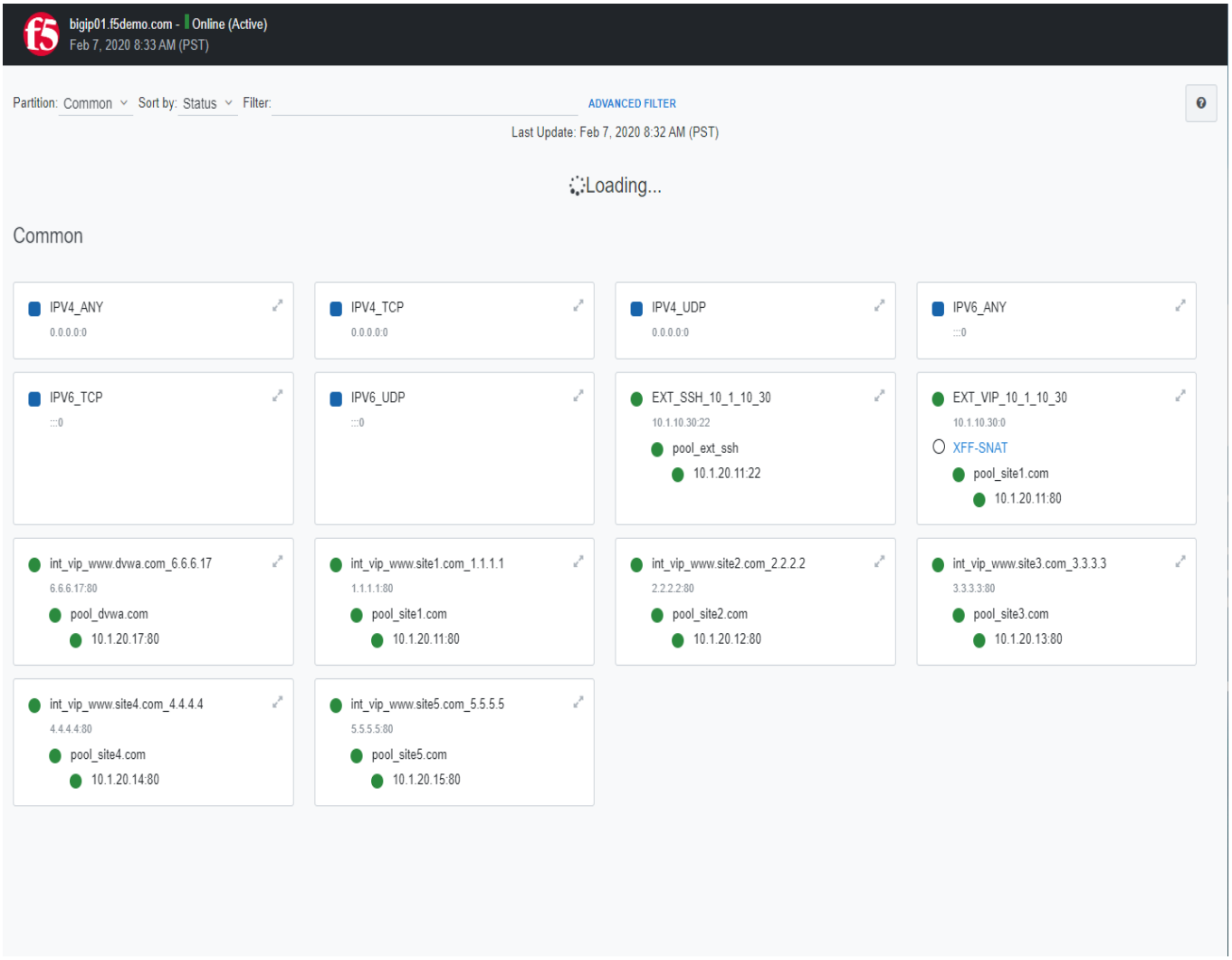
By using the term 'internal' we are creating the virtual servers on what is essentially a loopback VLAN which prevents them from being exposed. The EXT_VIP in this exercise is used to forward traffic with specific characteristics to the internal VIP's. This is accomplished by assigning a traffic policy to the VIP. The traffic policy is described and inspected in the next section. For this class, the Wildcard Virtual servers (Blue Square status indicator) are not used.

Navigation: Local Traffic > Virtual Servers > Virtual Server List



Inspect the Local Traffic Network Map

Navigation: Local Traffic > Network Map



Note: The virtual servers should show a green circle for status.

Note: This completes Module 1 - Lab 1

2.1.2 Lab 2: Leverage LTM Policies To Direct SSL Terminated Applications To Secondary Virtual Servers

What is SNI? Introduced in TLS 1.0 as a TLS extension, Server Name Indication (SNI) allows the client to send the hostname they are trying to connect to in the SSL handshake. This allows the Application Delivery Controllers (ADC) such as the BIG-IP and the Application servers to identify the appropriate application the client is trying to connect to. From this information, the ADC can respond with the proper SSL certificate to the client allowing the ADC to provide SSL enabled services for multiple applications from a single IP address.

LTM policies are another way to programatically modify traffic as it is flowing through the data plane of the BIG-IP. This functionality can also be accomplished with F5 iRules. The advantage this has over iRules is

that LTM policies can be modified and appended to the existing configuration without replacing the entire application configuration. This lends itself to being updated through the CLI or via the REST API easily.

If you make a single change to an iRule, the entire iRule needs to be re-uploaded and applied.

The LTM policy is what directs application traffic to flow from the external virtual server to the internal virtual servers based on the Layer 7 request. In this case, since we are using SNI to terminate multiple applications (mysite,yoursite,theirsite, api, downloads) we need to be able to direct that traffic to the appropriate application pools. Some can even come back to the same application pool.

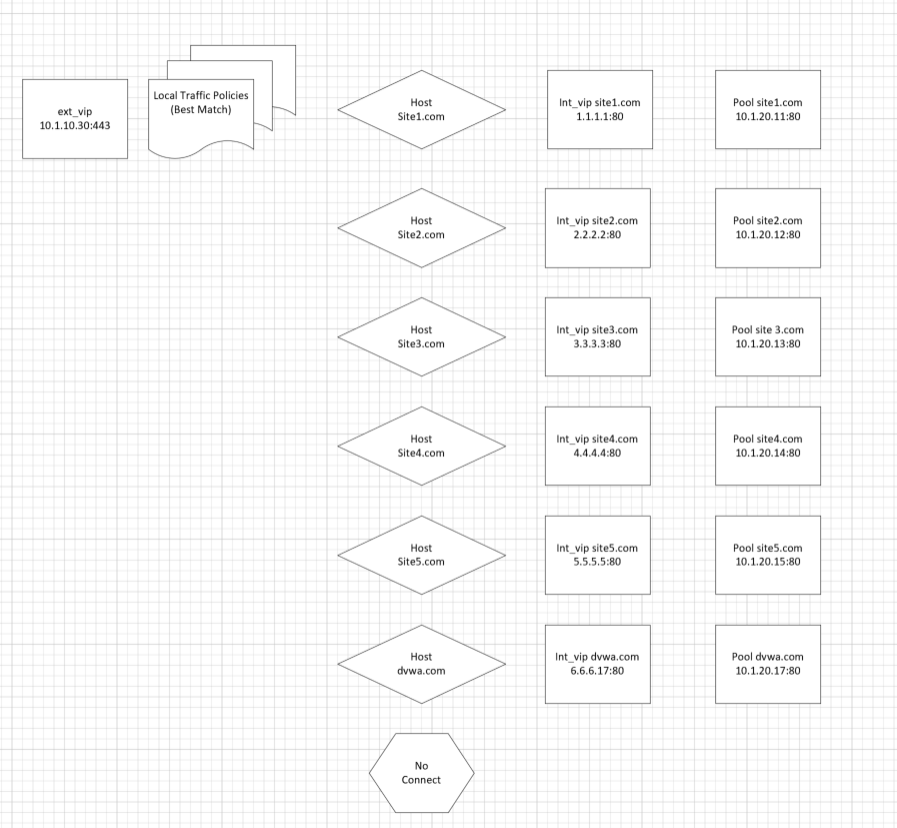
Whether it is based on the hostname or the URI path, the request can be forwarded to a different virtual server or an application pool of servers.

Inspect the LTM Policies

Take a few minutes to open the draft policy and review the iptions. Policy is a very flexible tool to direct traffic based on the packet content. In this use case we distribute traffic to a subset of internal VIP's, Policy can be configured to forward traffic directly to pools or nodes based on the packet content and many other attributes

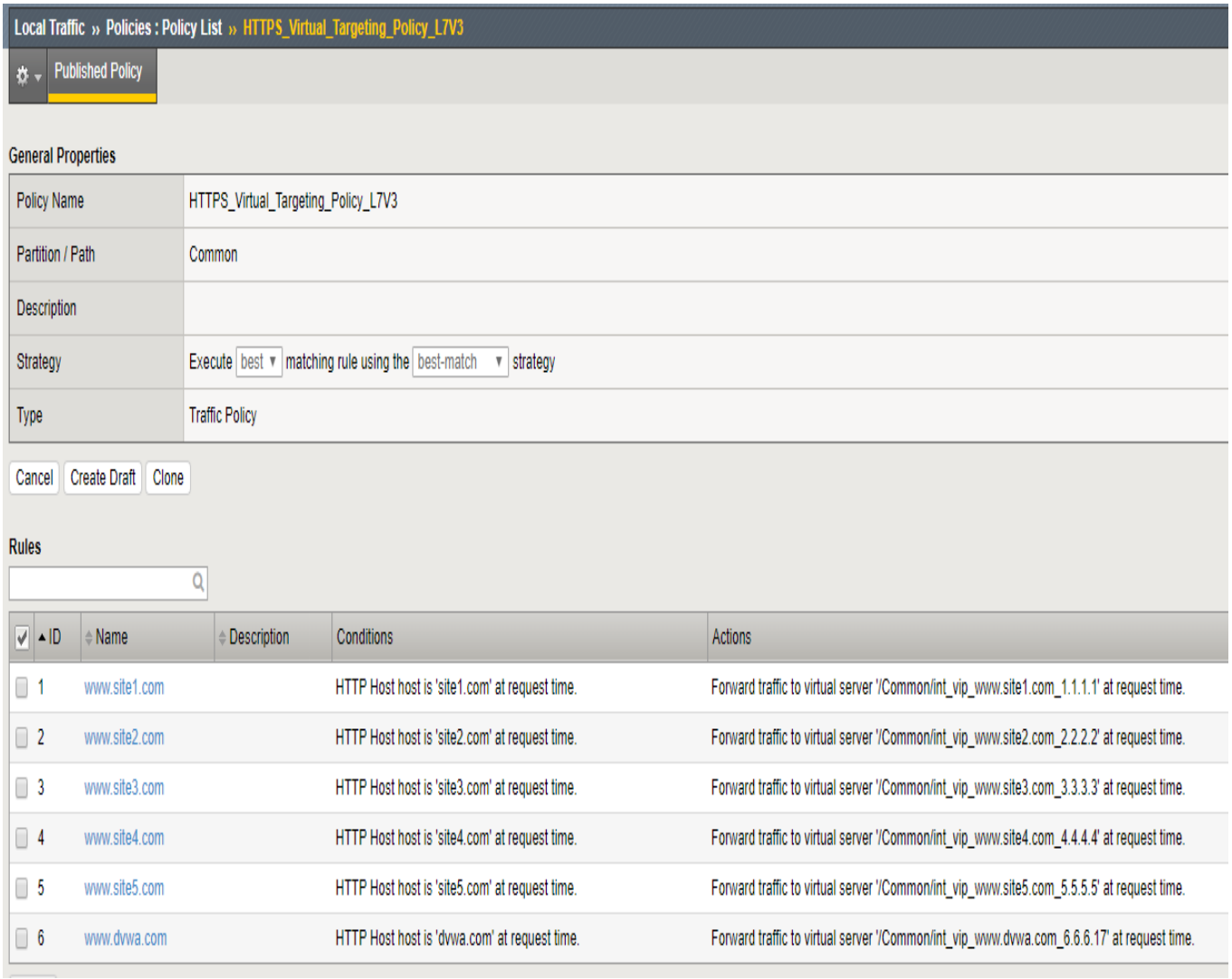
---

Note: As shown in this diagram, there is an external VIP and internal VIPs. The external VIP has the local traffic policies on it.



Navigation: Local Traffic > Policies : Policy List

Navigation: Select HTTPS_Virtual_Targeting_Policy_L7V3 from the published policies

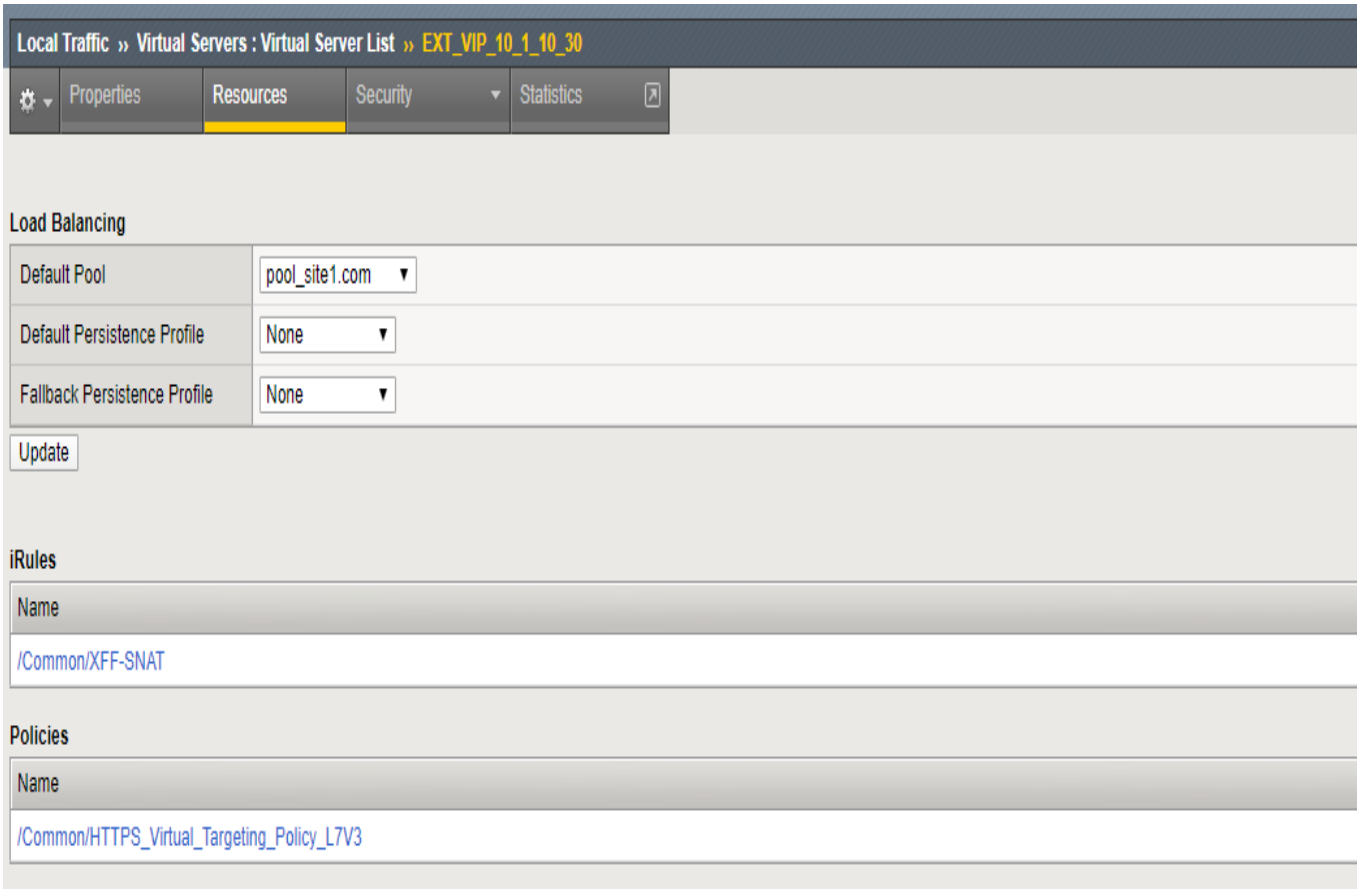


Verify that the Policy is assigned To The External Virtual Server

Navigation: Local Traffic > Virtual Servers : Virtual Server List

Navigation: Click the EXT_VIP_10_1_10_30

Navigation: Click the Resources Tab



Note: there is a policy and an iRule is assigned to the VIP:

Create An ACL to allow web traffic and SSH

The rules created in this section allow basic connectivity to the resources. We will add enforcement rules at the Virtual server level to demostrate functionality

On bigip01.f5demo.com (10.1.1.4) create a rule list to allow traffic. A logical container will be created before the individual rules can be added. You will create a list with rules to allow port 80 (HTTP), 443 (HTTPS), and 22 (SSH) to servers 10.1.20.11 through 10.1.20.17 We will also create a rules which allows HTTPS and SSH traffic to access 10.1.10.30

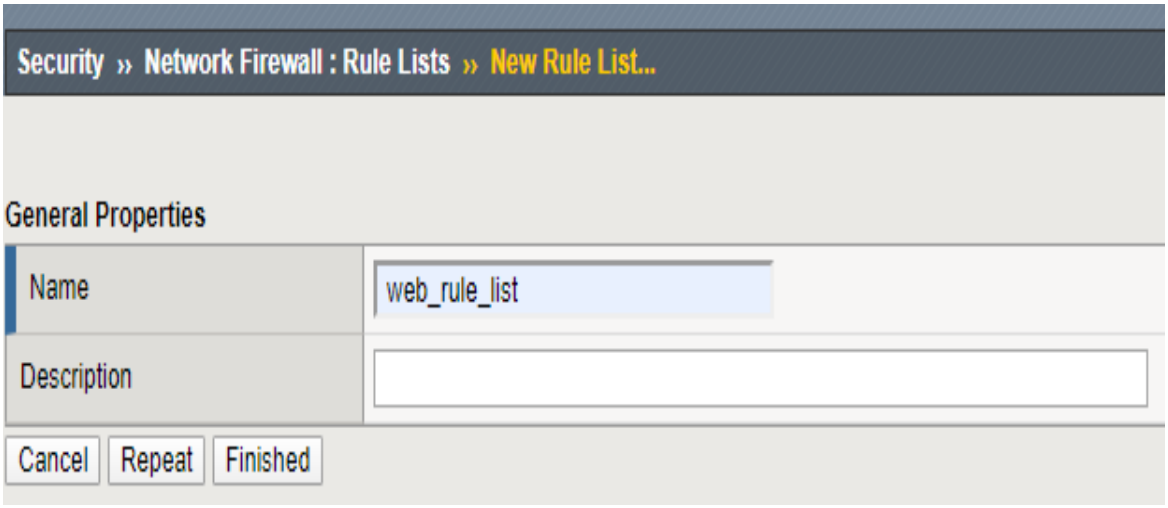
Create a container for the rules by going to:

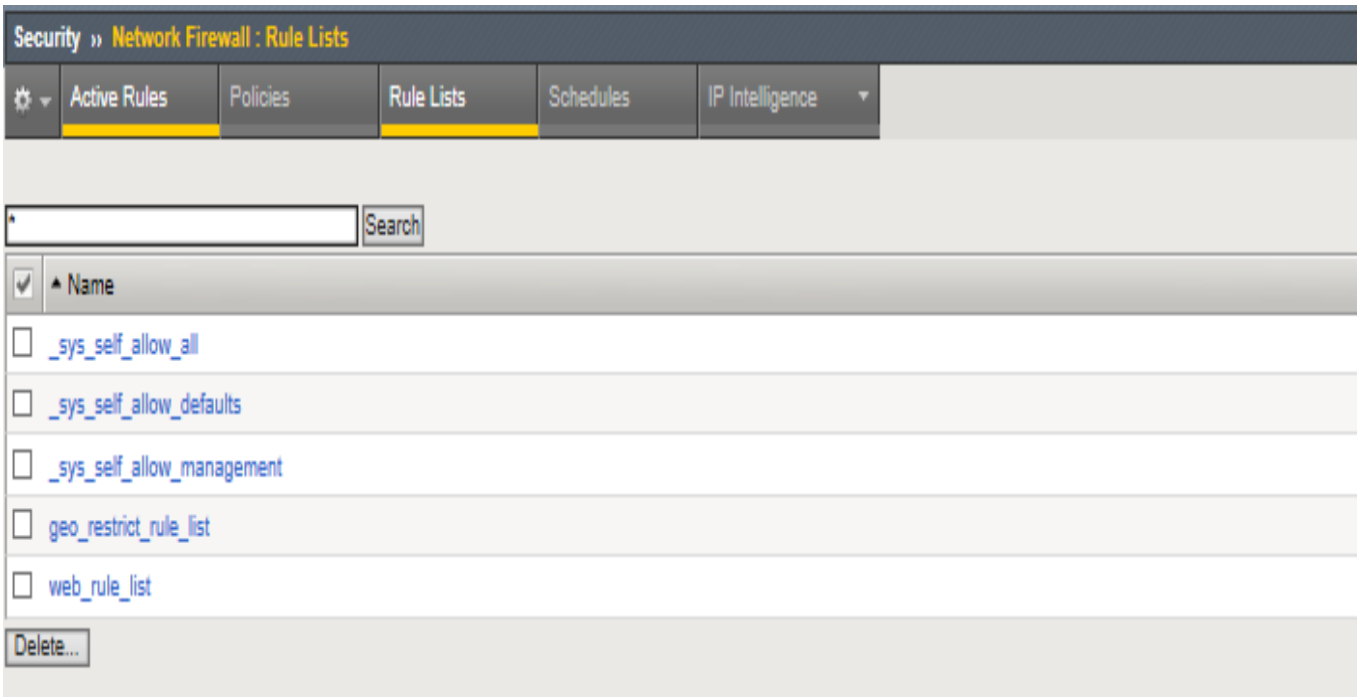
Navigation: Security > Network Firewall > Rule Lists

Navigation: select Create

For the **Name** enter **web_rule_list**, provide an optional description

Navigation click **Finished**

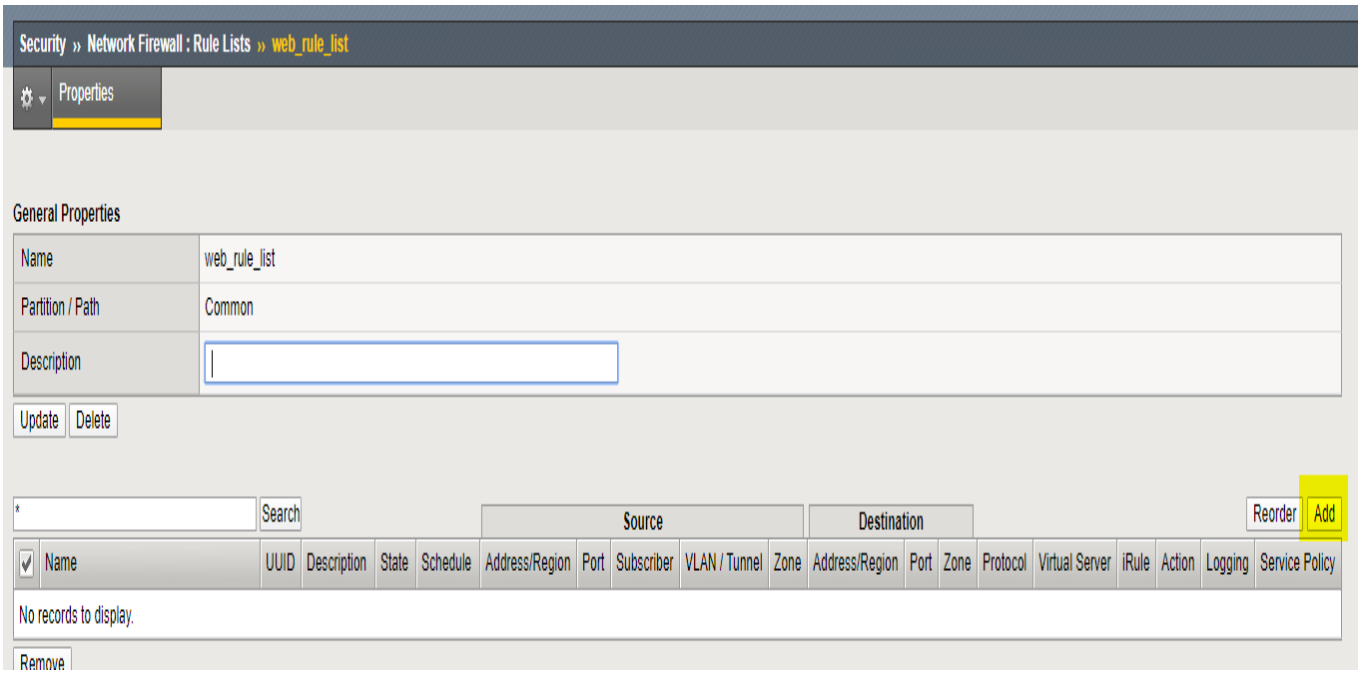




Navigation Select the **web_rule_list** by clicking on it in the Rule Lists table

Navigation click the **Add** button in the Rules section.

Add a rules into the list to allow HTTP, HTTPS, and SSH traffic as described in the next steps



Name	allow_http_and_https
Protocol	TCP
Source	Leave at Default of **Any**
Destination Address	**Specify Address Range** 10.1.20.11 to 10.1.20.17, then click **Add**
Destination Port	**Specify...** Port **80**, then click **Add Specify...** Port **443**, then click **Add**
Action	**Accept**
Logging	Enabled

Navigation: Click Repeat

Add a rule into the list to allow HTTPS to Virtual Server 10_1_10_30.

Navigation: Click **Finished**

Navigation: Click Finished

Assign the Rule List to a Policy

Navigation: Security > Network Firewall > Policies

Navigation Click Create

For the **Name** enter **rd_0_policy**, provide an optional description

Navigation click **Finished**.

(Note: We commonly use “RD” in our rules to help reference the “Route Domain”, default is 0)**

_static/class2/image273.png

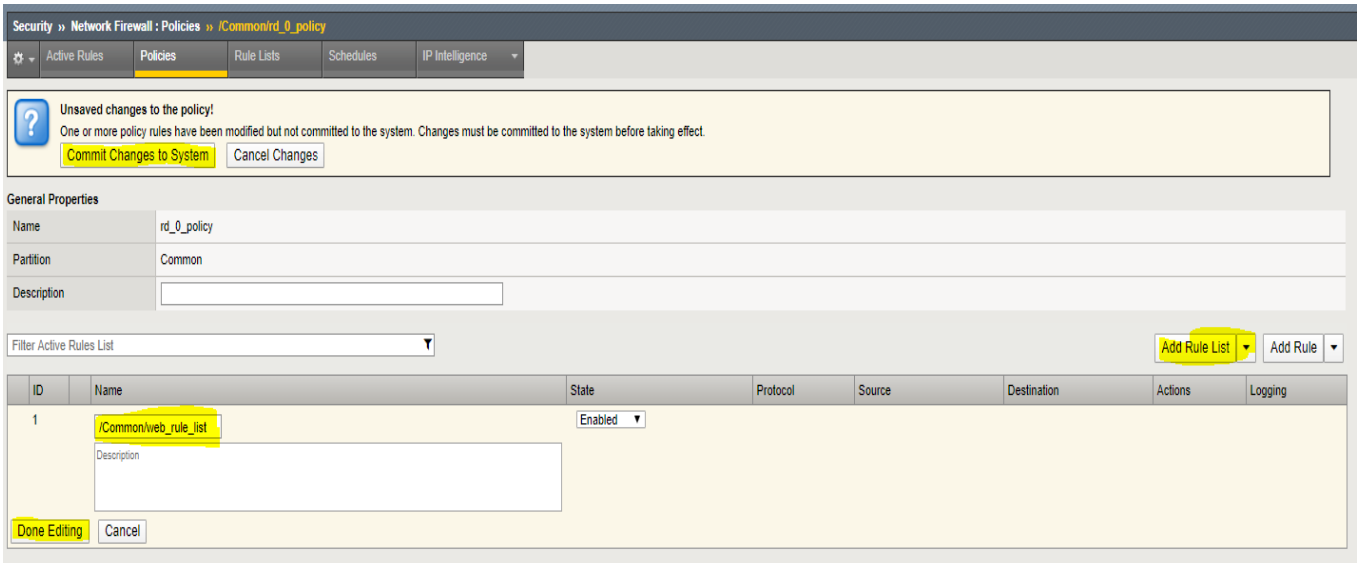
Navigation Edit the **rd_0_policy** by clicking on it in the Policy Lists table,

Navigation click the **Add Rule List** button.

Navigation For the **Name,** start typing **web_rule_list**, you will notice the name will auto complete,

Navigation select the rule list /**Common**/**web_rule_list**, provide an optional description

Navigation click **Done Editing.**



You will notice the changes are unsaved and need to be committed to the system. This is a nice feature to have enabled to verify you want to commit the changes you’ve just made without a change automatically being implemented.

Navigation click **“Commit** Changes **to System**

Assign the rd_0_policy to Route Domain 0

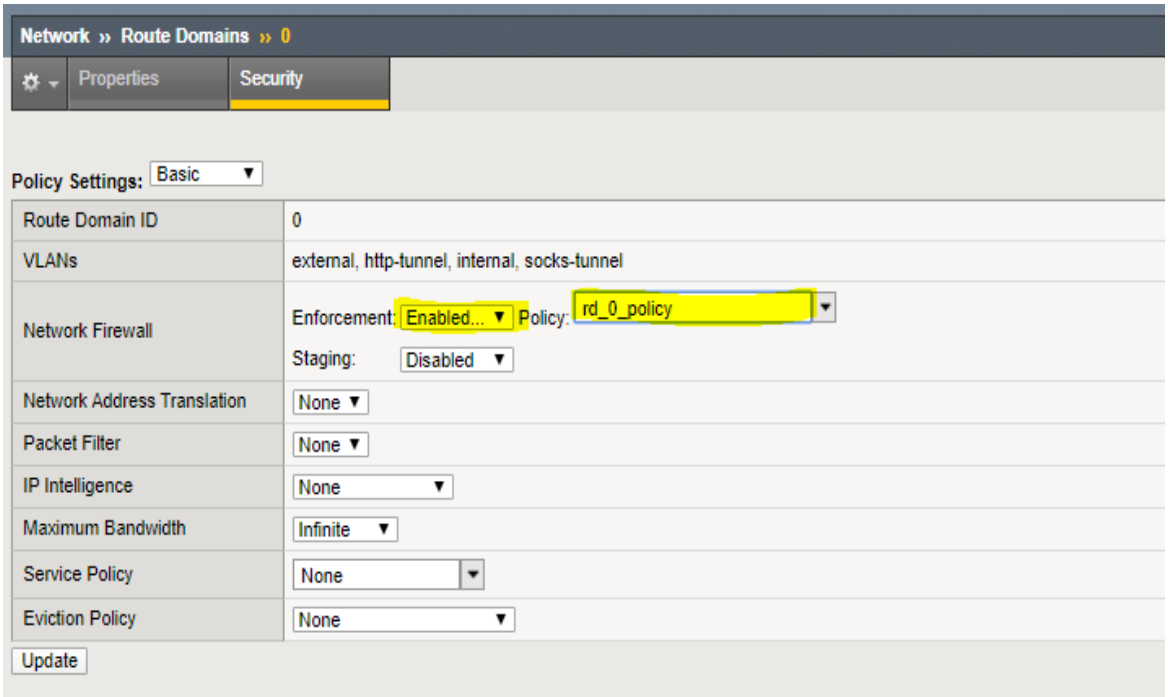
Navigation: Network > Route Domains

Navigation: Click on the “0” to select Route Domain 0

Navigation: Select the Security Tab

Set **Enforcement** to **Enable** and select the **rd_0_policy**

Navigation Click Update



Configure BIG-IP Firewall in ADC Mode

By default, AFM firewall is configured in ADC mode, which is a default allow configuration. In Firewall mode, all traffic is blocked at the firewall, and any traffic you want to allow must be explicitly specified.

In deployments where there are a large number of VIP's, deploying in Firewall mode would require significant preperation. Firewall functionality is easier to introduce in ADC mode.

Navigation: Security > Options > Network Firewall

Virtual Server & Self IP Contexts	Accept

Navigation Click **Update*

|image251|

Open the Firewall Options tab

Validate Lab 2 Configuration

Note: Open a tab on the Chrome Browser to test access to the URL's below

---

Validation: This lab is using self-signed certificates. You can either open a web browser on the test client or run CURL from the CLI to validate your configuration.

You will need to accept the certificate to proceed to the application sites

```
URL: https://site1.com

URL: https://site2.com

URL: https://site3.com

URL: https://site4.com

URL: https://site5.com

URL: https://dvwa.com    Username:  admin    Password: password
```

With curl you need to use the -k option to ignore certificate validation

Note: From a terminal window (use Cygwin on Win7 Client Desktop). Curl will let us do some of the additional testing in later sections. If you scroll up to the text immediately following the command you will see the IP address of the pool member you connected to.

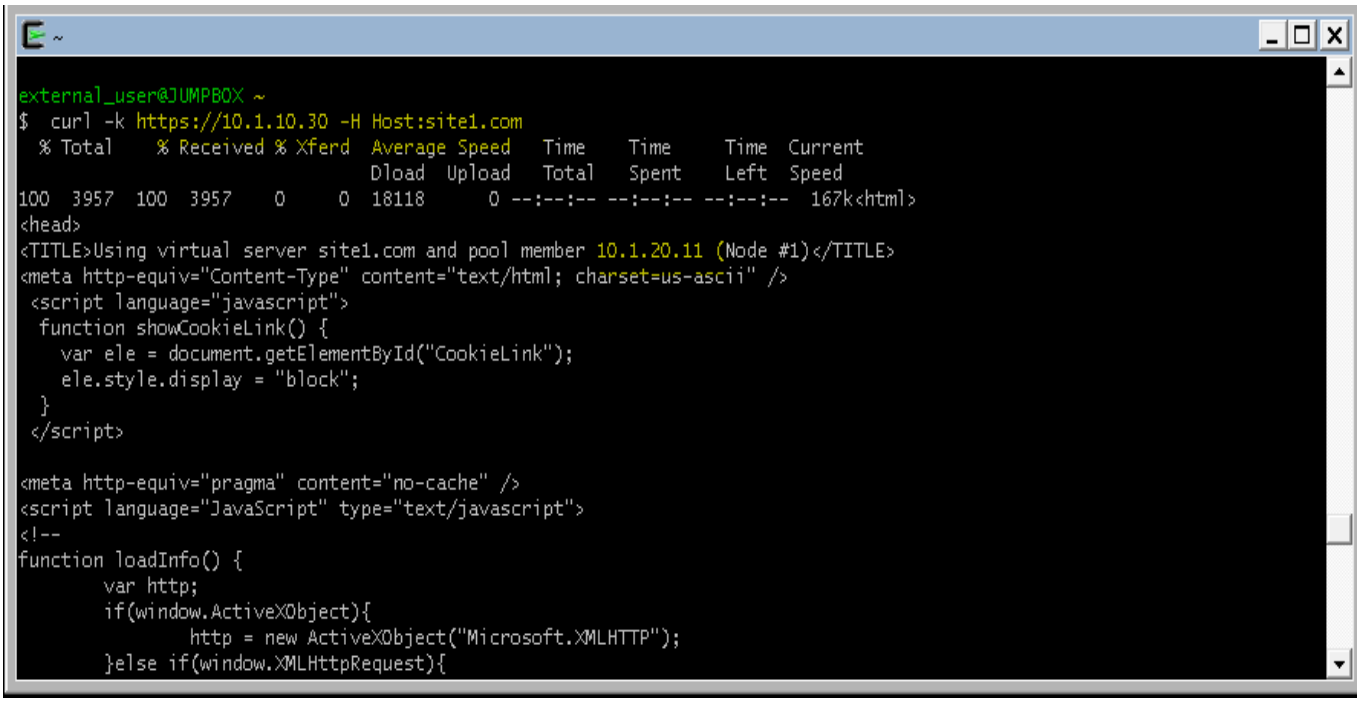
```
curl -k https://10.1.10.30 -H Host:site1.com

curl -k https://10.1.10.30 -H Host:site2.com

curl -k https://10.1.10.30 -H Host:site3.com

curl -k https://10.1.10.30 -H Host:site4.com

curl -k https://10.1.10.30 -H Host:site5.com
```



Note: for site 1 connected to 10.1.20.11, site 2 10.1.20.12 etc:

Note: This completes Module 1 - Lab 2:

2.1.3 Lab 3: Configure Local Logging For Firewall Events

Security logging needs to be configured separately from LTM logging.

High Speed Logging for modules such as the firewall module requires three componenets.

- A Log Publisher
- A Log Destination (local-db for this lab)
- A Log Profile

For more detailed information on logging please consult the BIG-IP documentation.

https://askf5.f5.com/kb/en-us/products/big-ip_ltm/manuals/product/bigip-external-monitoring-implementations-13-0-0/3.html

In this lab, we will configure a local log publisher and log profile. The log profile will then be applied to the virtual server and tested.

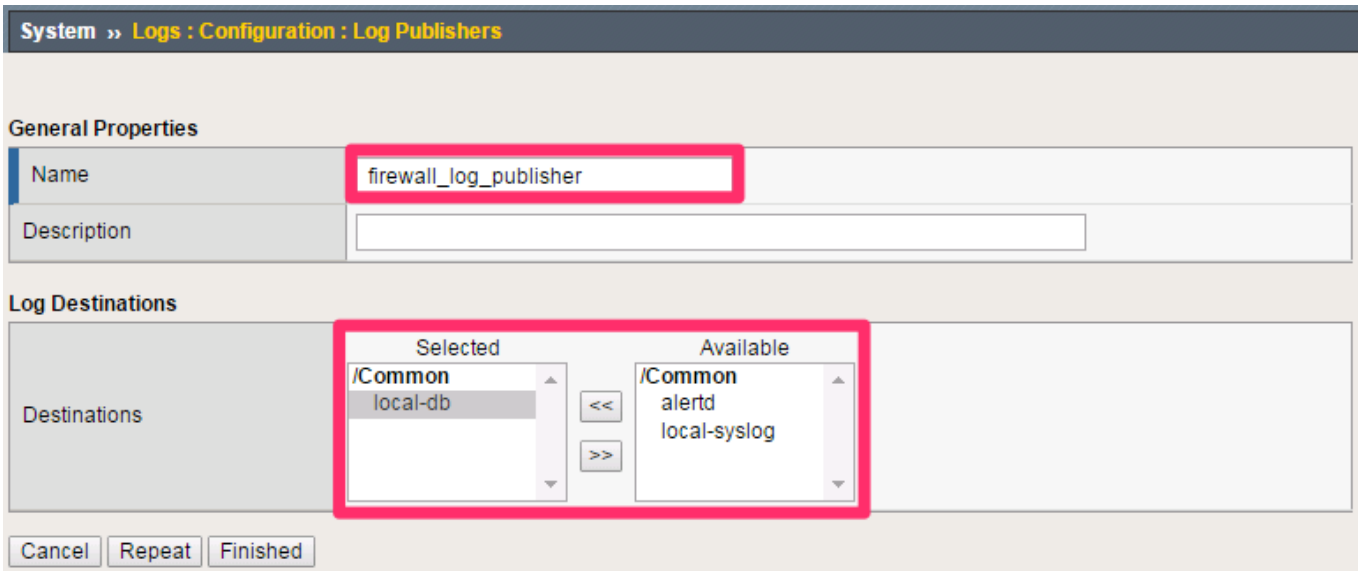
Create A Log Publisher

This will send the firewall logs to a local database

Create the log publisher using the following information:

Navigation: System > Logs > Configuration > Log Publishers, then click Create

Name	firewall_log_publisher
Destinations (Selected)	local-db



Note: Leave all other fields using the default values.

Navigation: Click Finished

Create A Log Profile

Create the log profile using the following information:

Navigation: Security > Event Logs > Logging Profiles, then click Create

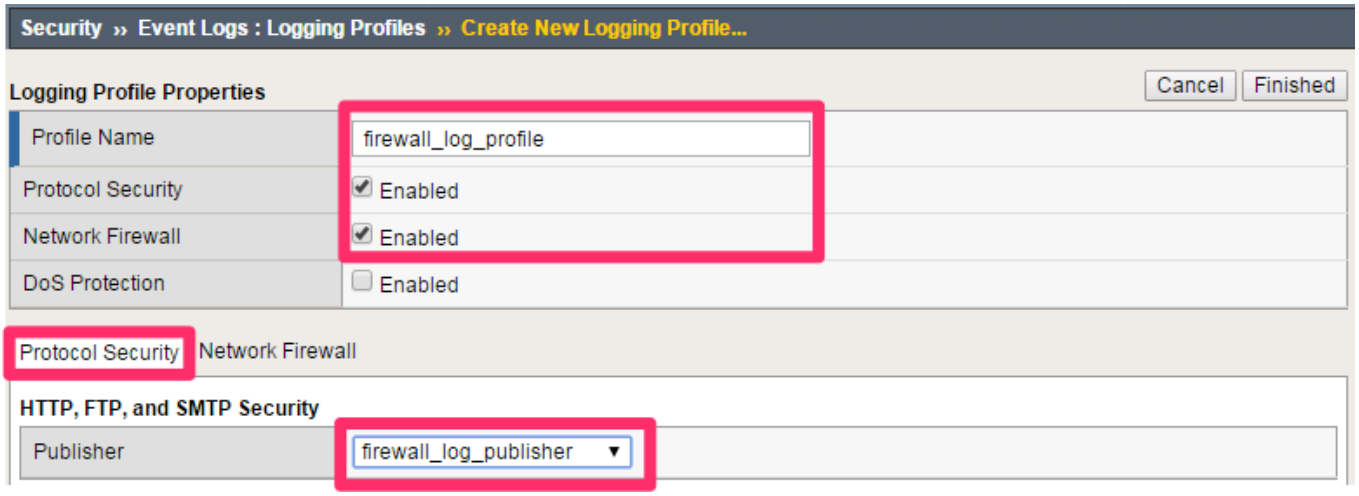
Name	firewall_log_profile
Protocol Security	Checked
Network Firewall	Checked

Modify The Log Profile To Collect Protocol Security Events

Edit log profile protocol security tab using the following information:

Navigation: Click on the Protocol Security tab and select the firewall_log_publisher

firewall_log_publisher



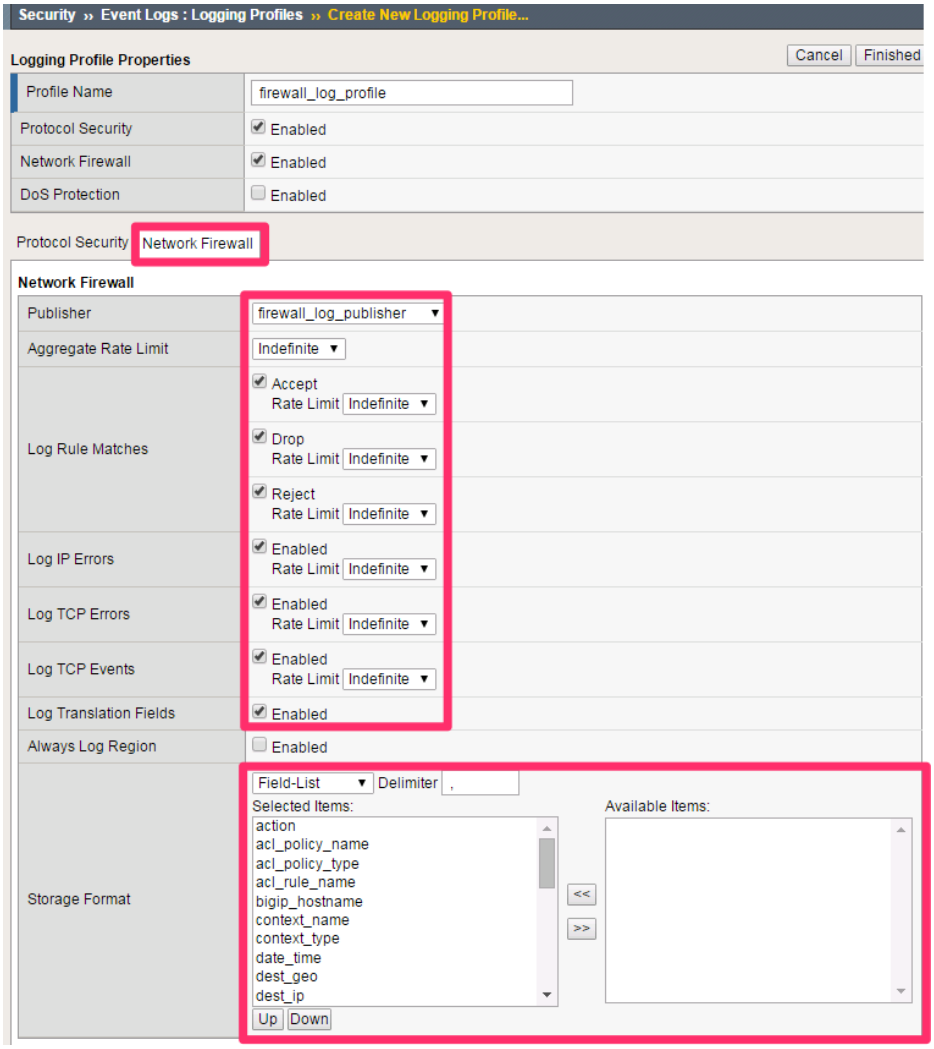
Note: Leave all other fields using the default values.

Modify The Log Profile To Collect Firewall Security Events

Edit log profile network firewall tab using the following information:

Navigation: Click on the Network Firewall tab

Network Firewall Publisher	firewall_log_profile
Log Rule Matches	Check Accept Check Drop Check Reject
Log IP Errors	Checked
Log TCP Errors	Checked
Log TCP Events	Checked
Log Translation Fields	Checked
Storage Format	Field-List (Move all to Selected Items)



Note: Leave all other fields using the default values.

Navigation: Click Create

Apply The Logging Configuration

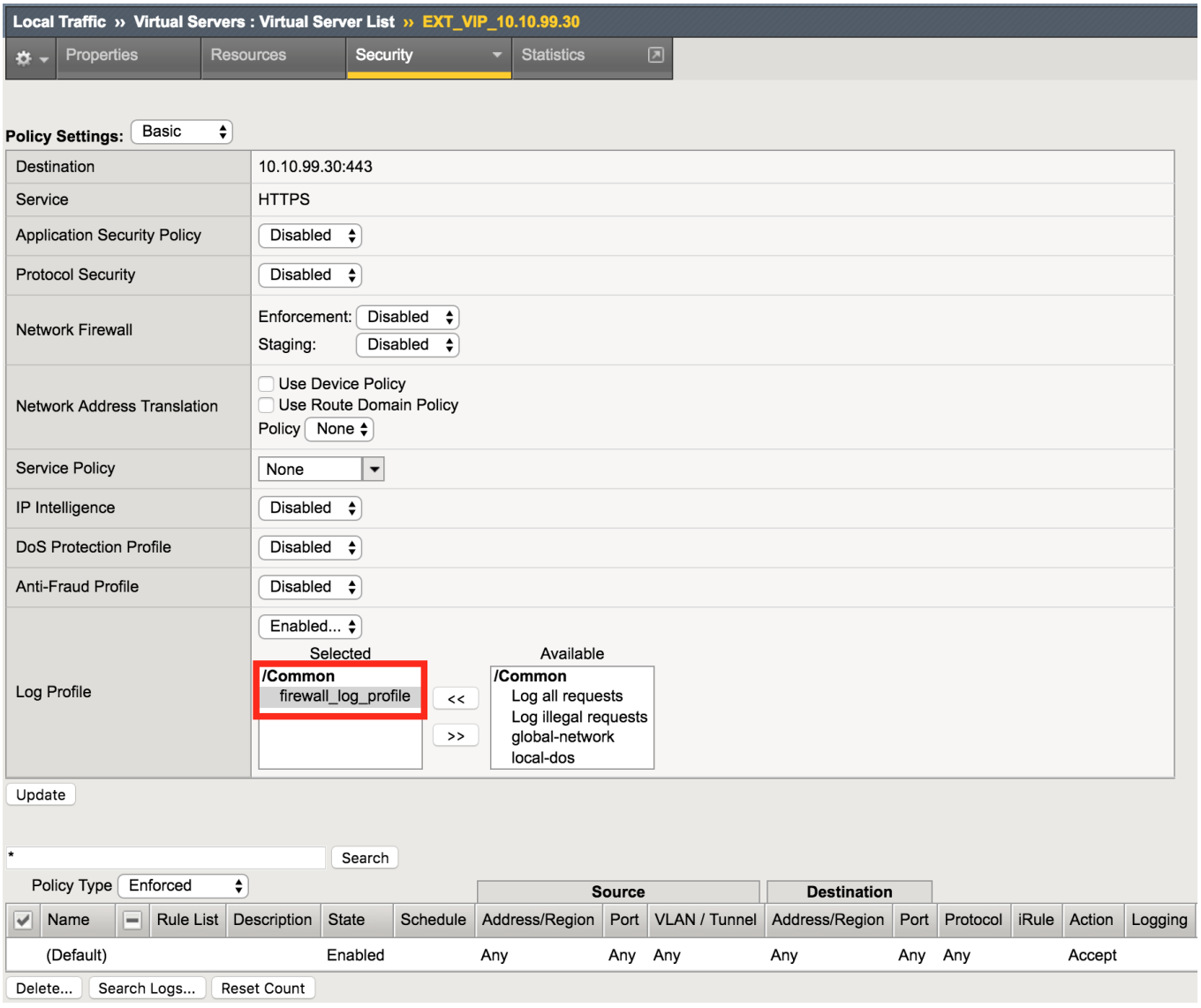
Apply the newly created log profile to the external virtual server created in the previous lab.

Navigation: Local Traffic > Virtual Servers > Virtual Server List

Navigation: Click on EXT_VIP_10.1.10.30

Navigation: Security tab > Policies

Log Profile	firewall_log_profile

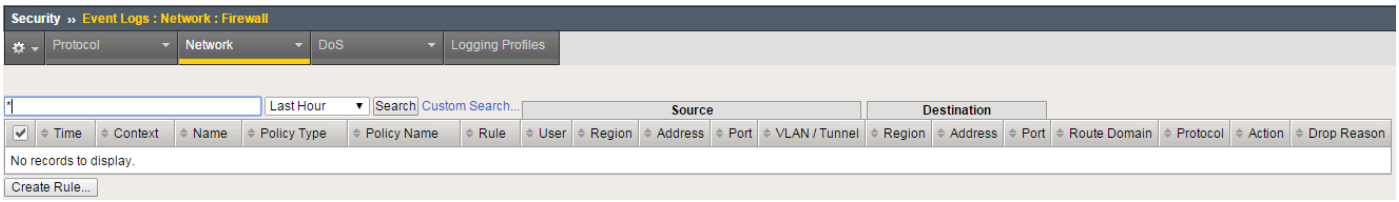


Note: Leave all other fields using the default values.

Navigation: Click Update

View network firewall logs.

Navigation: Security > Event Logs > Network > Firewall



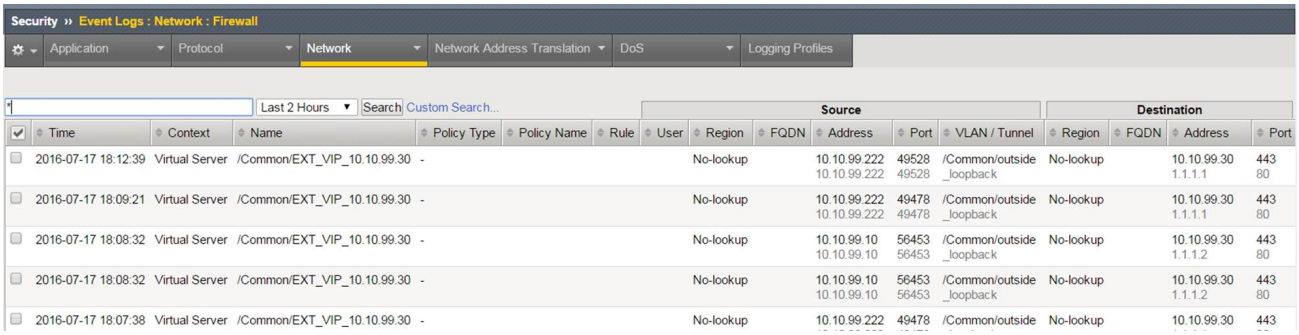
Validate Lab 3 Configuration

Open a new web browser tab and access the virtual server or repeat the curl statements from the previous sections.

URL: https://site1.com

Note: This test generates traffic that creates network firewall log entries.

Navigation: Security > Event Logs > Network > Firewall



Note: View new network firewall log entries. Examine the data collected there.

Note: This completes Module 1 - Lab 3

2.1.4 Lab 4: Configure A Firewall Policy and Firewall Rules For Each Application

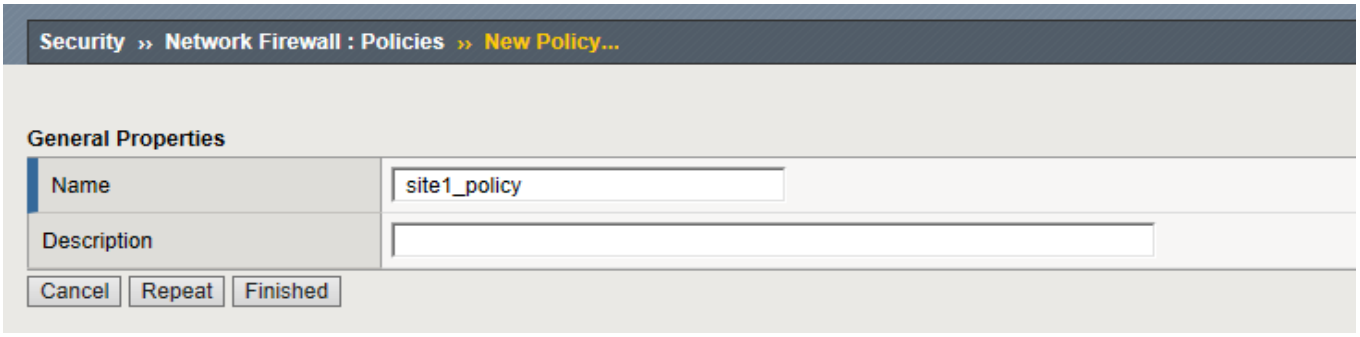
A network firewall policy is a collection of network firewall rules that can be applied to a virtual server. In our lab, we will create two policies, each of which includes two rules. This policy will then be applied to the appropriate virtual servers and tested.

Create The geo_restrict Firewall Rule List and Firewall Policy.

This example provides a firewall policy to the **www.site1.com** portion of the application. A real world example of this would be with companies hosting cryptographic software which is subject to export restrictions. In this case we will use the Geolocation feature to block access from a couple countries only and only on the site1.com application.

Navigation: Security > Network Firewall > Policies, then click Create

Name	site1_policy



Note: Leave all other fields using the default values.

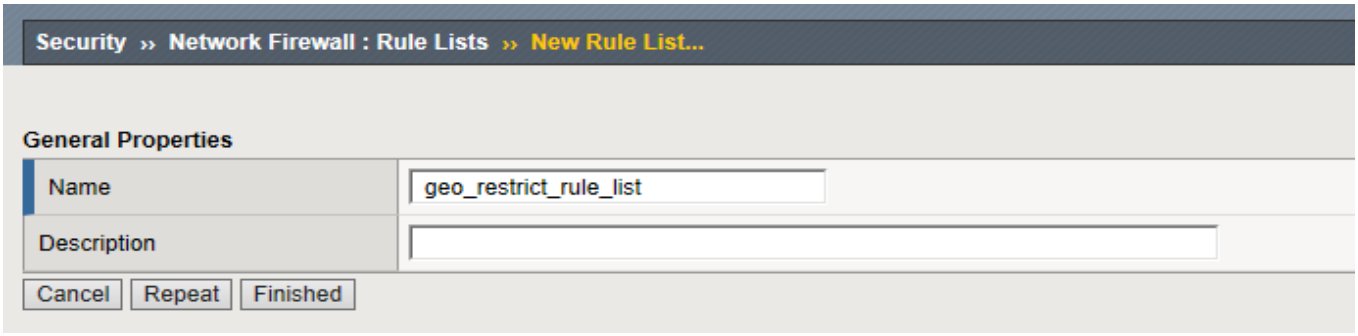
Navigation: Click Finished

Create an IP Drop Network Firewall Rule List

Note: we could have created a rule directly in the policy. Using Rule lists allows us to re-use this in multiple policies

Navigation: Security > Network Firewall > Rule Lists then click Create

Name	geo_restrict_rule_list



Navigation: Click Finished

Navigation: Click the geo_restrict_rule_list you just created

Navigation: Click Add

Name	block_AF_CN_CA
Order	First
Protocol	Any
Source	Country/Region: AF,CN,CA
Action	Drop
Logging	Enabled

Note: Leave all other fields using the default values.

Navigation: Click repeat

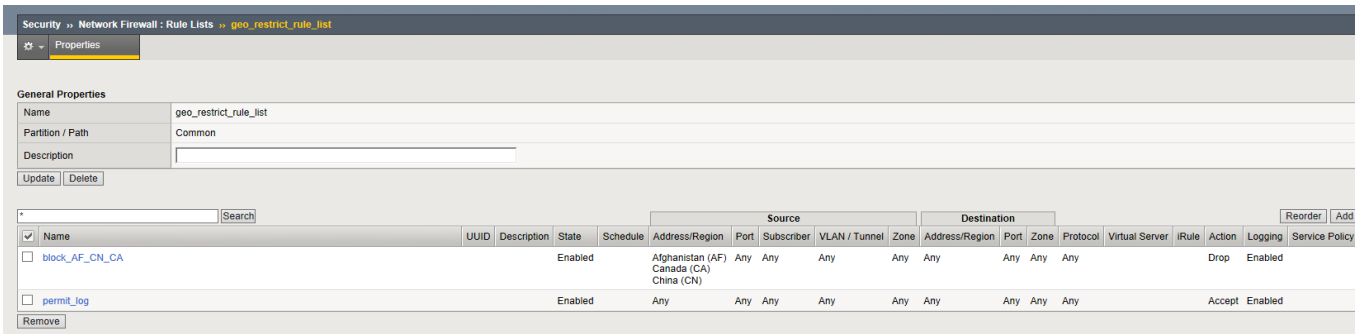
Navigation: Click Add

Name	permit_log
Order	Last
Action	Accept
Logging	Enabled

Create Permit Log Network Firewall Rule.

Note: Leave all other fields using the default values.

Navigation: Click Finished



Assign the geo_restrict_rule_list to the site1_policy

Navigation: Security > Network Firewall > Policies

Navigation: Click on **site1_policy** then click Add Rule List

In the name field start typing **geo** in the rule listfield. Select **geo_restrict_rule_list**

Navigation: Click Done Editing

Navigation: Click Commit Changes to System

Note: We want to validate the site is available before and after applying the Network Firewall Policy

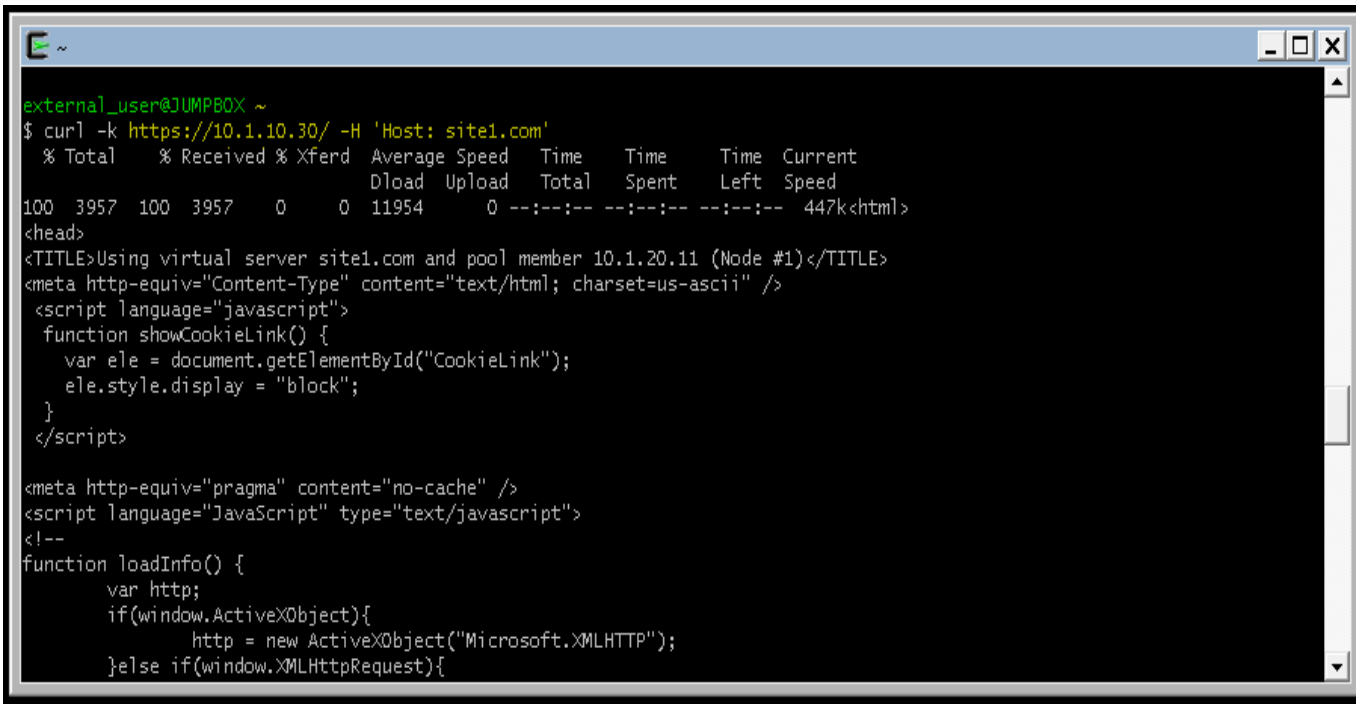
---

From client machine try to connect again to the application site.

URL: https://site1.com

We will use Cywin Terminal for more controlled testing in

```
curl -k https://10.1.10.30/ -H 'Host: site1.com'
```



Note: We want to validate the site is available before and after applying the Network Firewall Policy

Assign The Policy To The Virtual Server

A unique feature of the BIG-IP Firewall Module allows L3-4 security policies to be assigned specifically to an application i.e. Virtual Server. So each application can have its own firewall policy separate from other application virtual servers.

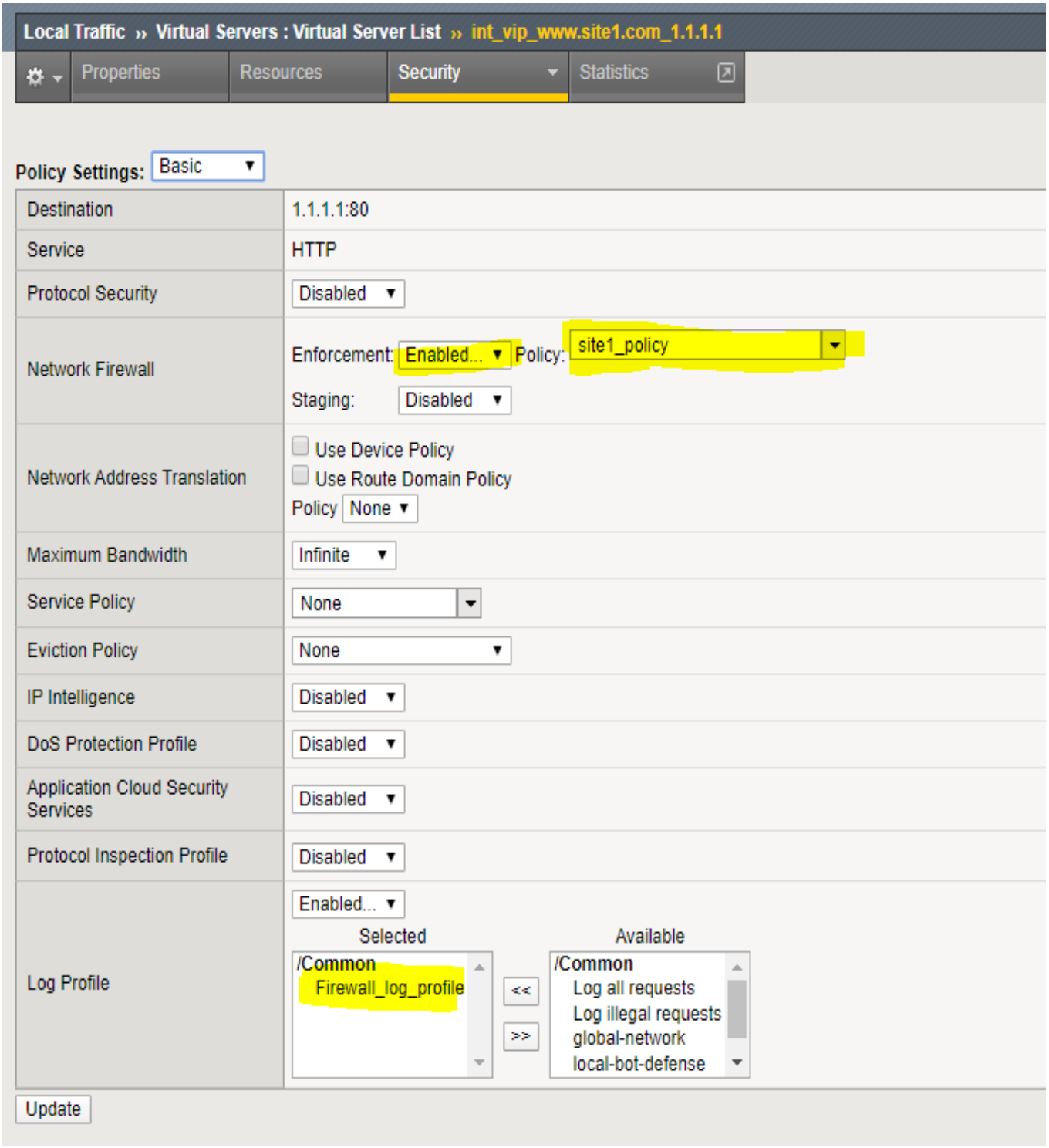
Apply the Network Firewall Policy to Virtual Server

Navigation: Local Traffic > Virtual Servers then click int_vip_www.site1.com_1.1.1.1

Navigation: Click on the Security Tab and select Policies

Edit the Network Firewall section of the screen

Virtual Server	int_vip_www.site1.com_1.1.1.1
Enforcement	Enabled
Policy	site1_policy
Log Profile	enabled
Log Profile	firewall_log_profile



Note: Leave all other fields using the default values.

Navigation: Click Update

From client machine validate the behavior of the Policy and the associated Rule List

Many enterprise sites have some or all of their content served up by Content Delivery Networks (CDN). This common use case leverages proxies to provide static content closer to the end client machines for performance. Because of this there may only be one or two IP addresses connecting to the origin website. The original IP address of the client in this case is often mapped to a common HTTP header X-Forwarded-For or some variation. In this deployment, the BIG-IP can translate the original source of the request in the XFF to the source IP address.

Use Cywin Terminal to allow us to specify the X-Forwarded-For header. . There is an iRule applied to EXT_VIP_10_1_10_30 which SNAT's the source IP to match the X-Forwarded-For header

XFF-SNAT iRule

```
when HTTP_REQUEST {
  if {[HTTP::header exists "X-Forwarded-For"]}  {
  snat [HTTP::header X-Forwarded-For]
  log local0. '[HTTP::header X-Forwarded-For]'
  }
}
```

```
curl -k https://10.1.10.30/ -H 'Host: site1.com'
```

Note: Since we did not define the header, the firewall will see the RFC-1918 address of the jimp host (10.1.10.199)

URL: https://site1.com

Use the -H option in curl to define the X-Forwarded-For Header. This will trigger the iRule addigned to the External VIP to simulate specific IP addresses in the header

```
curl -k https://10.1.10.30/ -H 'Host:site1.com' -H 'X-Forwarded-For: 172.16.99.5'
```

Review the logs. each connection will log events from the external and internal virtual server

Navigation: Security > Event Logs > Network > Firewall

Next we will simulate a connection an IP address in Bejing, China

The BIG-IP Geolocation database is supplied by Digital Element

URL: http://www.digitalelement.com/

URL: https://whatismyipaddress.com/ip/1.202.2.1 shows that this address is in Beijing , China

Note: You can check the geo classification of an address from the BIG-IP CLI using the command geoip_lookup 1.202.2.1

```
curl -k https://10.1.10.30/ -H 'Host: site1.com' -H 'X-Forwarded-For: 1.202.2.1'
```

This connection attempt will fail. Return to the BIG-IP GUI and refresh the firewall event log.

Note: you may need to zoom the browser to see the "Action" collumn at the right sie of the screen

Create A Separate Policy For The site2 Virtual Server

Now we want to create a second policy for access to site2

Create Network Firewall Policy

Navigation: Security > Network Firewall > Policies, then click Create

Note: Leave all other fields using the default values.

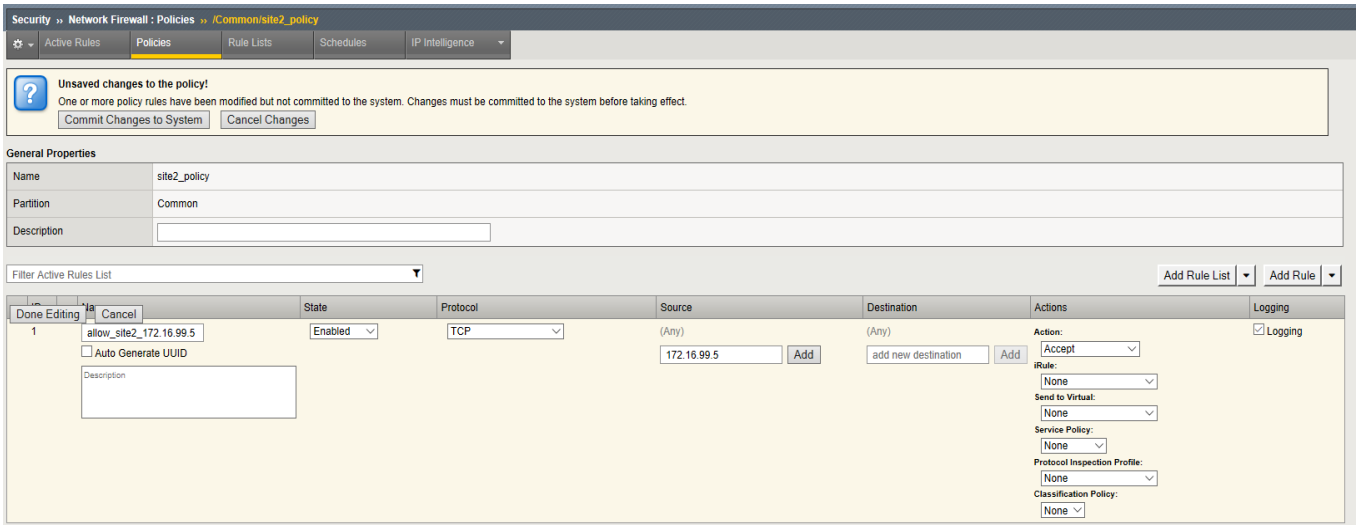
Navigation: Click Finished

Modify the policy with rules to Allow TCP Port 80 From Host 172.16.99.5 Network Firewall Rule and deny all other adresses . This time we will build the rules directly into the policy instead of using a Rule List

Navigation: Click on the site2_policy you just created

Navigation: Click Add Rule pull down on the upper right - Add rule at beginning

Name	allow_site_172.16.99.5
Protocol	TCP (6)
Source	Address: 172.16.99.5
Action	Accept
Logging	Enabled



Note: Leave all other fields using the default values.

Navigation: Click Done Editing

Create Deny Log Network Firewall Rule

Navigation: Click Add Rule pull down on the upper right - Add rule at end

Note: As we are deployed in “ADC Mode” where the default action on a virtual server is ‘Accept’, we must also create a default deny rule.

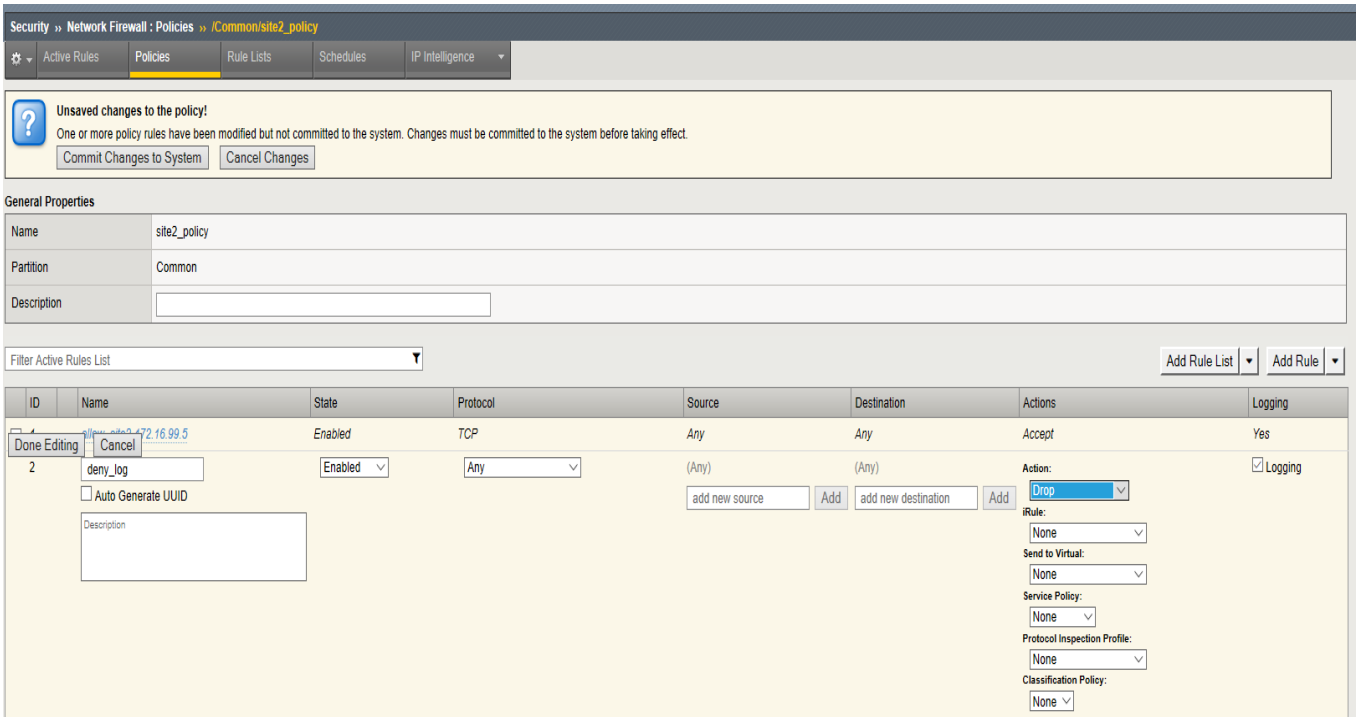
For further discussion of Firewall vs ADC modes, please consult the F5 BIG-IP documentation.

URL: https://support.f5.com/kb/en-us/products/big-ip-afm/manuals/product/network-firewall-policies-implementations-13-0-0/8.html

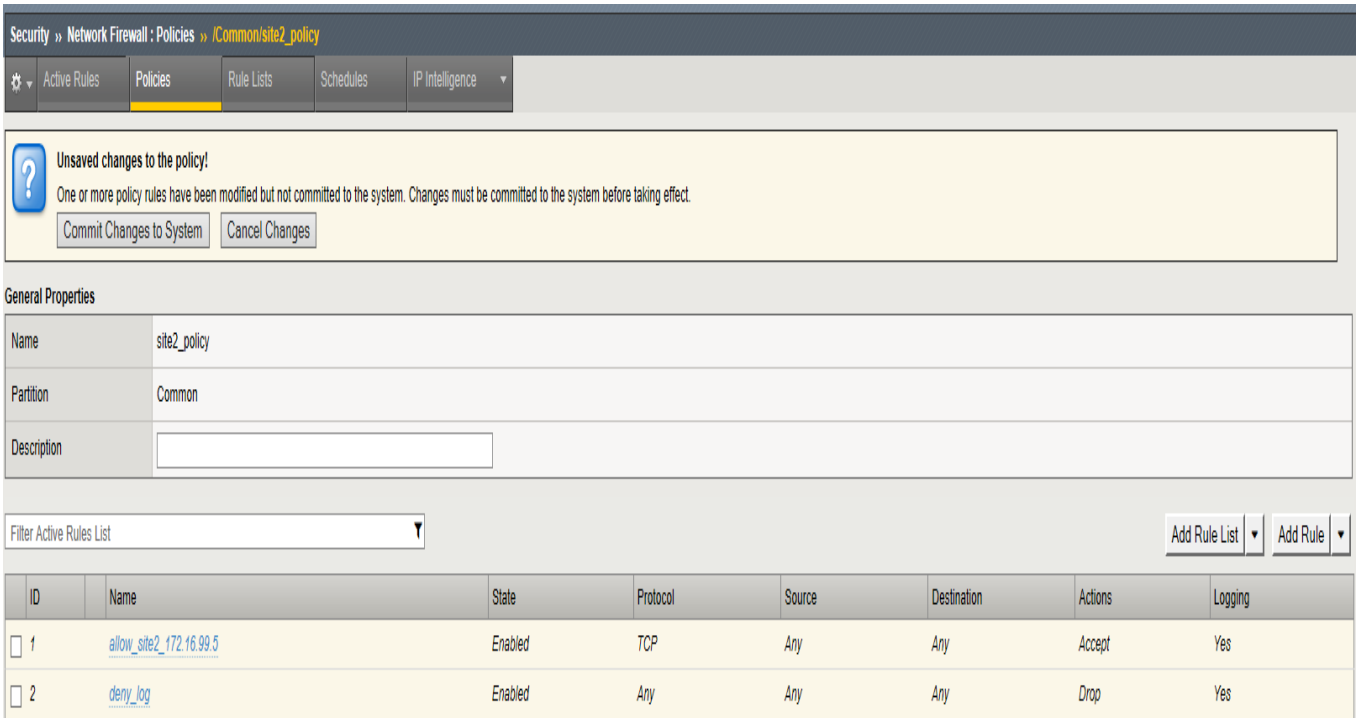
Name	deny_log
Action	Drop
Logging	Enabled

Note: Leave all other fields using the default values.

Navigation: Click Done Editing



Navigation Click Commit Changes To System



Navigation: Click Finished

Apply the Network Firewall Policy to Virtual Server

Navigation: Local Traffic > Virtual Servers

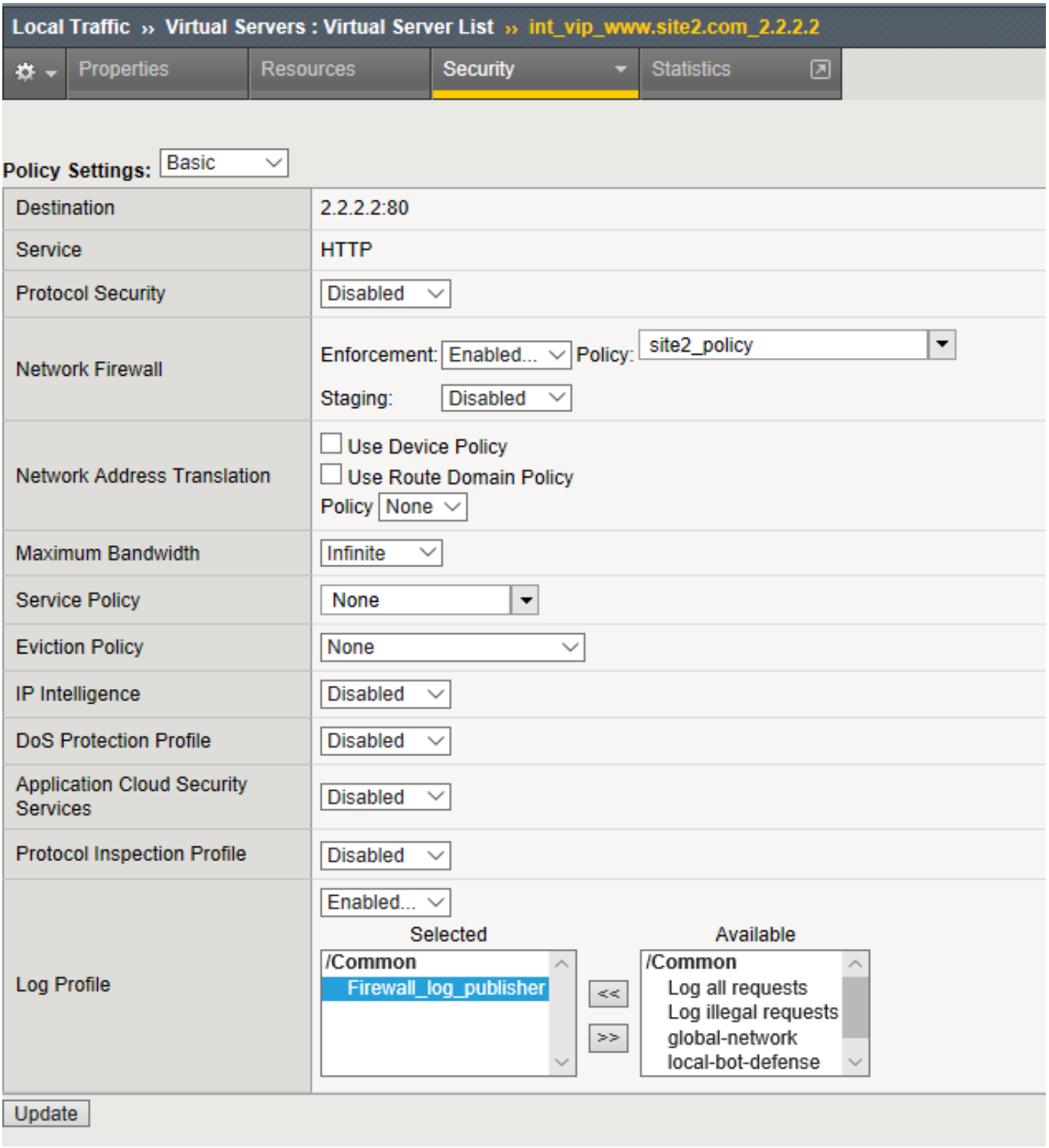
Navigation: Click on int_vip_www.site2.com_2.2.2.2

Navigation: Select the Security Tab and select Policies

Virtual Server	int_vip_www.site2.com_2.2.2.2
Network Firewall	Enabled
Policy	site2_policy
Log Profile	enabled
Log Profile	firewall_log_profile

Note: Leave all other fields using the default values.

Navigation: Click Update



Note: Leave all other fields using the default values.

Navigation: Click Update

From client machine

From client machine validate the behavior of the Policy and the associated Rule List

We will use Cywin Terminal to allow us to specify the source IP address. This is done by leveraging an iRule which SNAT's the source IP to match the X-Forwarded-For header. This iRule is applied to EXT_VIP_10_1_10_30

```
curl -k https://10.1.10.30/ -H 'Host:site2.com' -H 'X-Forwarded-For: 172.16.99.5'
```

```
curl -k https://10.1.10.30/ -H 'Host:site2.com' -H 'X-Forwarded-For: 172.16.99.7'
```

Note: This is expected to fail

Note: This concludes Module 1 - Lab 4

2.1.5 Lab 5: Provide Firewall Security Policies For CDN Enabled Applications

Many enterprise sites have some or all of their content served up by Content Delivery Networks (CDN). This common use case leverages proxies to provide static content closer to the end client machines for performance. Because of this there may only be one or two IP addresses connecting to the origin website. The original IP address of the client in this case is often mapped to a common HTTP header X-Forwarded-For or some variation. In this deployment, the BIG-IP can translate the original source of the request in the XFF to the source IP address.

In this case we are going to leverage iRules to modify the traffic coming from the CDN networks so we can apply a firewall policy to it. The iRule to accomplish this is already installed on your BIG-IP and applied to EXT_VIP_10_1_10_30. We have been leveraging it to run the tests in previous exercises

```
when HTTP_REQUEST {
   if { [HTTP::header exists "X-Forwarded-For"] } {
      snat [HTTP::header X-Forwarded-For]
      log local0. [HTTP::header X-Forwarded-For]
   }
}
```

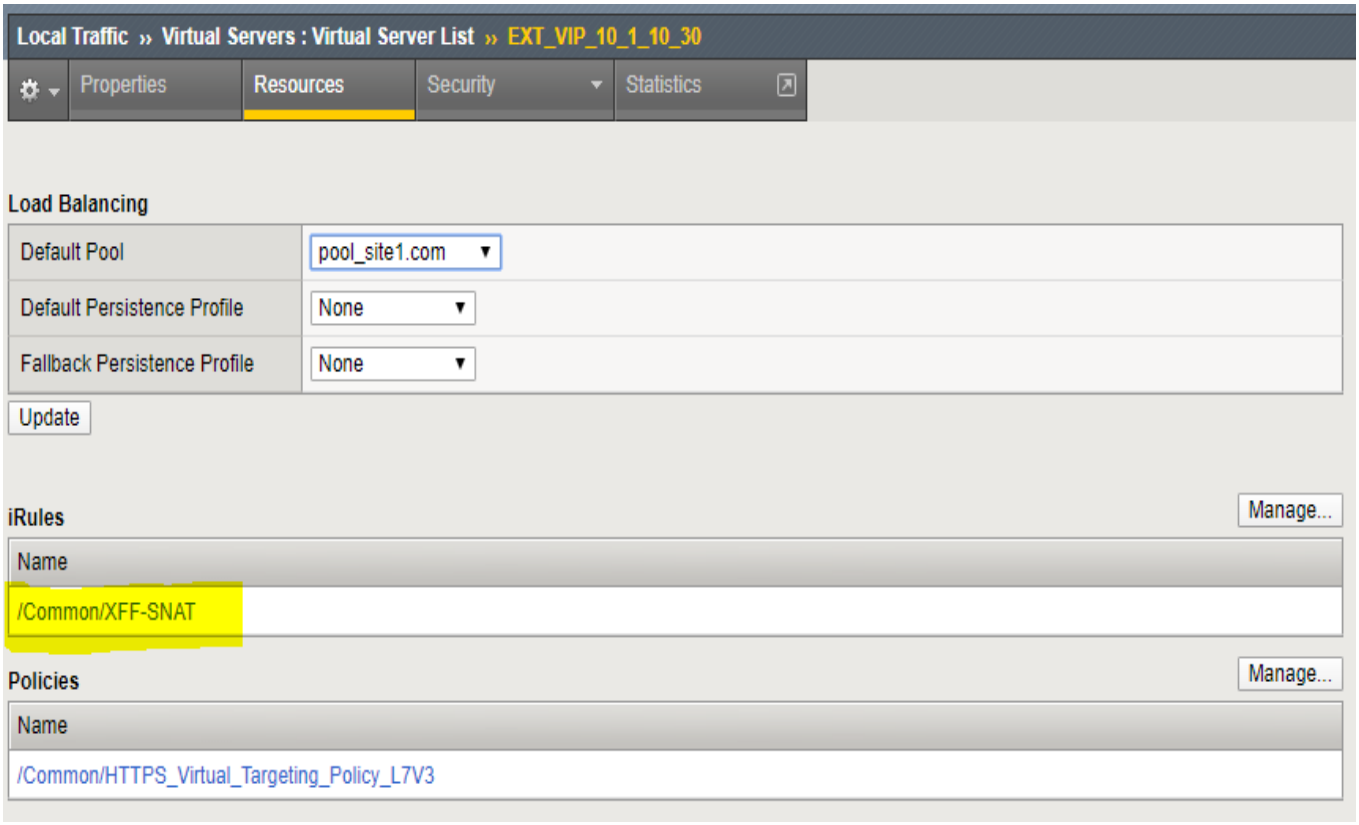
Examminig the iRule we find that it is called when an HTTP request happens. It then checks to see if the `X-Forwarded-For` header exists (We wouldn't want to SNAT to a non-existent IP address) and if it does it modifies the source IP address of the request to the IP address provided in the header.

Verify that the iRule is assigned to the Virtual Server

Navigation: Local Traffic > Virtual Servers

Navigation: Click on the `EXT_VIP_10.1.10.30` virtual server

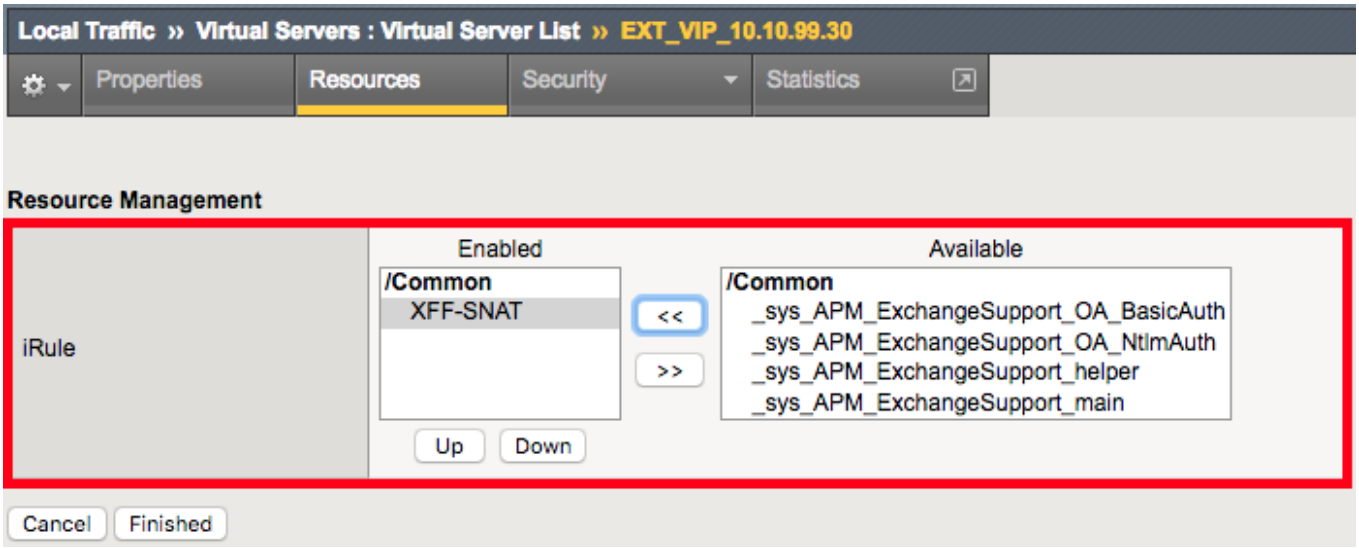
Navigation: Click on the Resources tab



Navigation: Click on the Manage button

This is where you assign iRules

Navigation: Click on the Cancel button since the iRule is already assigned



Validate SNAT Function

We tested functionality in prior exercises with the commands below. Leverage curl from the Cygwin Terminal to insert the X-Forwarded-For header in to the request.

```
curl -k https://10.1.10.30 -H 'Host: site1.com' -H 'X-Forwarded-For: 1.202.2.1'
```

Expected Result: The site should be blocked by the geo_restrict_rule_list and generate a 403 Forbidden response

---

Note: Optionally you can log into the CLI on the BIG-IP. Putty BIGIP_A –Uersname: root Password: f5DEMOs4u Then tail -f /var/log/ltm. The iRule logs the SIP

Validate that requests sourced from the X-Forwarded-For IP address of 172.16.99.5 allowed.

```
curl -k https://10.1.10.30 -H 'Host:site1.com' -H 'X-Forwarded-For: 172.16.99.5'
```

Expected Result: Page will work

```
{
  "web-app": {
    "servlet": [
    {
    "servlet-name": "cofaxCDS",
    "servlet-class": "org.cofax.cds.CDSServlet",
```

Solve For TCP Issues With CDN Networks

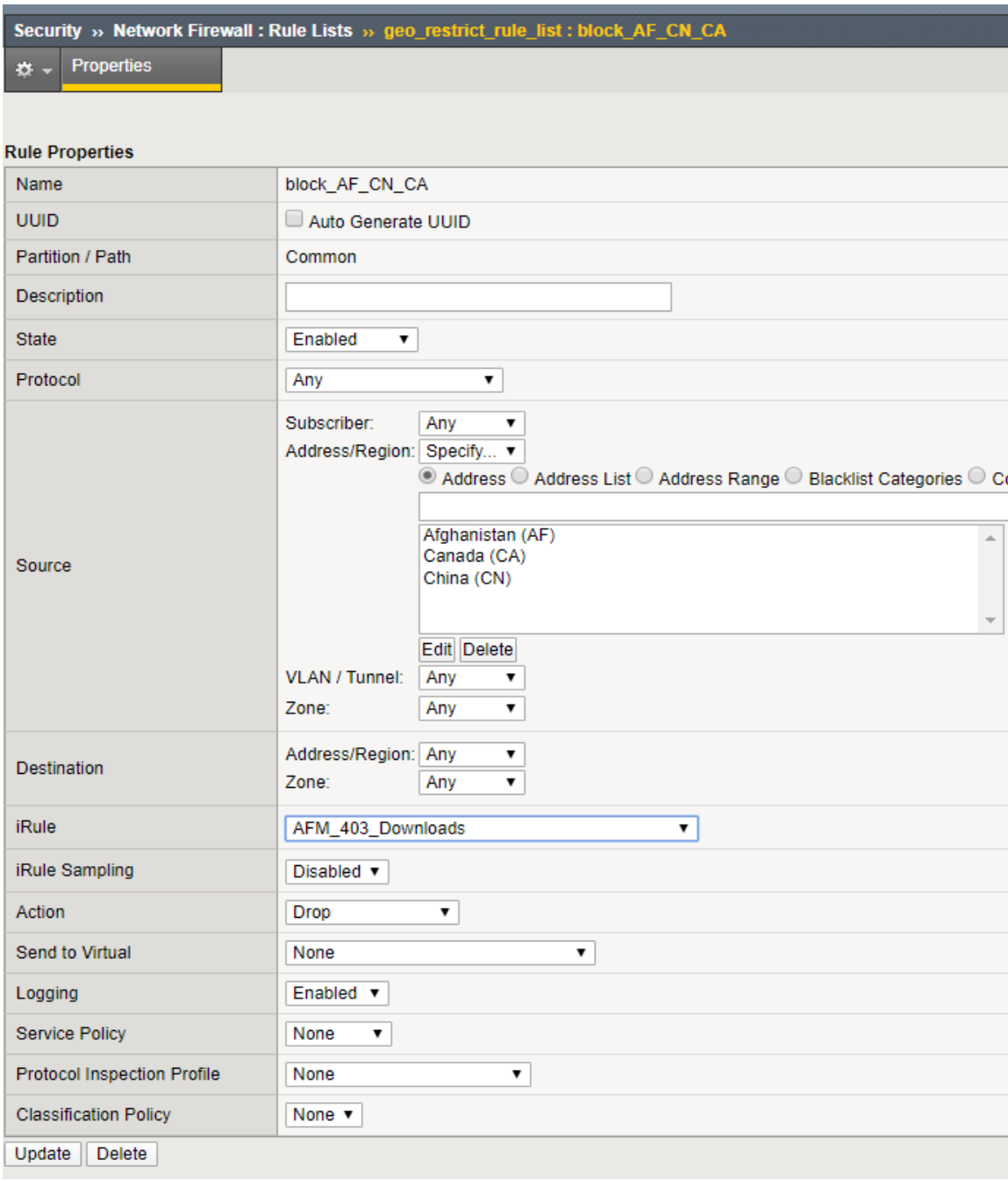
The next step is to solve for the TCP connection issue with CDN providers. While we are provided the originating client IP address, dropping or reseting the connection can be problematic for other users of the application. This solution is accomplished via AFM iRules. The iRule is already provided for you. We need to apply it to the Network Firewall downloads_policy Policy. It still is logged as a drop or reset in the firewall logs. We allow it to be processed slightly further so that a Layer 7 response can be provided.

Navigation: Security > Network Firewall > Rule Lists

Navigation: Select **geo_restrict_rule_list**

Navigation: Select **block_AF_CN_CA**

Navigation: Add the AFM_403_Downloads iRule to the rule list



Navigation Click Update

Validate that denied requests are now responded with a Layer 7 **403 Error** Page.

```
curl -k https://10.1.10.30/ -H 'Host:site1.com' -H 'X-Forwarded-For: 1.202.2.1'
```

Expected Result: Instead of the traffic getting dropped, a 403 error should be returned.

```
<html>
  <head>
    <title>403 Forbidden</title>
  </head>
  <body>
```

(continues on next page)

(continued from previous page)

```
    403 Forbidden Download of Cryptographic Software Is Restricted
  </body>
</html>
```

Attention: Since a TCP solution could cause users to be blocked without explanation , the HTML error response will traverse the CDN network back only to the originating client. Using a unique error code such as 418 (I Am A Teapot) would allow you to determine that the webserver is likely not the source of the response. It would also allow the CDN network providers to track these error codes. Try to find one that has a sense of humor.

Note: This concludes Module 1 - Lab 5

2.1.6 Lab 6: Configure HTTP security

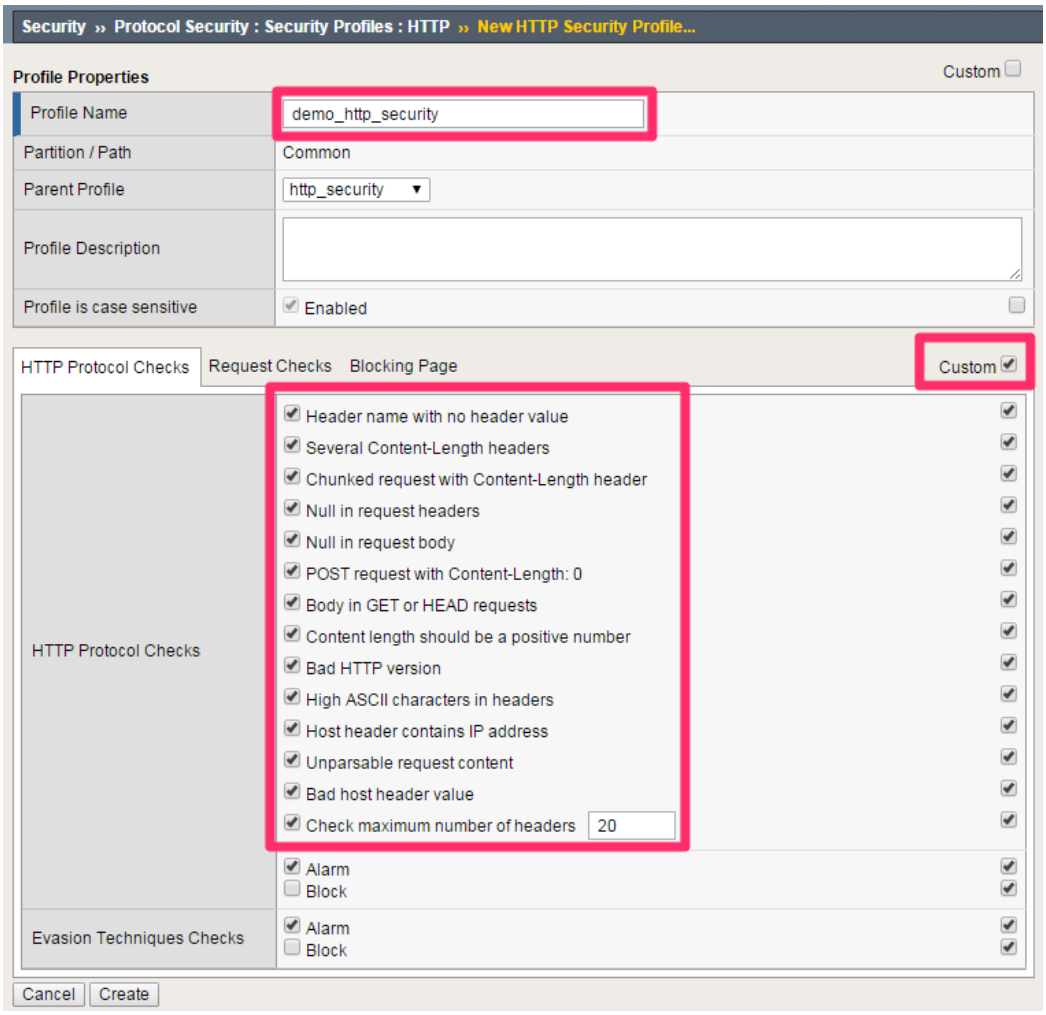
HTTP security profiles are used to apply basic HTTP security to a virtual server. Significantly more advanced HTTP security is available by adding ASM (Application Security Manager).

Configure An HTTP Security Profile And Apply It To The External Virtual Server.

On the BIG-IP:

Navigation: Security > Protocol Security > Security Profiles > HTTP, then click Create.

Profile Name	demo_http_security
Custom	Checked
Profile is case sensitive	Checked
HTTP Protocol Checks	Check All

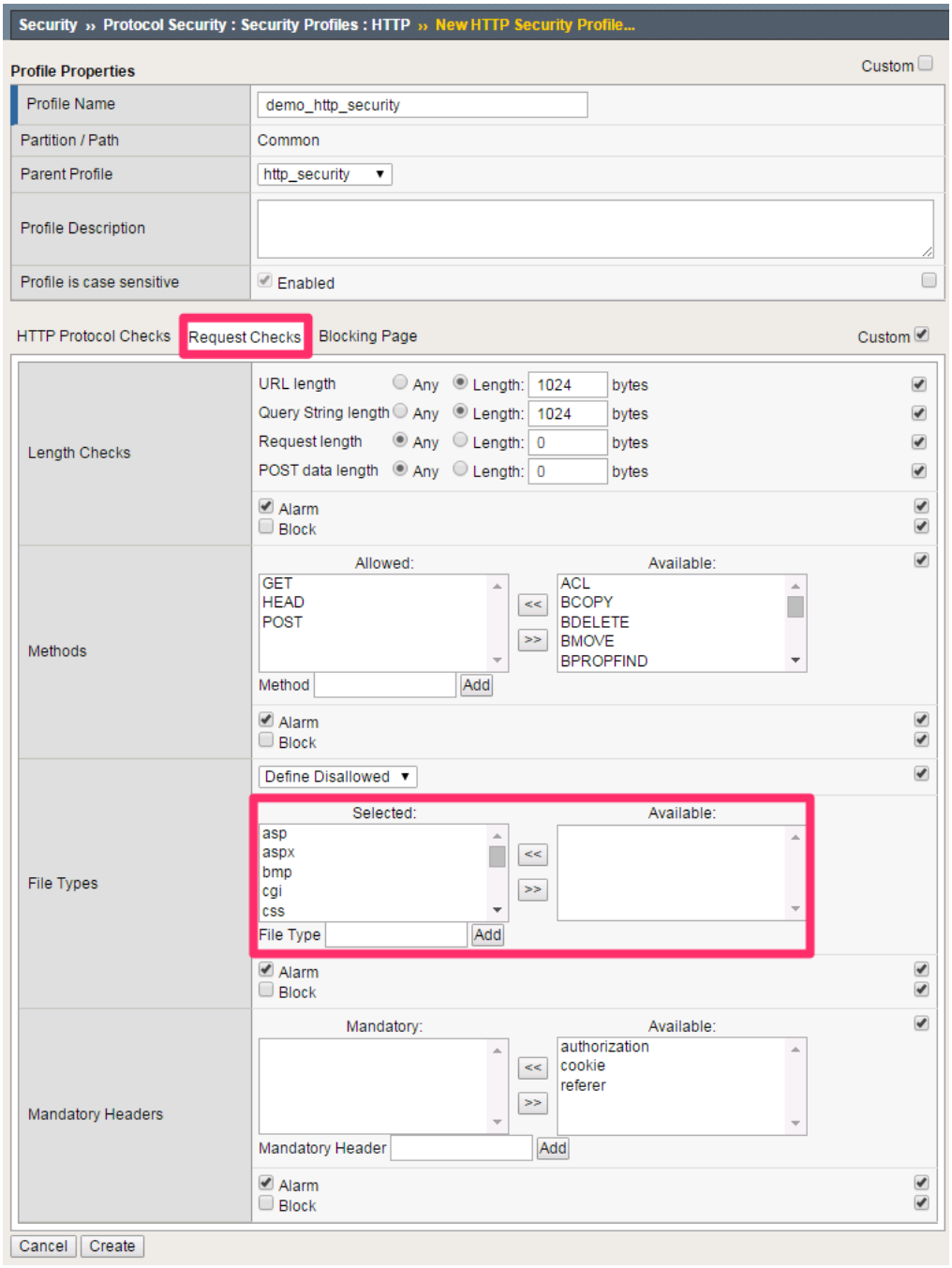


Note: Leave all other fields using the default values.

Navigation: Click Request Checks Tab.

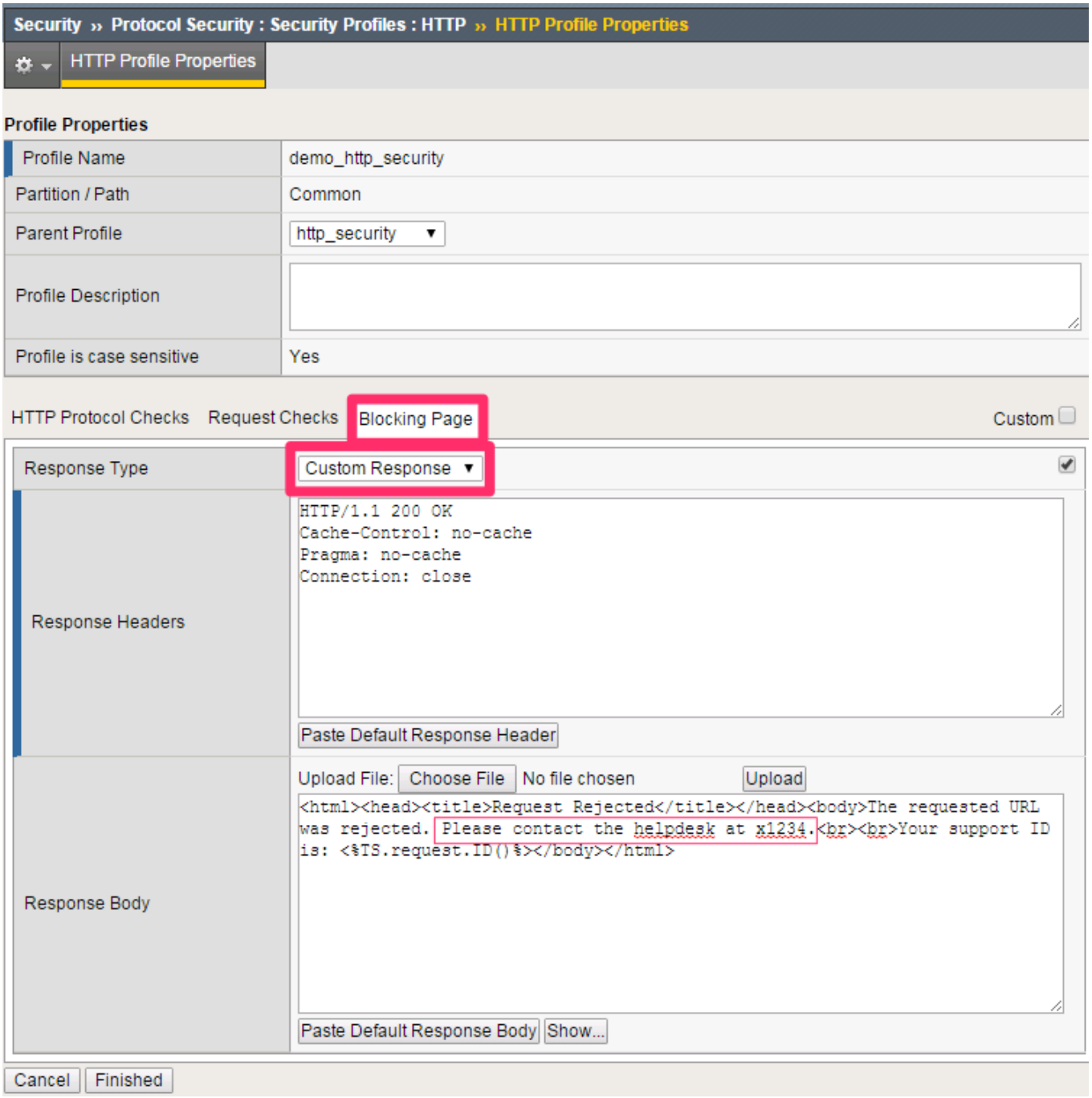
Note: Leave the defaut Methods. Changing Methods is a powerful way to protect your web sites

File Types	Select All



Navigation: Click Blocking Page Tab.

Response Type	Custom Response
Response Body	Insert “Please contact the helpdesk at x1234” as noted below



Note: Leave all other fields using the default values.

Navigation: Click Create

Note: We did not put the policy in Blocking mode. We will do that after we verify functionality

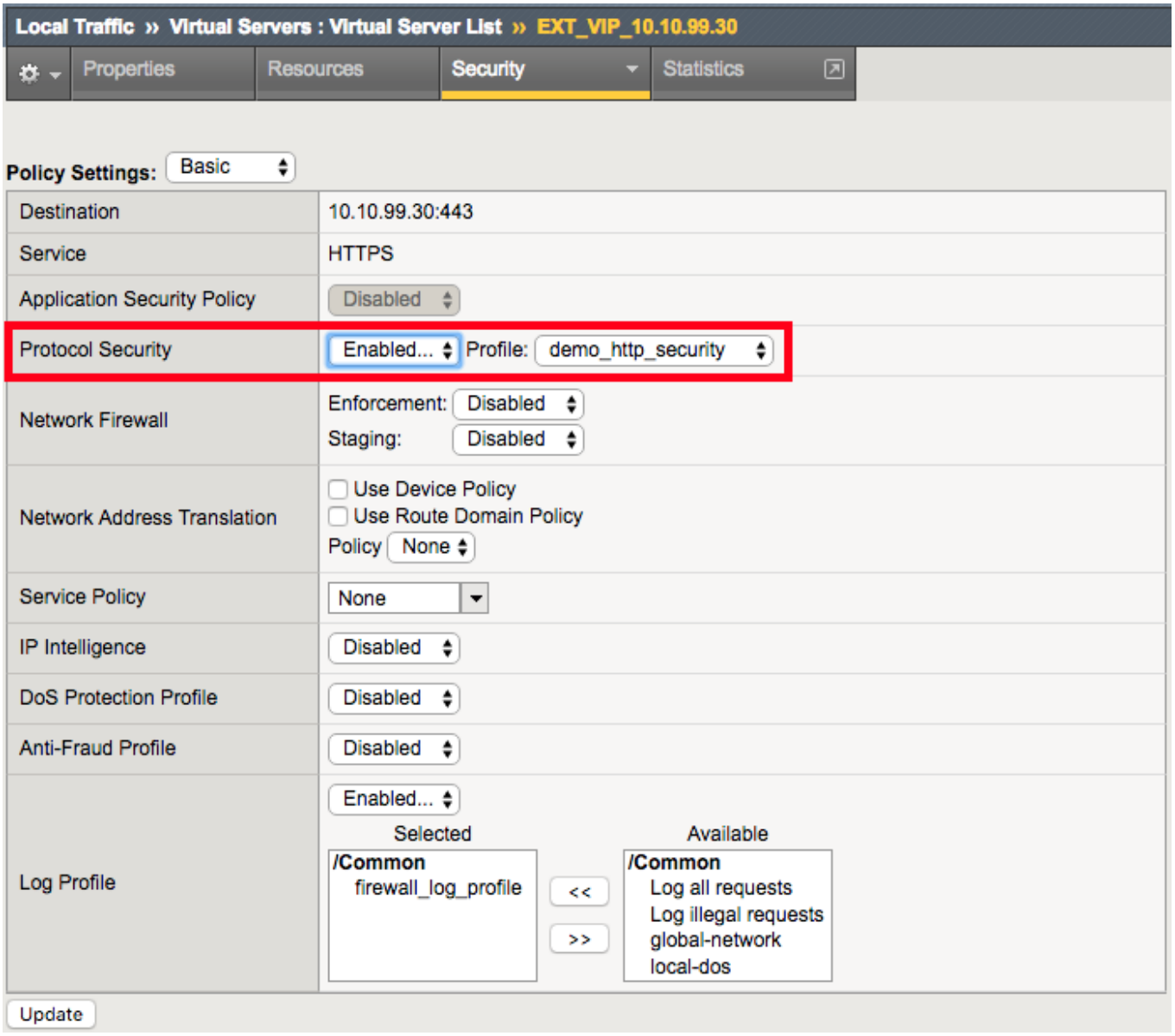
Apply the HTTP security profile to the external virtual server.

Navigation: Local Traffic > Virtual Servers > Virtual Server List >

Navigation: Select EXT_VIP_10.1.10.30

Navigation: Select the Security tab

Protocol Security	Enabled	demo_http_security



Note: Leave all other fields using the default values.

Navigation: Click Update.

Open a new web browser tab, access the virtual server and log into the application.

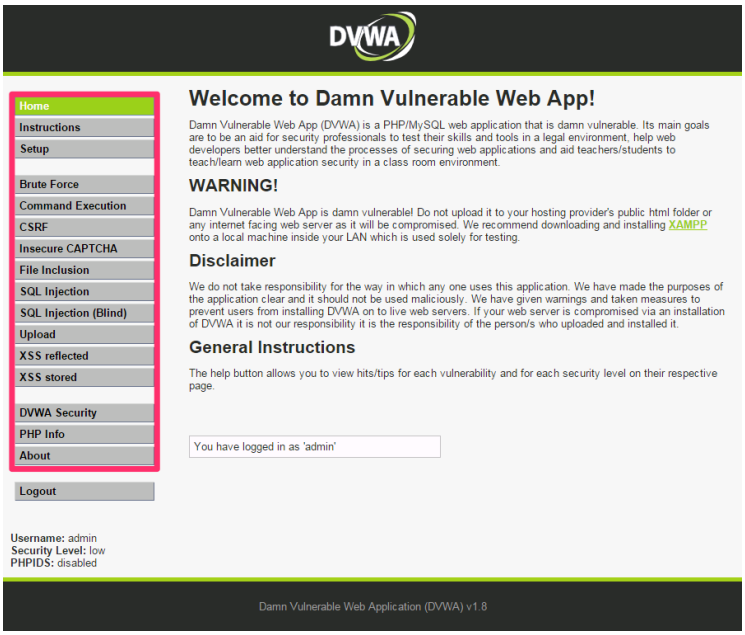
URL: https://dvwa.com

Credentials: admin/password

Note: This application is accessible, even though there are policy violations, because the “Block” option in the HTTP security policy is not selected.

Browse the application.

Navigation: Click on various links on the sidebar.

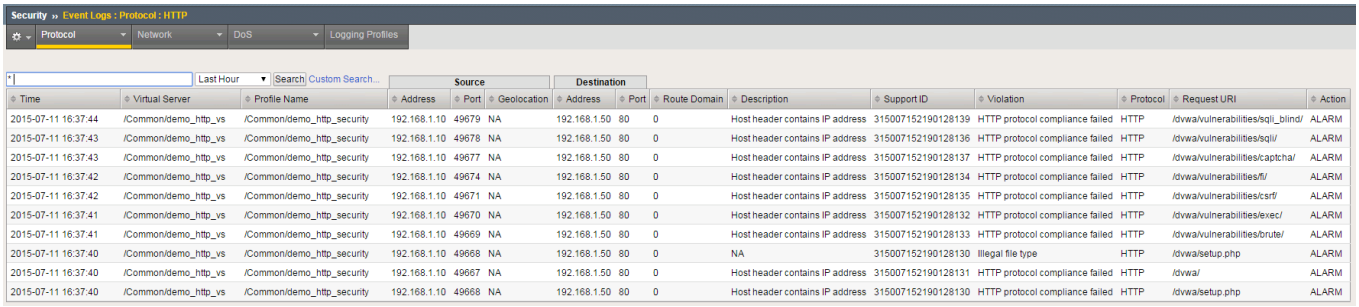


Note: This traffic will generate network firewall log entries because the Alarm option in the HTTP security policy is selected.

On BIG-IP

Review the log entries created in the previous step.

Navigation: Security > Event Logs > Protocol > HTTP



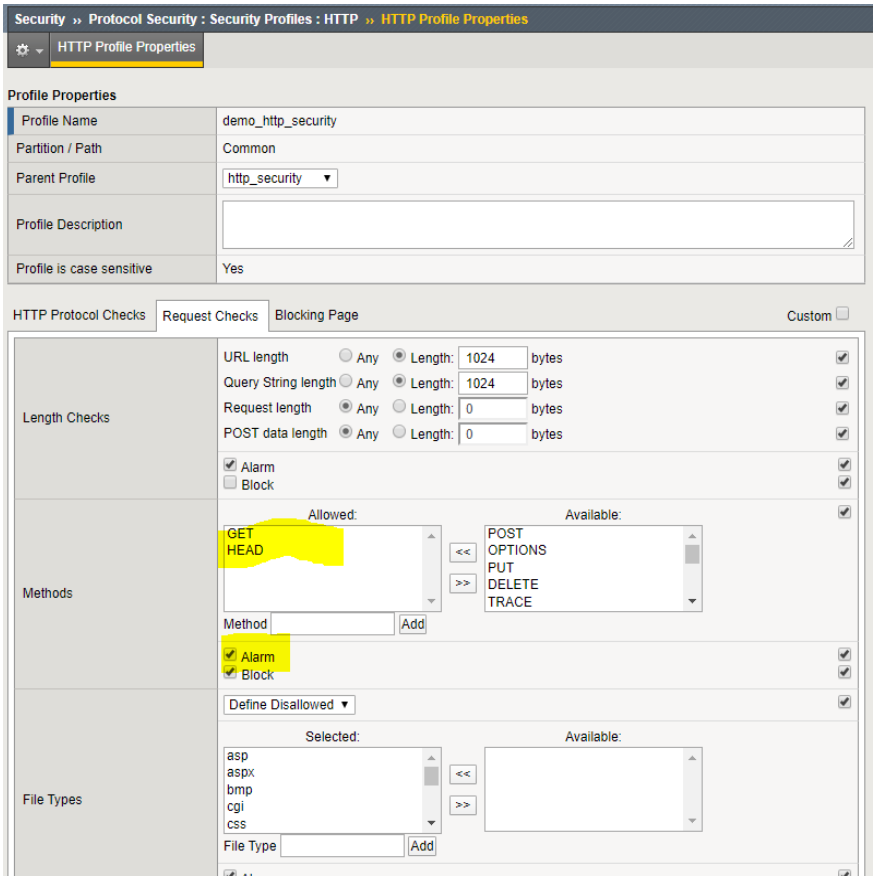
Note: Your log entries may be different than the example shown above but the concept should be the same.

Edit the demo_http_security HTTP security profile.

Navigation: Security > Protocol Security > Security Profiles > HTTP

Navigation: Select the **demo_http_security** profile

Navigation: Select the Request Checks Tab



Note: Leave all other fields using the default values.

Navigation: Click Finished.

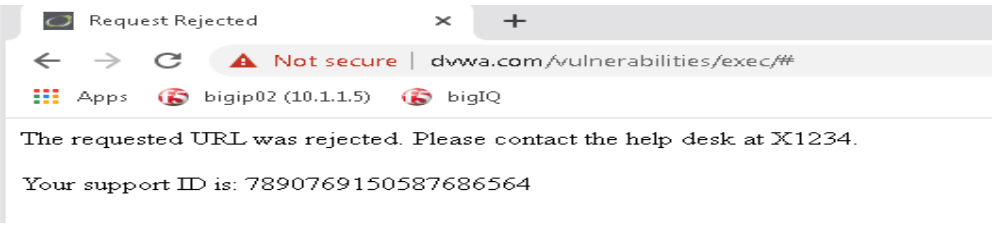
On Windows jumpbox

Close the Browser window to dvwa.com

Open a new web browser tab and access the virtual server.

URL: https://dvwa.com

Credentials: admin/password



Attention: This action requires a "POST" action and will be blocked because this is not allowed.

Note: This is the end of Module 1 - Lab 6

2.1.7 Lab 7: Configure A Clone Pool For SSL Visibility To IDS Sensors Or Other Security Tools

SSL encrypted traffic poses a problem for most security devices. The performance of those devices is significantly impacted when trying to decrypt SSL traffic. Since the BIG-IP is designed to handle SSL traffic with specialized hardware and optimized software libraries, it is in the unique position to 'hand-off' a copy of the decrypted traffic to other devices.

In this solution, since the BIG-IP is terminating SSL on the external virtual server, when we forward the traffic to the secondary virtual server in clear-text we have an opportunity to make an unencrypted copy of the application traffic and send it to an external sensor such as an IDS for further security assessment.

On BIG-IP

Inspect the preconfigured IDS_Pool.

Navigation: Local Traffic > Pools > Pool List >

Navigation: Click on the **Members** Tab

Note: Unencrypted traffic will be forwarded to this IP address

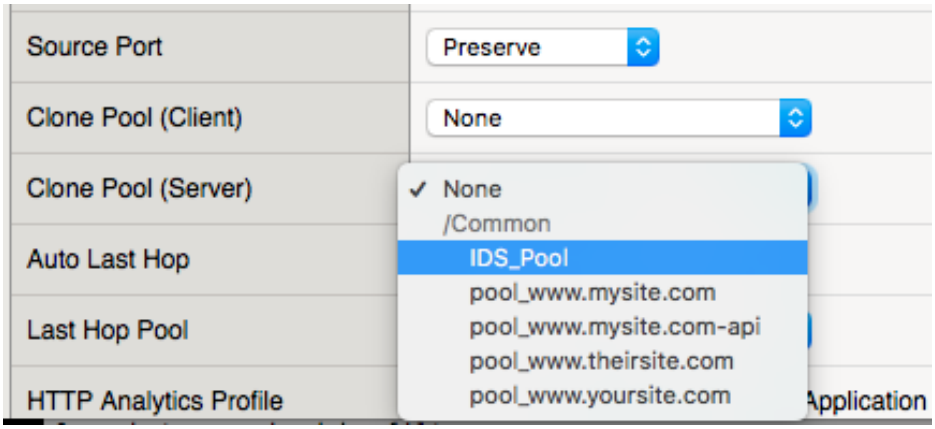
Attach the *IDS_Pool* as a clone pool to the server side of the external virtual server

Navigation: Local Traffic > Virtual Servers > Virtual Server List > EXT_VIP_10_1_10_30.

Navigation: Select **Advanced** from the pulldown at the top of the Configuration section

Navigation: Scroll to the configuration for Clone Pool (Client) and select None

Navigation: Scroll to the configuration for Clone Pool (Server) and select IDS_pool



Navigation: Click on update at the bottom of the page.

Note: Leave all other fields using the default values.

Select the Putty application from the desktop on the jump host

Load **Lamp Server** from the sessions list

Open Lamp Server

Accept the certificate warning

login as **f5**

> **Attention:** It will take about 30 seconds for the certificate login process- No password required

Input the TCPDUMP command to start capturing traffic

```
sudo tcpdump -i eth1 -c 200 port 8081
```

Initiate another attempt to connect to the website via curl using the Cygwin application on the desktop. Position windows on the desktop so that you can see both the Putty session and the Cygwin session

```
curl -k https://10.1.10.30:8081 -H 'Host:site1.com' -H 'X-Forwarded-For: 172.16.99.5'

curl -k https://10.1.10.30:8081 -H 'Host:site3.com' -H 'X-Forwarded-For: 172.16.99.5'
```

Initiate another attempt to connect to the websites using the browser

```
https://site2.com:8081

https://site4.com:8081
```

View the tcpdump output on the syslog-webserver.

> **Attention:** It will take about 20 seconds after the transaction to appear in the tcpdump session. This is a performance problem on the lamp server

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth2, link-type EN10MB (Ethernet), capture size 262144 bytes
17:25:42.585675 IP 10.10.99.222.50924 > 1.1.1.1.http: Flags [S], seq 912073522, win␣
↪4380, options [mss 1460,sackOK,eol], length 0
17:25:42.585905 IP 1.1.1.1.http > 10.10.99.222.50924: Flags [S.], seq 1263282834, ack␣
↪912073523, win 4380, options [mss 1460,sackOK,eol], length 0
17:25:42.585918 IP 10.10.99.222.50924 > 1.1.1.1.http: Flags [.], ack 1, win 4380,␣
↪length 0
17:25:42.585926 IP 10.10.99.222.50924 > 1.1.1.1.http: Flags [P.], seq 1:79, ack 1,␣
↪win 4380, length 78
17:25:42.586750 IP 1.1.1.1.http > 10.10.99.222.50924: Flags [.], ack 79, win 4458,␣
↪length 0
17:25:42.673178 IP 1.1.1.1.http > 10.10.99.222.50924: Flags [P.], seq 1:252, ack 79,␣
↪win 4458, length 251
17:25:42.673231 IP 10.10.99.222.50924 > 1.1.1.1.http: Flags [.], ack 252, win 4631,␣
↪length 0
17:25:42.676360 IP 10.10.99.222.50924 > 1.1.1.1.http: Flags [F.], seq 79, ack 252,␣
↪win 4631, length 0
17:25:42.676972 IP 1.1.1.1.http > 10.10.99.222.50924: Flags [.], ack 80, win 4458,␣
↪length 0
17:25:42.688028 IP 1.1.1.1.http > 10.10.99.222.50924: Flags [F.], seq 252, ack 80,␣
↪win 4458, length 0
17:25:42.688057 IP 10.10.99.222.50924 > 1.1.1.1.http: Flags [.], ack 253, win 4631,␣
↪length 0
```

Note: Inspect the source and destination addresses. This traffic is cloned from the EXT_VIP

Note: This is the end of Module 1 - Lab 7.

2.2 Module 2: F5 Dynamic Firewall Rules With iRules LX

This lab introduces iRules Language eXtensions (LX) or iRulesLX which enables node.js on the BIG-IP platform. The lab uses Tcl iRules and JavaScript code to make a MySQL call to look up a client IP address providing access control in the Multi-Layered Firewall.

This could be useful in developer driven / devops environments where the development team can modify firewall policies simply by updating a database.

Warning: IP addresses in screenshots are examples only. Please read the step-by-step lab instructions to ensure that you use the correct IP addresses.

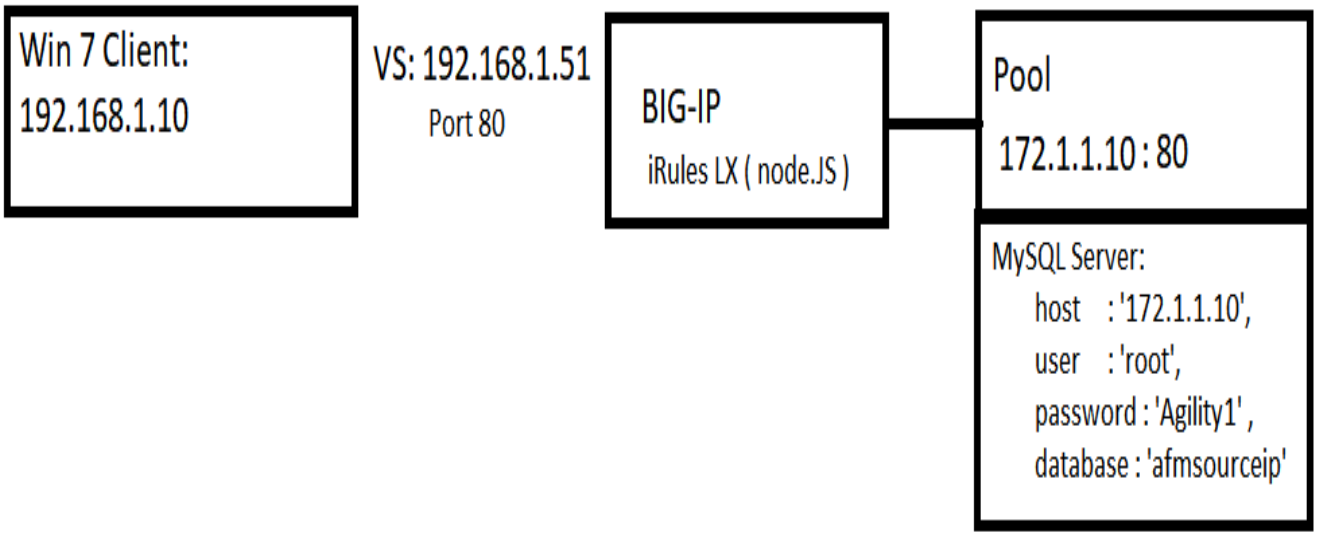
2.2.1 AFM with iRules LX

Estimated completion time: 15 minutes

Beginning in TMOS 12.1 BIGIP offers iRules LX which is a node.js extension to iRules IRules LX does not replace iRules, rather allows iRules to offer additional functionality. In this lab you see how iRules LX can be used to look up client ip addresses that should be disallowed by AFM.

Note: You do not need skills or knowledge of iRules LX to do this lab. This lab will not go into detail on iRules LX nor will it go into detail on Node.JS, rather, this lab shows an application of this with AFM.

Note: We are using a different set of IP subnets just for this module, as shown in this network diagram:



Note: You should be comfortable creating pools and virtual servers by now. Therefore, the following steps to create pools, virtual servers, and AFM policies are kept brief and to the point.

Create the Pool and VS

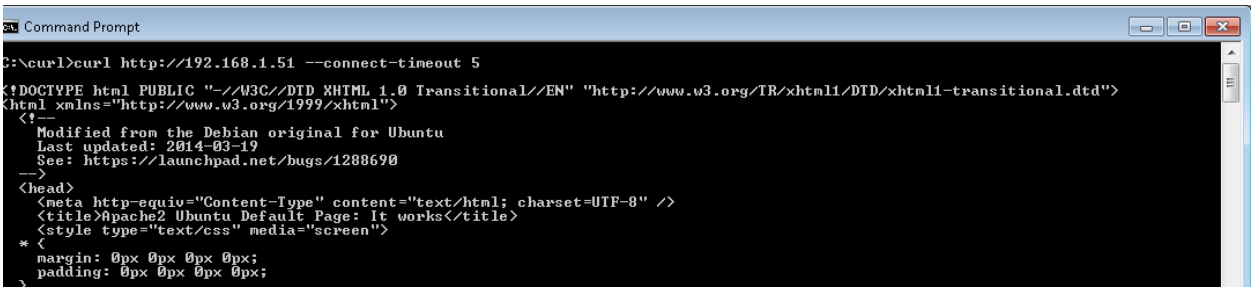
1. Create a pool named afmmysql_pool with one pool member ip address 172.1.1.10 and port 80, and a tcp half-open monitor. Leave all other values default.
2. Create a TCP VS named afmmysql_vs with a destination address of 192.168.1.51, port 80, snat Automap, and set it to use the afmmysql_pool pool. Leave all other values default.

Test the Virtual Server

On the Win7 client, use curl in the cygwin cli (or from the c:\curl directory in a windows command line shell) to test the Virtual Server.

```
curl http://192.168.1.51 --connect-timeout 5
```

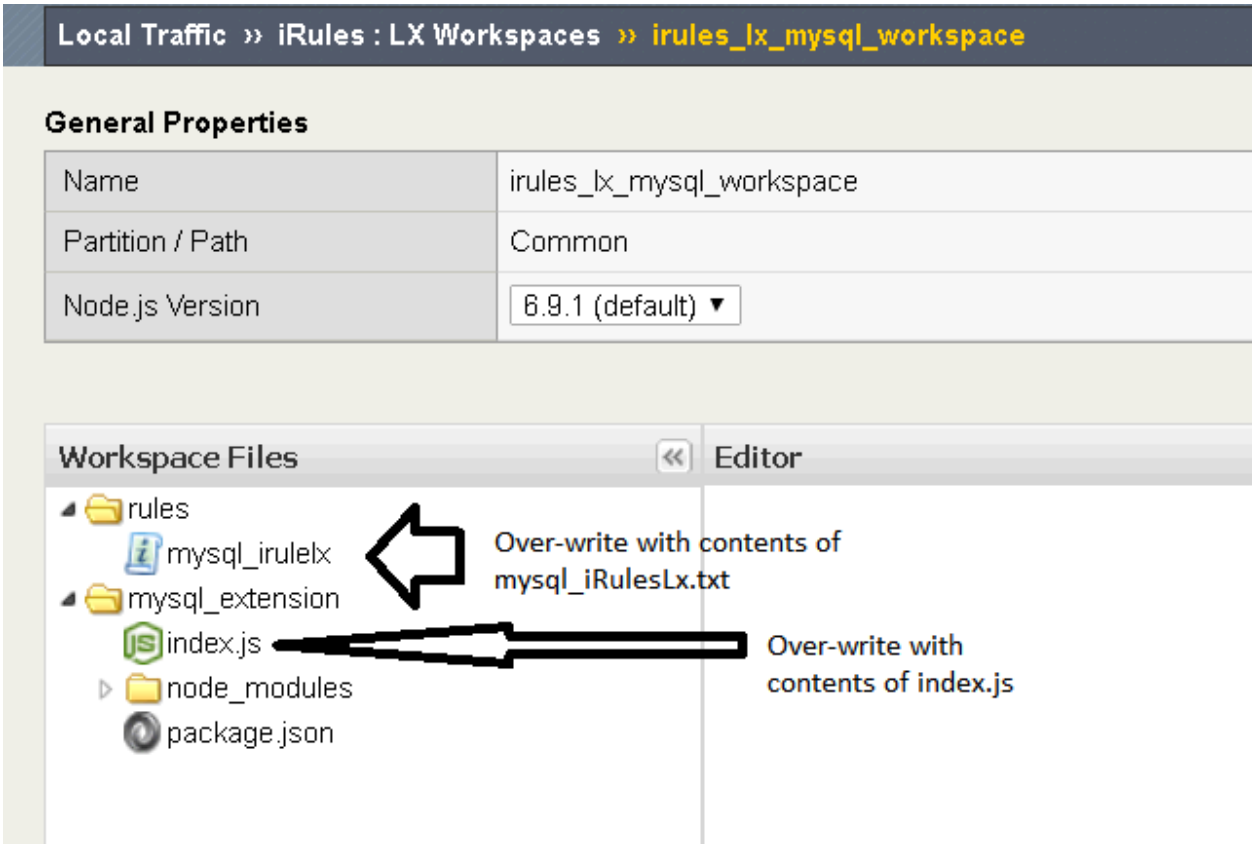
You will notice that you connect, and web page is shown.



Copy & Paste LX Code

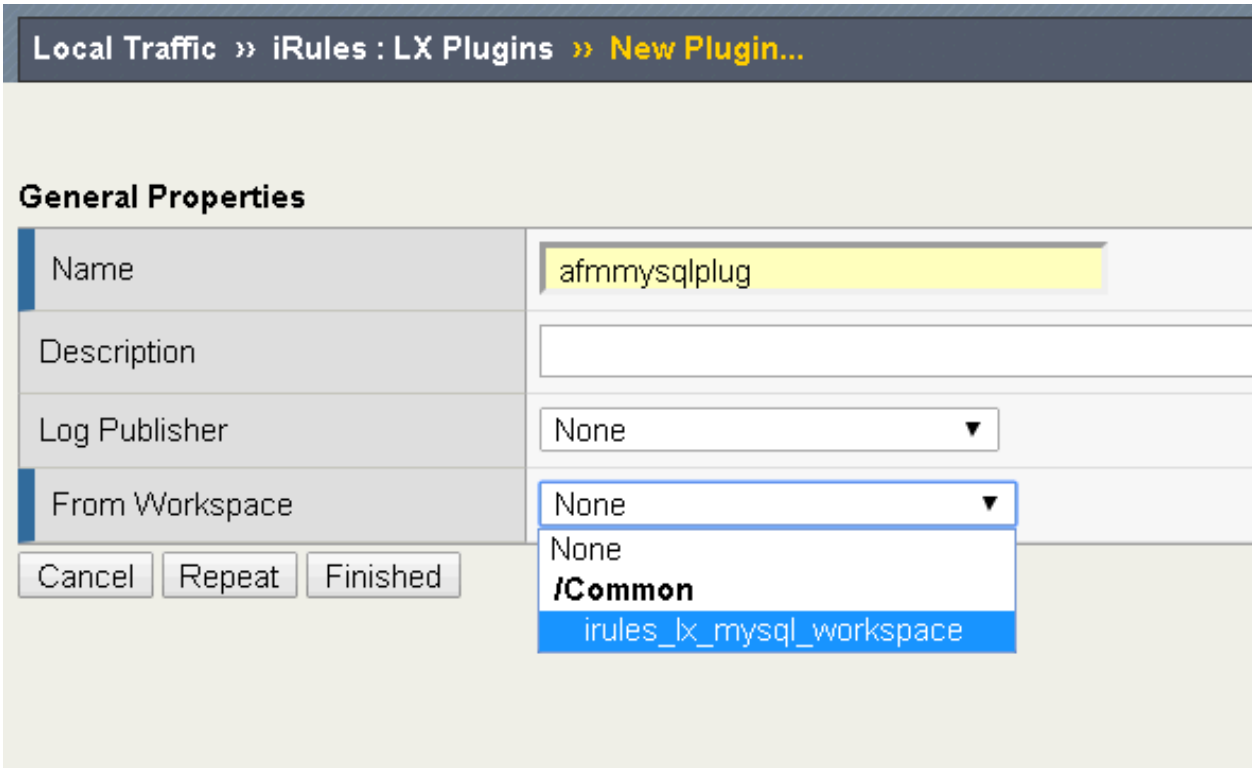
Note: Dont' worry, you're not doing any coding here today. Just a little copy and paste excersize. You are going to copy two files from the Windows desktop and paste them into the iRules LX workspace.

1. **Navigate:** In the BIG-IP webgui, navigate to Local Traffic->iRules-> LX Workspaces-> irules_lx_mysql_workspace
2. Open the mysql_iRulesLx.txt file in Notepad (located on the Windows Desktop) and copy (Ctrl-C or use Mouse) the entire contents
3. In the Big-IP webgui, Click on rules->mysql_irulelx
4. Replace the contents of this with the text you just copied from the mysql_irulesLx.txt file.
5. Click "Save File"
6. In Windows, open the index.js file located on the Desktop (it should open in NotePad), select all, and copy (Ctrl-C or use Mouse) its entire contents.
7. In the Big-IP gui, click on mysql_extension/index.js. Replace the contents of mysql_extension/index.js with the contents of the index.js that you just copied.
8. Click "Save File"



Create LX Plug-In

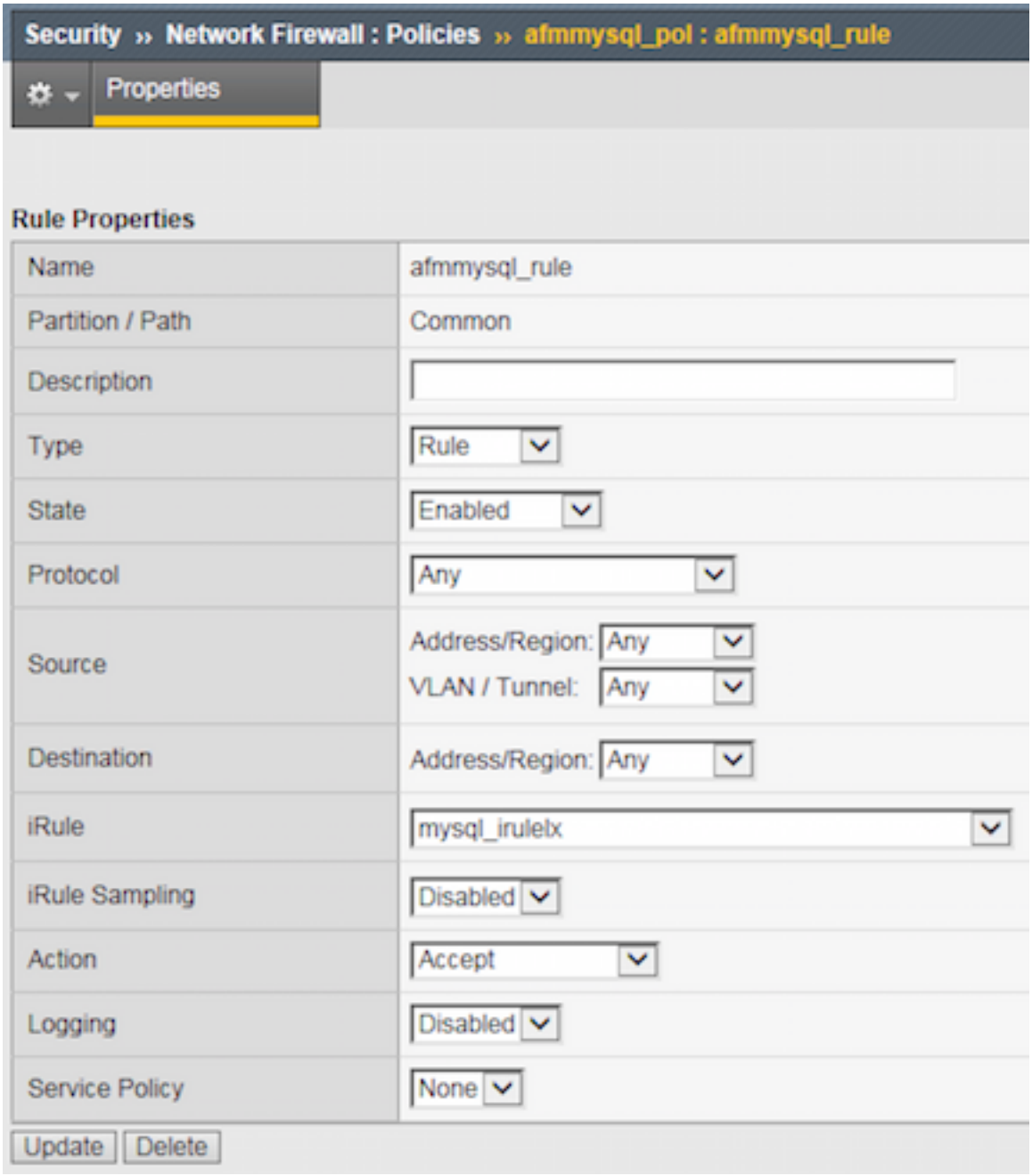
1. **Navigate:** to Local Traffic->iRules-> LX Plugins and create a new LX Plugin named "afmmysqlplug" using the workspace (From Workspace dropdown) irules_lx_mysql_workspace.
2. Click "Finished"



Create a new AFM Policy to use this LX Rule

Note: You are assumed to be pretty familiar with creating AFM policies by now, hence the following steps are kept brief and to the point.

1. Create a new AFM policy named afmmysql_pol
2. Add a rule named afmmysql_rule and click iRule to assign the “mysql_lrulelx” iRule.



3. Click “Finished”
4. Assign this rule to the afmmysql_vs virtual server.

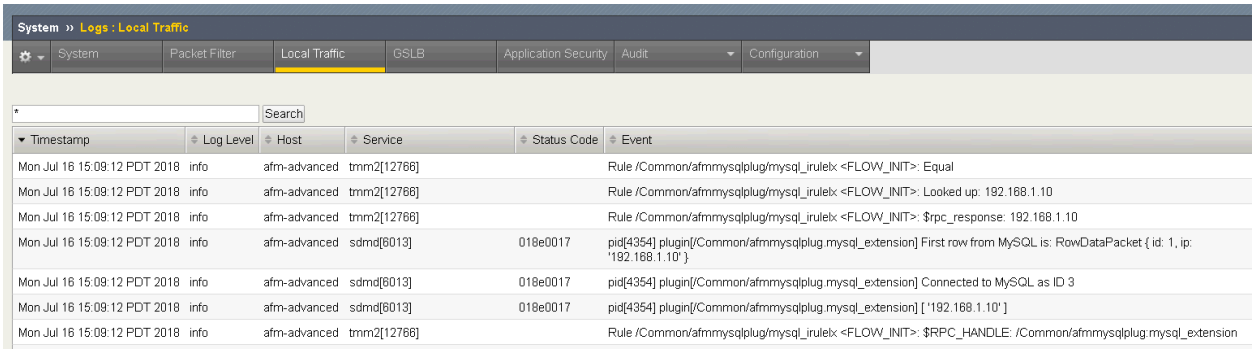
Test the VS with the LX Rule in Place

On the Win7 client, use curl in the cygwin cli (or from c:\curl directory in a windows command line shell) to test that the client is being blocked, as the Win7 client’s ip is in the mysql database.

```
curl http://192.168.1.51 --connect-timeout 5
```

If everything went successfull, this should now timeout.

Attention: Ensure that the iRule is working properly, by going back to the AFM rule and setting the iRule back to None. Also examine the log files at `/var/log/ltm` on the BIG-Ip (or look in the GUI Log as shown here)



Note: This completes Module 3 - Lab 1

2.3 Module 3: AFM Protocol Inspection IPS

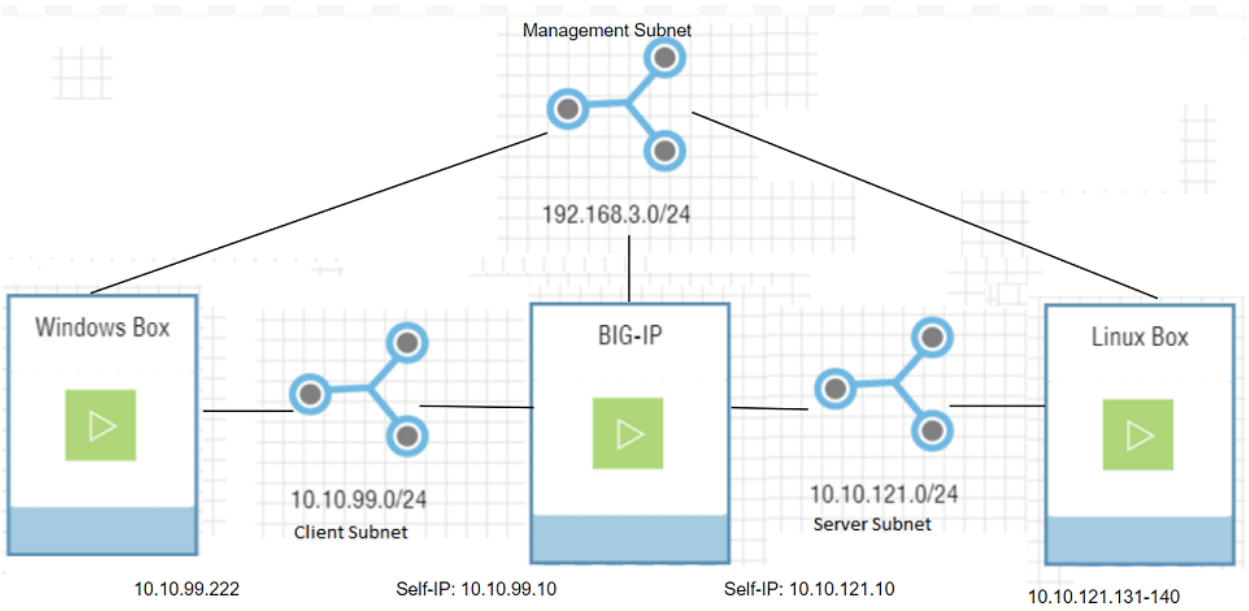
In this lab you will explore the new Intrusion Prevention System feature in 13.1.X, which is called Protocol Inspection.

Protocol Inspection includes Compliance Checks and Signatures. This lab will introduce both, including a section on writing custom Signatures.

2.3.1 Lab 1: Preconditions

Estimated completion time: 15 minutes

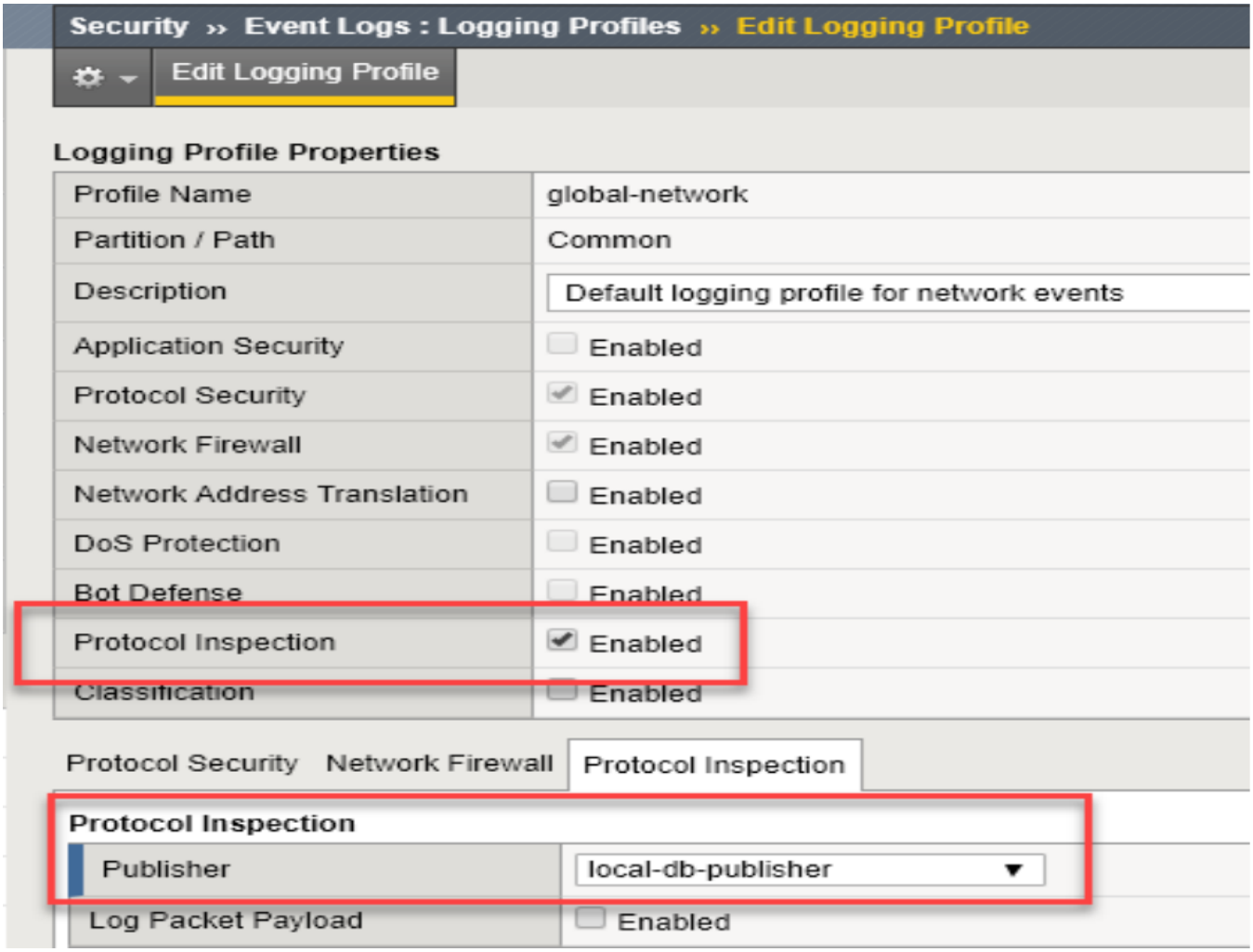
Diagram for Module 4:



There are some steps we need to complete to get the system to work as expected. We're going to get more feedback if we enable logging.

Task 1: Enable Logging for Inspections

1. Navigate to Security > Event Logs > Logging Profiles > global-network
2. Enable Protocol Inspection
3. Click the Protocol Inspection tab and select Publisher 'local-db-publisher'
4. Click 'Update'



Note: This completes Module 4 - Lab 1

2.3.2 Lab 2: Protocol Inspection - Compliance Checks

Estimated completion time: Thirty Five 35 minutes

Compliance Checks model protocols and applications and flag deviations from the model. End users can't add compliance checks, but some of them have parameters the user can modify. We'll look at a couple of these checks and modify one. Have fun!

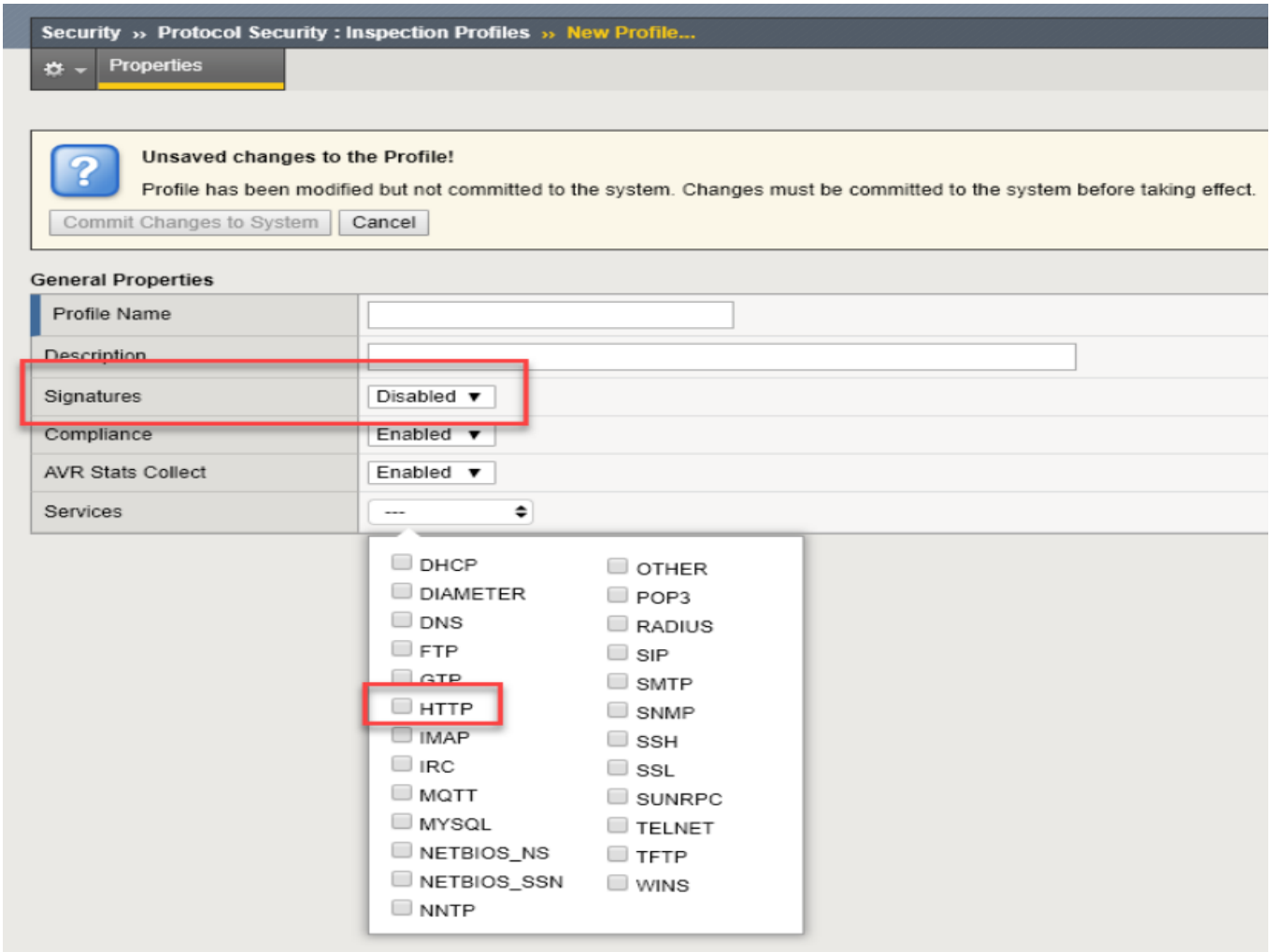
Task 1: The Inspection Profile

You will create an Inspection Profile containing compliance checks.

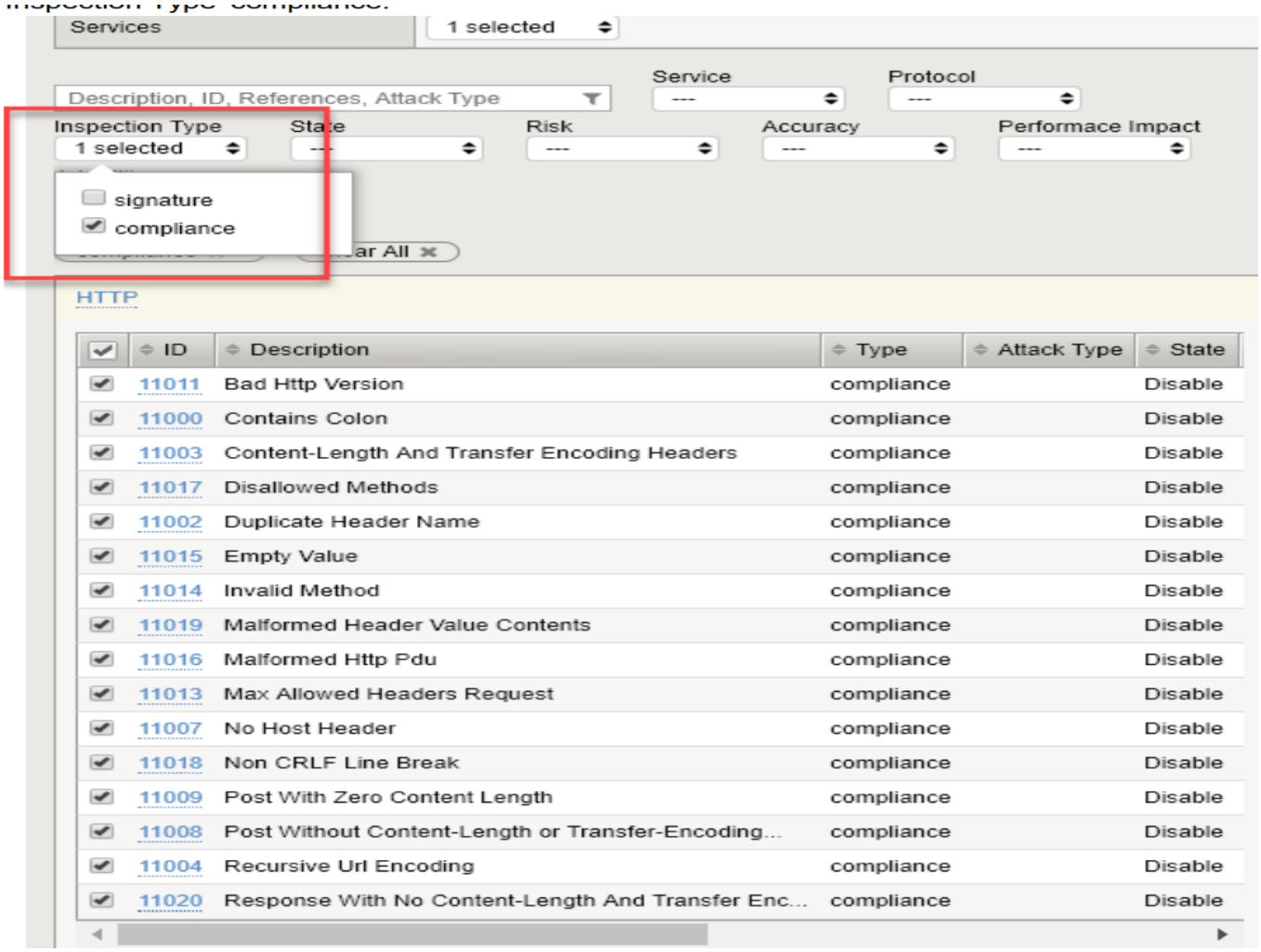
1. Navigate to Security > Protocol Security > Inspection Profiles and click 'Add', select 'New'
2. Name the profile 'my-inspection-profile'
3. Disable Signatures

4. Make sure Compliance is enabled.
5. Under Services, Select HTTP.

Note: You have to wait a few seconds after selecting HTTP

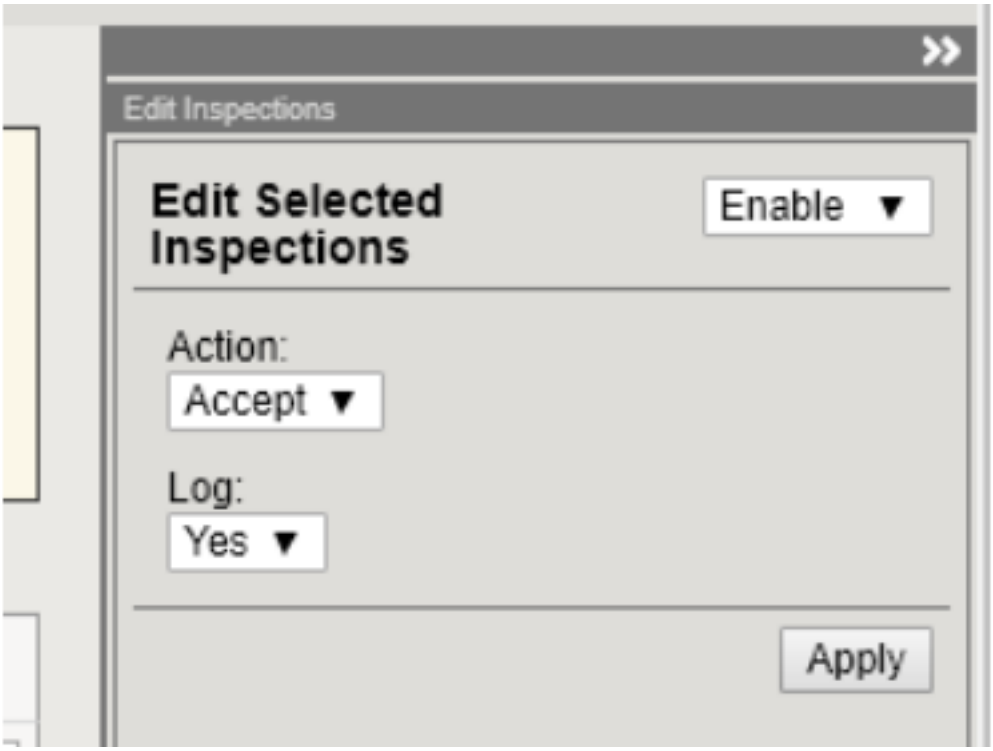


6. When the HTTP Service appears, click to open the Inspection list for HTTP, and select Inspection Type ‘compliance.’



7. Click the checkbox to select all the HTTP compliance checks.
8. In the edit window in the upper-right of the F5 GUI, make the following selections:
   - Enable the selected inspections
   - Set the 'Action' to 'Accept'
   - Enable logging

Note: These should be the default actions, so they most likely are already set for you.



- Click ‘Apply’

9. Click ‘Commit Changes to System’

You should now have an Inspection Policy.

Task 2: Apply the Profile to the Global Policy

1. Navigate to Security > Network Firewall > Active Rules
2. Change Context to ‘Global’
3. Click ‘Add Rule’
4. Make a new policy named ‘global-fw-policy’
5. Make a new rule named fw-global-http-inspection’
6. Configure the new rule:
   - Protocol ‘TCP’
   - Set the Destination port to 80
   - Action ‘Accept’
   - Protocol Inspection Profile: ‘my-inspection-profile’
   - Enable logging
7. Click Save

General Properties

Context	Global
Policy Type	Enforced
Policy	New... Name: global-fw-policy

Rule Properties

Name	fw-global-http-inspection
Description	
Order	Last
Type	Rule
State	Enabled
Protocol	TCP 6
Source	Subscriber: Any Address/Region: Any Port: Any VLAN / Tunnel: Any
Destination	Address/Region: Any Port: Specify... Port Port Range Port List 80 Add 80 Edit Delete
iRule	None
Action	Accept
Send to Virtual	None
Logging	Enabled
Service Policy	None
Protocol Inspection Profile	my-inspection-profile

Task 2.5: Create testing Virtual server on port 80

To get an understanding of how the IPS function works, we need the manual commands we can issue via Telnet. Because Telnet does not work very well with SSL, we need to create a virtual server on port 80 instead of the one on 443 that we have been using so far. Remember this is only for testing, and the IPS functionality can work perfectly well on encrypted traffic (as long as we terminate the SSL)

1. Check if the pool "pool_www.mysite.com" exists. Does it already exist? Only if it does not exist, please create it as follows:

Name	Health Monitor	Members	Service Port
pool_www.mysite.com	tcp_half_open	10.10.121.129	80

2. Create a virtual server with no HTTP profile. Use the following settings, leave everything else default.

Parameter	Value
name	IPS_VS
IP Address	10.10.99.40
Service Port	80
SNAT	automap
Pool	pool_www.mysite.com

Note: Note that we neither applied an Inspection Policy to this VS, nor did you apply a Firewall Policy to this VS. And yet, the IPS is now functional on this VS. Can you think why this is? This is because the global firewall policy is in affect, and the Inspection Policy will be invoked by the Global Firewall Policy.

Task 3: Test the Inspection Profile

1. From the Cygwin session, or from the DOS prompt, enter this command:

```
telnet 10.10.99.40 80
```

The expected output is:

```
Trying 10.10.99.40...
Connected to 10.10.99.40
Escape character is '^]'.
```

Enter the following (Suggestion: copy and paste):

```
GET /index.html HTTP/5
```

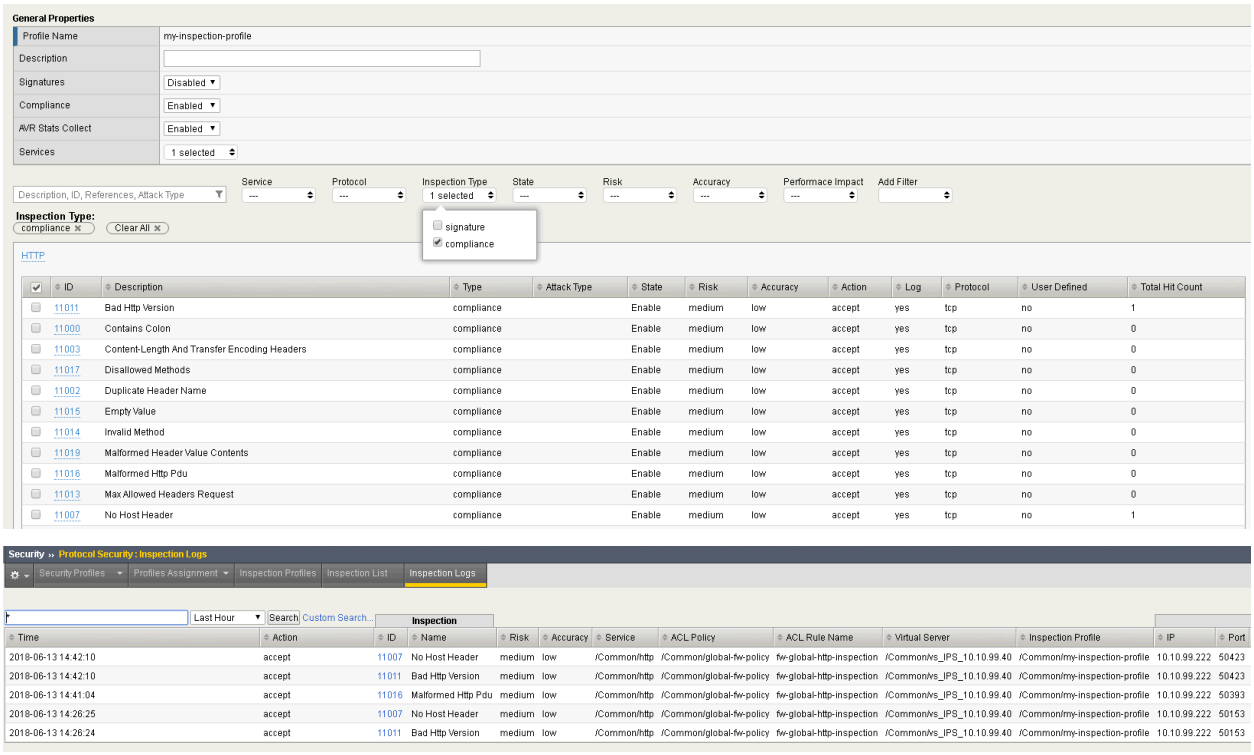
(hit Enter key two times)

The expected HTTP response is:

```
HTTP/1.1 200 OK
( and lots more HTTP headers, etc.)
```

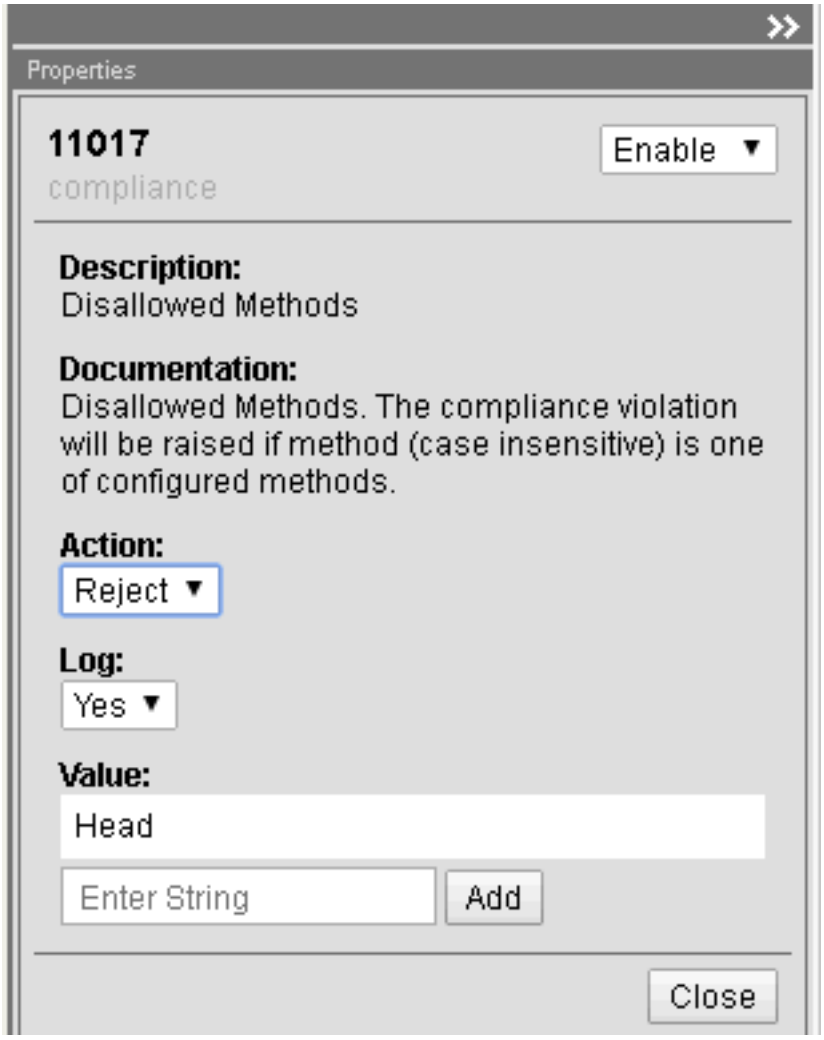
2. Check the results.
   - Navigate to Security > Protocol Security > Inspection Profiles > my-inspectionprofile
   - Filter for Inspection Type 'compliance'

- Look at the Total Hit Count for HTTP Compliance Check ID 11011 “Bad HTTP Version.” We expect to see a hit count of at least 1, and a missing host header count of at least 1.
- Look at the protocol inspection logs. Go to Security > Protocol Security > Inspection Logs. You can see the incoming ip address and port, among other things.



Task 4: Modify a Compliance Check

1. Select Compliance Check 11017 ‘Disallowed Methods’
2. Enter the value “Head” and click ‘Add’



3. Click 'Commit Changes to System'

Task 5: Test the Modified Compliance Check

1. From the Cygwin session, enter (or copy and paste) this command:

```
telnet 10.10.99.40 80
```

The expected output is:

```
Trying 10.10.99.40...
Connected to 10.10.99.40
Escape character is '^]'.
```

Enter the following (Suggestion: copy and paste):

```
HEAD /index.html HTTP/1.1
```

Expected output:

```
HTTP/1.1 400 Bad Request
```

2. Check the results.

Note: Just an interesting point to make again, this is the IPS code checking HTTP, not the HTTP Profile (This VS does not have an HTTP Profile)

- Navigate to Security > Protocol Security > Inspection Profiles > my-inspection-profile
- Filter for Inspection Type 'compliance'
- Look at the Total Hit Count for HTTP Compliance Check ID 11017 "Disallowed Methods." You may have to refresh the page.
- We expect to see a hit count of 1.

4. Look at the stats. Enter the following command on the Big-IP command line:

```
tmsh show sec proto profile my-inspection-profile
```

We expect to see a Hit Count of at least 1 (more if you've done it multiple times).

```
[root@afm301:Active:Standalone] config # tmsh show sec proto profile my-inspection-profile

-------------------------------------------------------------------------------------------------------------------------------
Security::Protocol Inspection::Profile
-------------------------------------------------------------------------------------------------------------------------------
Profile Name                        Inspection Id          Inspection Name            VS Name  Hit Count      Last Hit Time
-------------------------------------------------------------------------------------------------------------------------------
my-inspection-profile                      11007      http_no_host_header  vs_IPS_10.10.99.40          3  06/13/18 15:13:58
my-inspection-profile                      11011        http_bad_version  vs_IPS_10.10.99.40          2  06/13/18 14:42:10
my-inspection-profile                      11016       http_malformed_pdu  vs_IPS_10.10.99.40          1  06/13/18 14:41:04
my-inspection-profile                      11017  http_disallowed_methods  vs_IPS_10.10.99.40          1  06/13/18 15:14:34

[root@afm301:Active:Standalone] config #
```

Note: This completes Module 4 - Lab 2

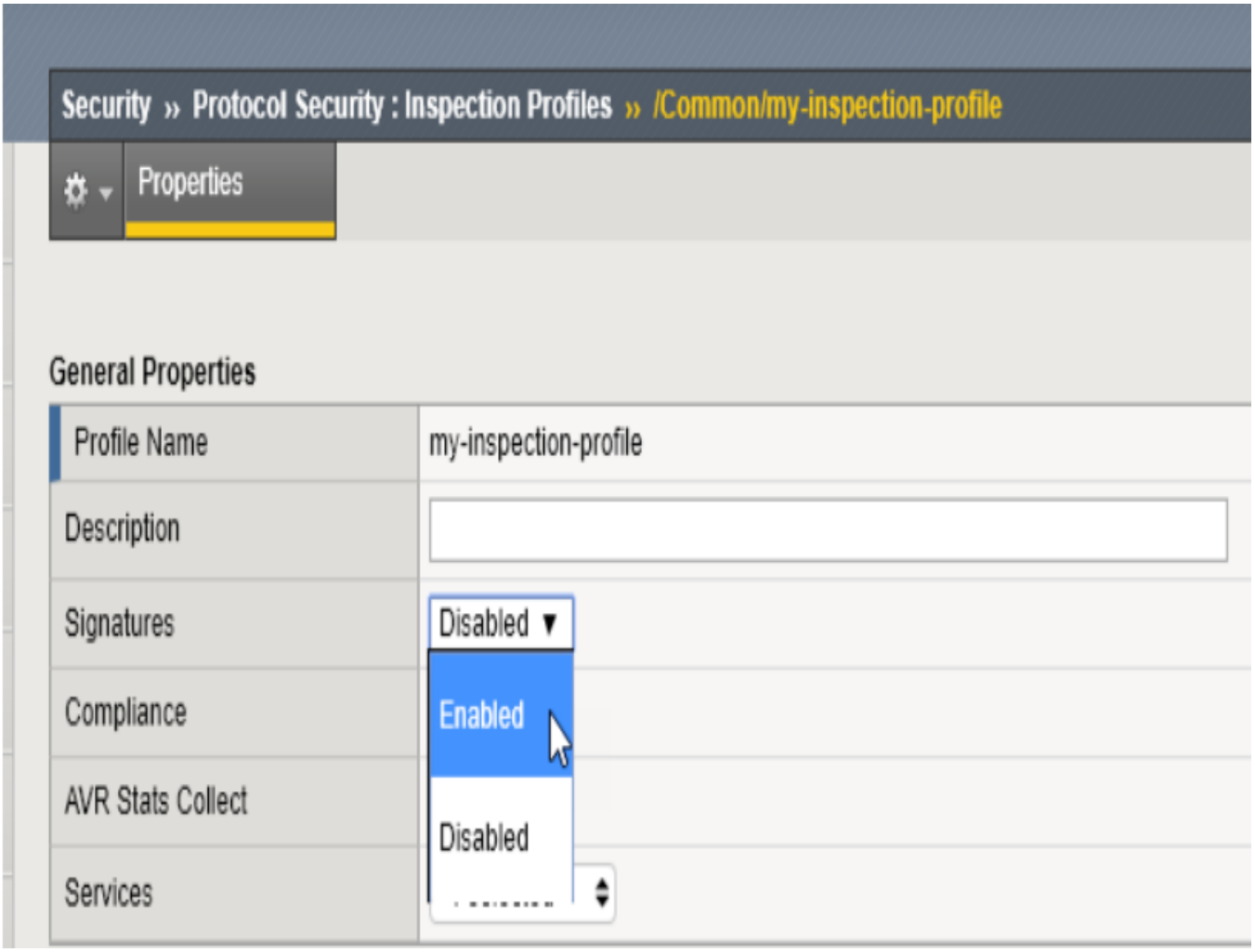
2.3.3 Lab 3: Protocol Inspection - Signatures

Estimated completion time: Five 5 minutes

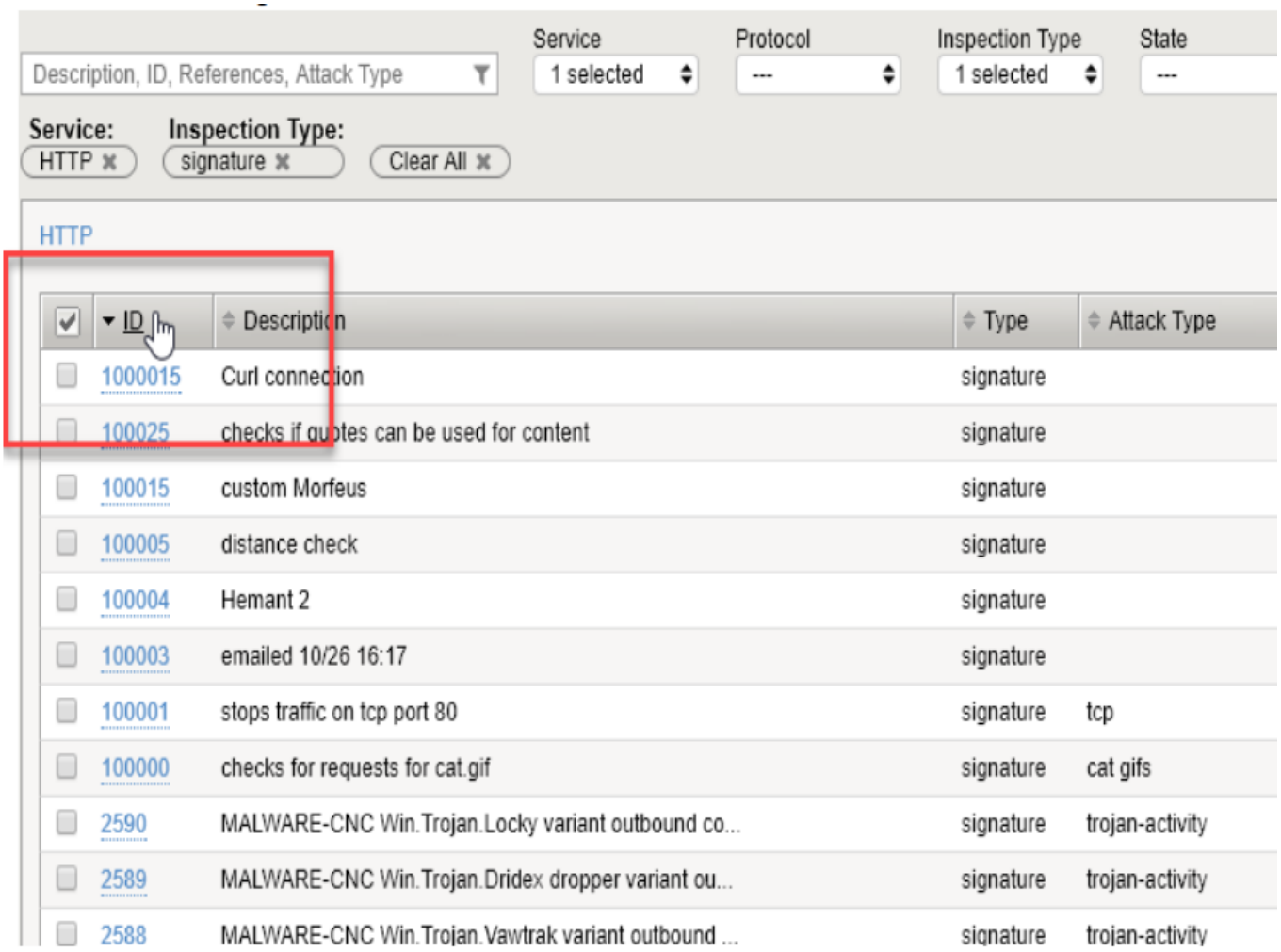
Signature Checks can be written by the user, unlike Compliance Checks which are programmatic inspections provided only by F5. We'll start with a lab procedure that explores the use of the provided signatures.

Task 1: Enabling Signatures

1. Navigate to Security > Protocol Security > Inspection Profiles > my-inspection-profile
2. Enable Signatures



3. Click ‘Commit Changes to System’
4. Now enable an individual signature
5. Filter on Service ‘HTTP’, Inspection Type ‘signature’
6. Sort the filtered signatures in reverse order of ID. Click the ID column twice.



c. Scroll down to 2538 and click to edit.

d. Configure the signature:

i. Enable

ii. Action: Reject

iii. Log: Yes

iv. Click 'Close'

v. Click 'Commit Changes to System'

You should now have an enabled HTTP signature. We don't know exactly what it's checking for, but we'll get to that in the next Procedure.

Task 2: Reviewing the actual pattern check

The UI currently doesn't give you the exact pattern being checked for in a Signature. We will search the file where the default signatures are defined and review the one with signature id 2538.

1. From the BIG-IP command line, enter the following command:

```
grep 2538 /defaults/ips_snort_signatures.txt
```

The expected output is:

alert tcp any any -> any any (content:"User-Agent|3A 20|Vitruvian"; fast_pattern:only; http_header; sig_id:2538;)

The Signature is looking for TCP traffic with http_header contents "User-Agent: Vitruvian"

Task 3: Test the Signature

1. From the Desktop terminal, issue the following command:

curl -A Vitruvian http://10.10.99.40/cat.gif

This uses curl which you area already familiar with, and specifies the USER-AGENT = "Vitruvian"

The expected output is:

curl: (56) Recv failure: Connection reset by peer

2. Check the results: refresh the Inspection Profiles page, filter as needed, sort as needed, and review the Total Hit Count for Signature ID 2538.
3. Since that is a pain, use the BIG-IP command line:

tmsh show sec proto profile my-inspection-profile

We expect to see a Hit Count of 1 for Inspection ID 2538.

This was a simple test of a simple pattern match. There are some tricks to testing signatures with more elaborate patterns, which we'll explore in the final lab.

Note: This completes Module 4 - Lab 3

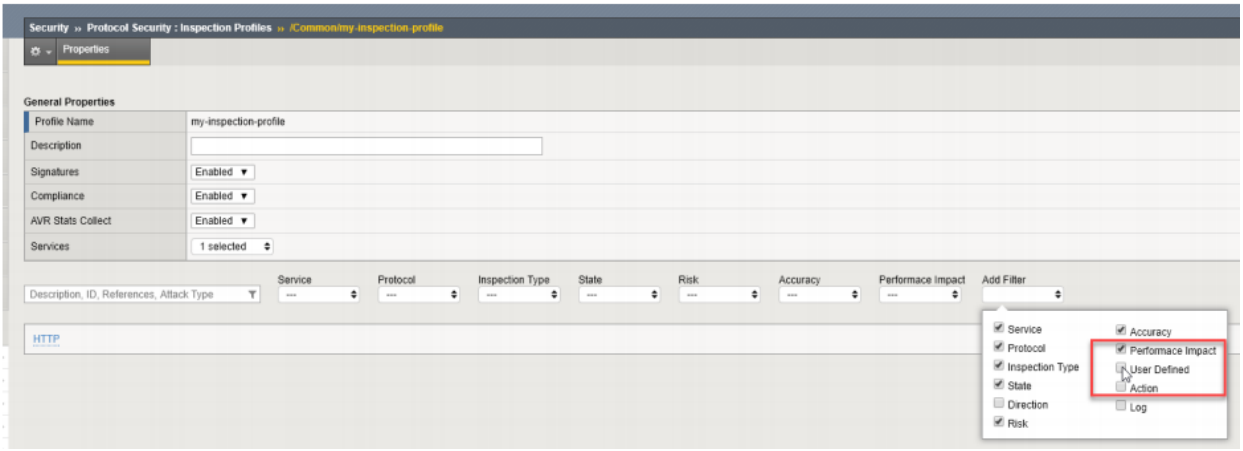
2.3.4 Lab 4: Protocol Inspection - Custom Signatures

Estimated completion time: 15 minutes

You can write custom signatures using a subset of the Snort® rules language. We'll walk through a couple of examples, but the intent is not to make you an expert. At most we can give you a head start in developing expertise. We'll start with a scenario: we want to detect sessions requesting a particular URI, /images/cat.gif where the User-Agent is "Attack-Bot-2000" When working with signatures, keep in mind there are just under 1600 signatures shipping with 13.1.0. It will be easier to work with custom signatures if you add a filter for them.

Task 1: Set Filter

1. Edit the Inspection Profile 'my-inspection-profile' Click 'Add Filter' and select 'User Defined'
2. When the User Defined filter is added, select 'yes'



Task 2: Cargo Cult Signature Authoring - finding an example to copy

It's often more pragmatic to modify an example that is close to what we want than to start from scratch. Let's start with a very simple example.

From the BIG-IP command line, issue the following command:

grep 1189 /defaults/ips_snort_signatures.txt

Expected output:

alert tcp any any -> any any (content:"/rksh"; fast_pattern:only; http_uri; sig_id:1189;)

Parsing this, there is a Header section and an Options section. The Header is the stuff outside the parenthesis:

alert means "match" or "do something." The BIG-IP/AFM Inspection Policy will actually determine what is done with a packet that matches a signature, so it doesn't matter which action you choose. For the greatest clarity, standardize on "alert" so you don't confuse others or yourself.

tcp is the L4 protocol. The Signature has a Protocol setting outside the signature definition. They should probably agree, don't you think?

any any -> any any means "FROM any source IP+port TO any destination IP+port." We will tighten this up in a later lab procedure. Note that the signature has its own direction outside the signature definition. We probably want to avoid a conflict between these direction settings.

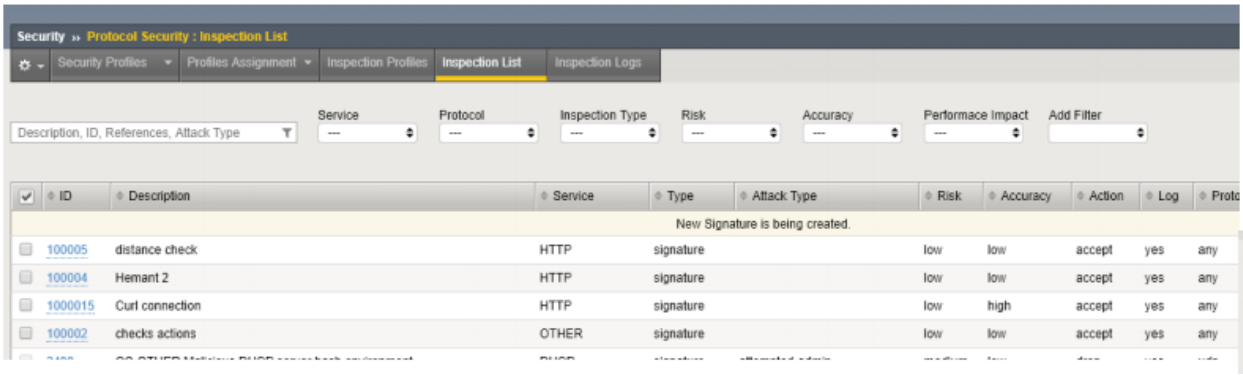
The Options are the elements inside the parenthesis. Each option is a Type: value pair, separated by a colon. Each Option is separated by a semicolon. The options in this example are:

- content - This is the pattern to match, in this case "/rksh."
- fast_pattern - applies to the previous content definition. It's intended to be used to prequalify a rule for further processing. If you have a bunch of expensive content checks, you can look for one characteristic string to see if you need to bother with the others. In this example the effective meaning is "If you see this, look into the other content to see if we match" but there's no other content! The key takeaway is that the rules provided are not optimized. We'll try to do better when we create our own.
- http_uri - also applies to the previous content definition. It restricts the search to the HTTP Uniform Resource Identifier.
- sig_id - the signature id

Task 3: Adapting our example in creating a custom signature

We're going to run into a problem that stems from MCPD parsing the contents of /defaults/ips_snort_signatures.txt differently than the UI parses custom signatures.

1. Create a new custom signature. Navigate to Security > Protocol Security > Inspection List and click "New Signature"



2. Enter the following:

a.Name - this is an odd field in that it doesn't show up in the Signatures page but it is the object name in the config.

Enter "no cat gif"

b. Description - this *does* show up in the Signatures page, Event Logs, tmsh show output, etc. Make it descriptive, systematic, and concise. Enter "HTTP cat.gif request"

c. Signature Definition - here's the big one. Based on our example, enter:

alert tcp any any -> any 80 (content:cat.gif;http_uri; sig_id:100000;)

This simply swaps the content URI string to match and provides a new signature ID.

d. Click "Create." We expect configuration validation to succeed.

From the Signatures page, open your new signature up for editing to add the rest of the signature elements.

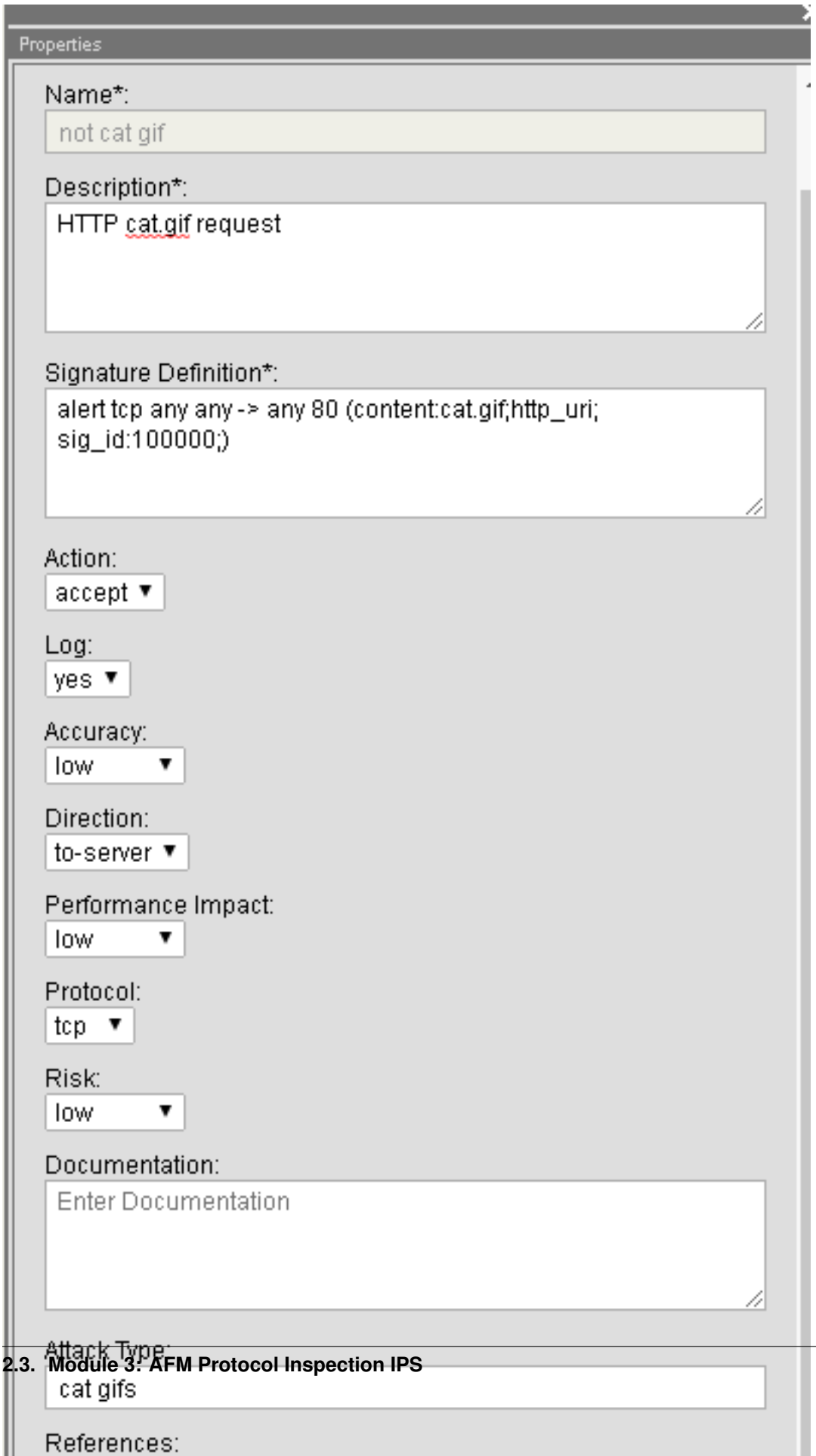
e. Direction: to Server (agreeing with our signature definition)

f. Protocol: TCP (agreeing with our signature definition)

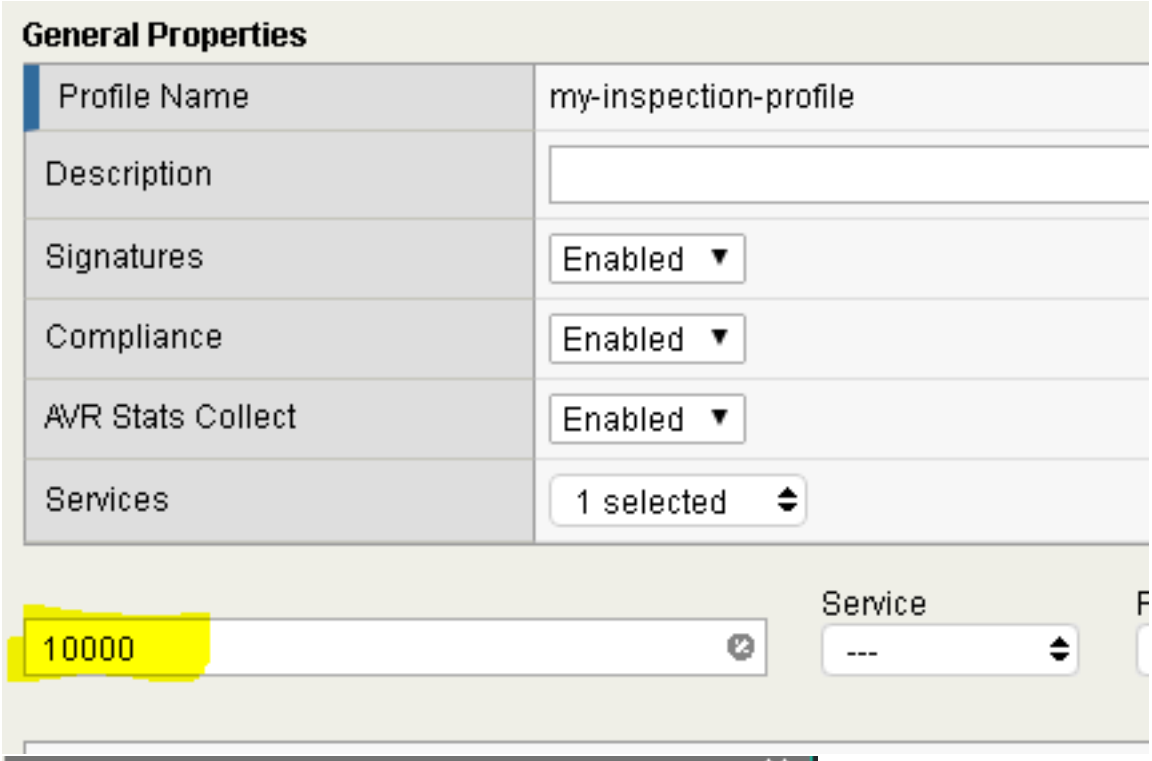
g. Attack type - "cat gifs"

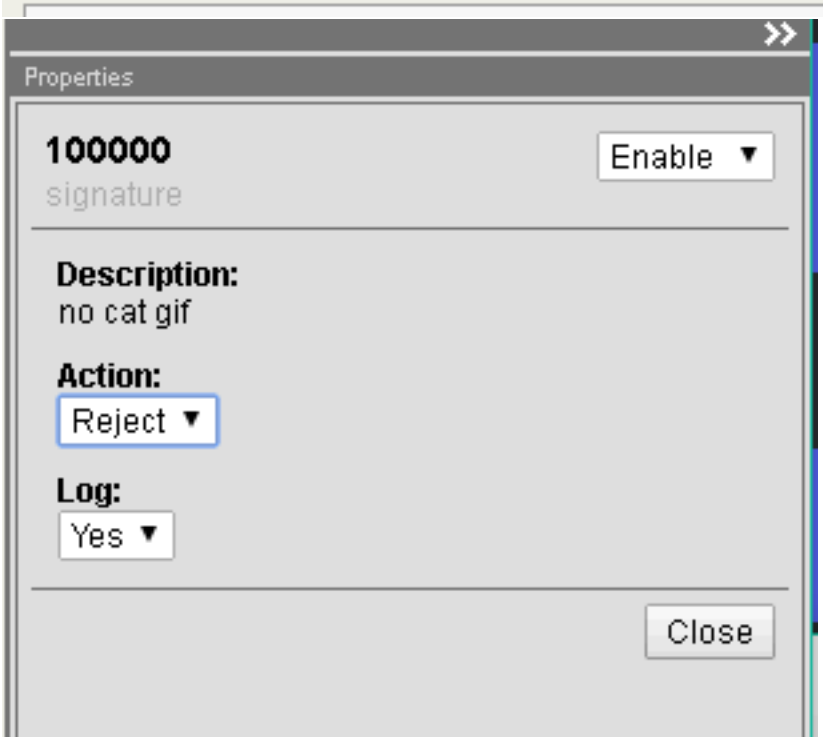
h. Service - select HTTP

i. Click "Save"


Properties
Name*:
not cat gif
Description*:
HTTP cat.gif request
Signature Definition*:
alert tcp any any -> any 80 (content:cat.gif;http_uri;
sig_id:100000;)
Action:
accept
Log:
yes
Accuracy:
low
Direction:
to-server
Performance Impact:
low
Protocol:
tcp
Risk:
low
Documentation:
Enter Documentation
Attack Type:
cat gifs
References:

3. Add this signature to the Inspection Profile my-inspection-profile
   - Navigate to Security > Protocol Security > Inspection Profiles > my-inspectionprofile
   - Select your new signature, 100000, and when the “Edit Inspections” window pops open, set “Action” to “Reject” and click “Apply” (“Enable” and Log: Yes are selected by default.)





c. Click “Commit Changes to Profile”

4. Test it out.

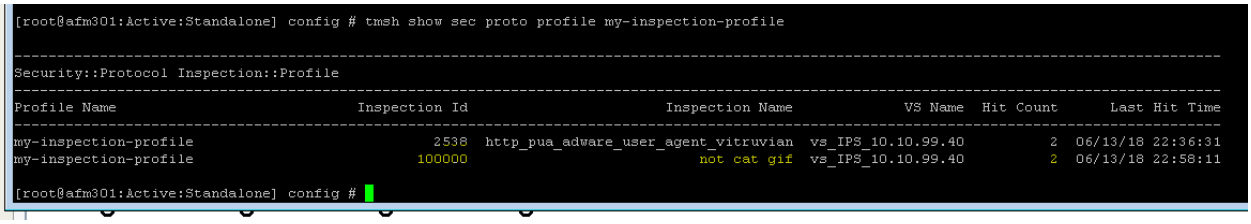
a. From the Desktop terminal, use the following command:

curl -A test http://10.10.99.40/cat.gif

b. Check stats. From the BIG-IP command line:

tmsh show sec proto profile my-inspection-profile

We expect to see a Hit Count of 1 for Inspection ID 100000.



Note: This completes Module 4 - Lab 4

3

Class - F5 BIG-IP DDoS and DNS DoS Protections

This class covers the following topics:

- Detecting and Preventing DNS DoS Attacks on a Virtual Server
- Detecting and Preventing System DoS and DDoS Attacks

Expected time to complete: **2 hours**

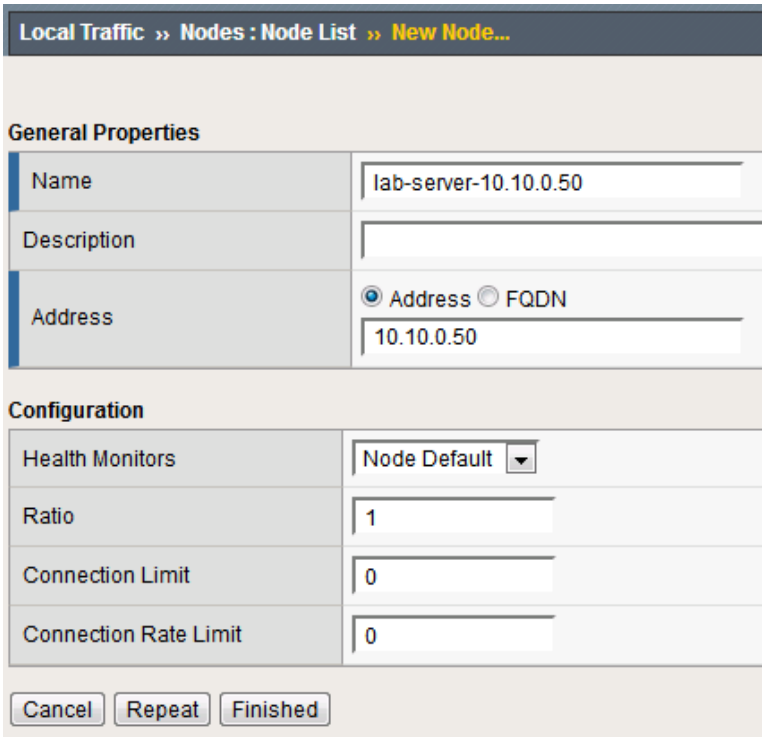
3.1 Module 1 – Detecting and Preventing DNS DoS Attacks on a Virtual Server

In this section of the lab, we'll configure the steps necessary to ensure that the BIG-IP can forward traffic to the back-end server that is hosting our DNS service. We will then attack the resources behind the virtual server, mitigate the attack, and finally review the reports and logs generated by the BIG-IP.

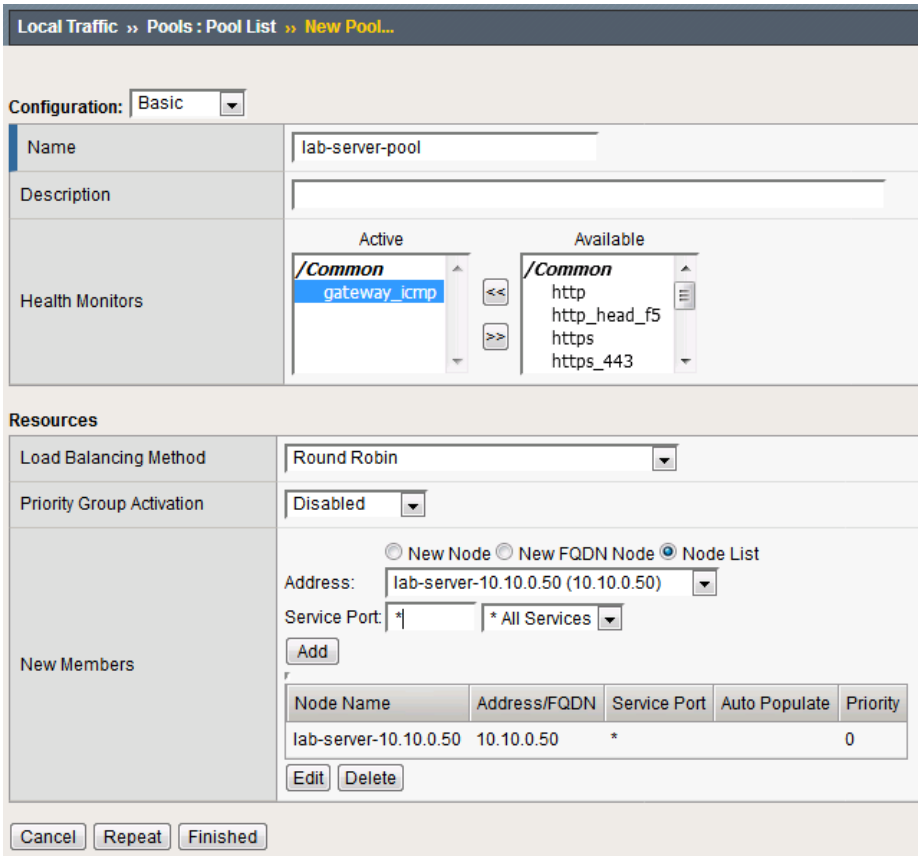
3.1.1 Base BIG-IP Configuration

In this lab, the VE has been configured with the basic system settings and the VLAN/self-IP configurations required for the BIG-IP to communicate and pass traffic on the network. We'll now need to configure the BIG-IP to listen for traffic and pass it to the back end server.

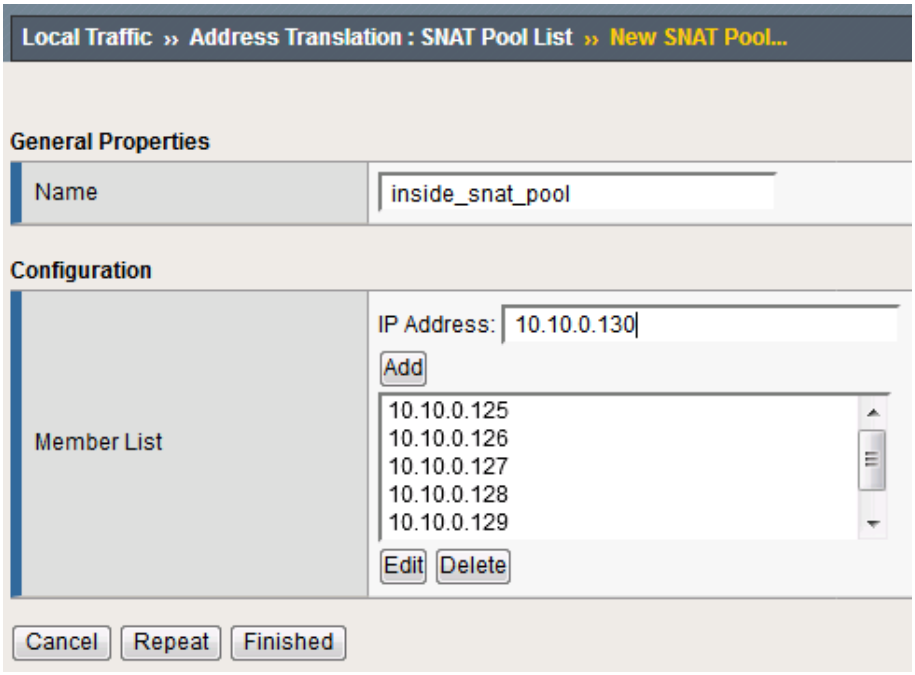
1. Launch the Firefox shortcut titled **Launch BIG-IP Web UI** on the desktop of your lab jump server. The credentials for the BIG-IP are conveniently displayed in the login banner. Just in case: **admin** / **401elliottW!**
2. Navigate to **Local Traffic** > **Nodes** and create a new node with the following settings, leaving unspecified fields at their default value:
    a. Name: lab-server-10.10.0.50
    b. Address: 10.10.0.50



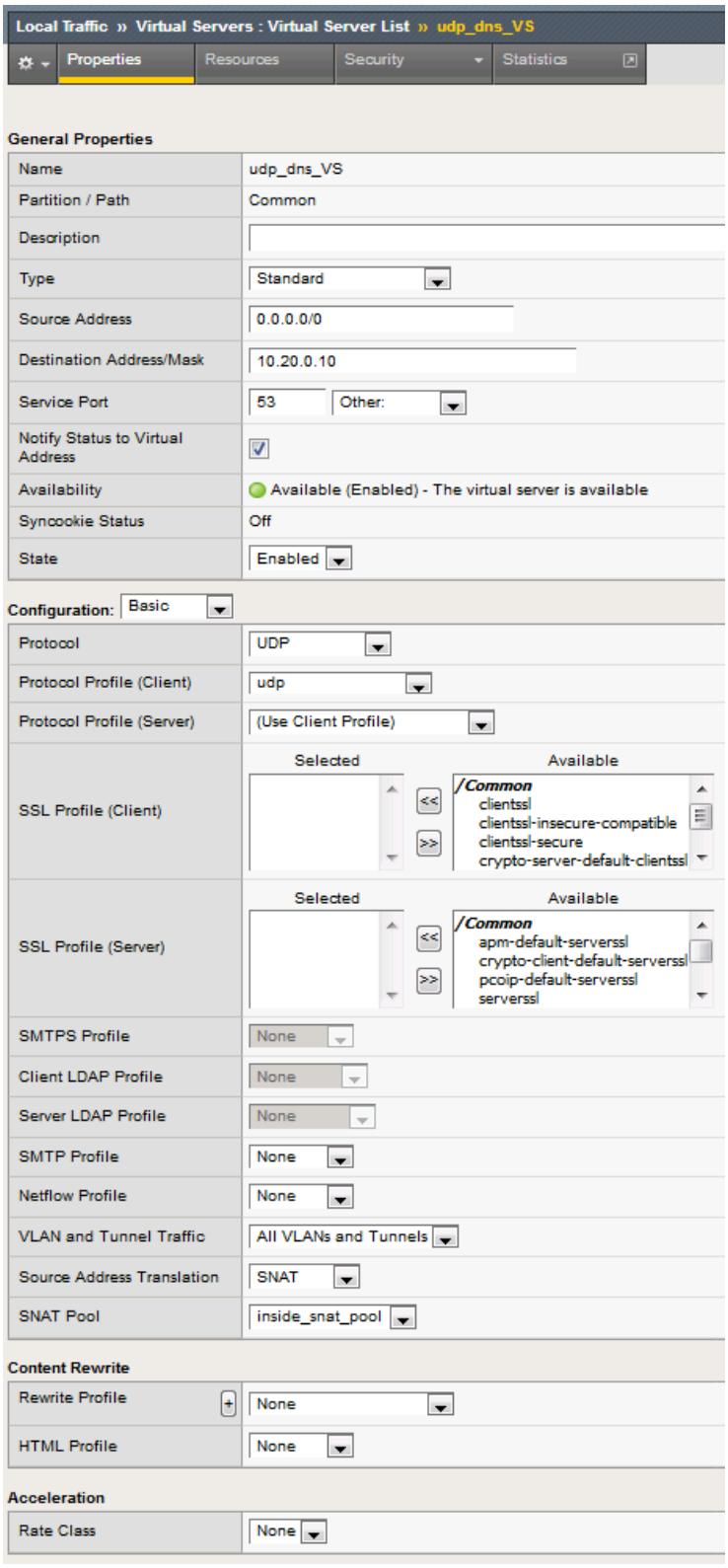
3. Click **Finished** to add the new node.
4. Navigate to **Local Traffic** > **Pools** and create a new pool with the following settings, leaving unspecified attributes at their default value:
    a. Name: lab-server-pool
    b. Health Monitors: gateway_icmp
    c. New Members: Node List - Address: lab-server-10.10.0.50 - Service Port: * (All Ports)
    d. Click **Add** to add the new member to the member list.



5. Click **Finished** to create the new pool.
6. Because the attack server will be sending a huge amount of traffic, we'll need a fairly large SNAT pool. Navigate to **Local Traffic** > **Address Translation** > **SNAT Pool List** and create a new SNAT pool with the following attributes:
   a. Name: inside_snat_pool
   b. Member List: 10.10.0.125, 10.10.0.126, 10.10.0.127, 10.10.0.128, 10.10.0.129, 10.10.0.130



7. Click **Finished** to commit your changes.
8. Navigate to **Local Traffic** > **Virtual Servers** and create a new virtual server with the following settings, leaving unspecified fields at their default value:
    a. Name: udp_dns_VS
    b. Destination Address/Mask: 10.20.0.10
    c. Service Port: 53
    d. Protocol: UDP
    e. Source Address Translation: SNAT
    f. SNAT Pool: inside_snat_pool
    g. Default Pool: lab-server-pool



9. Click **Finished**.
10. We'll now test the new DNS virtual server. SSH into the attack host by clicking the "Attack Host (Ubuntu)" icon on the jump host desktop.

11. Issue the dig @10.20.0.10 www.example.com +short command on the BASH CLI of the attack host. You should see output similar to:

```
ubuntu@dnsclient:~$ dig @10.20.0.10 www.example.com +short
10.10.0.99
```

    This verifies that DNS traffic is passing through the BIG-IP.

12. Return to the BIG-IP and navigate to **Local Traffic** > **Virtual Servers** and create a new virtual server with the following settings, leaving unspecified fields at their default value:

    a. Name: other_protocols_VS

    b. Destination Address/Mask: 10.20.0.10

    c. Service Port: * (All Ports)

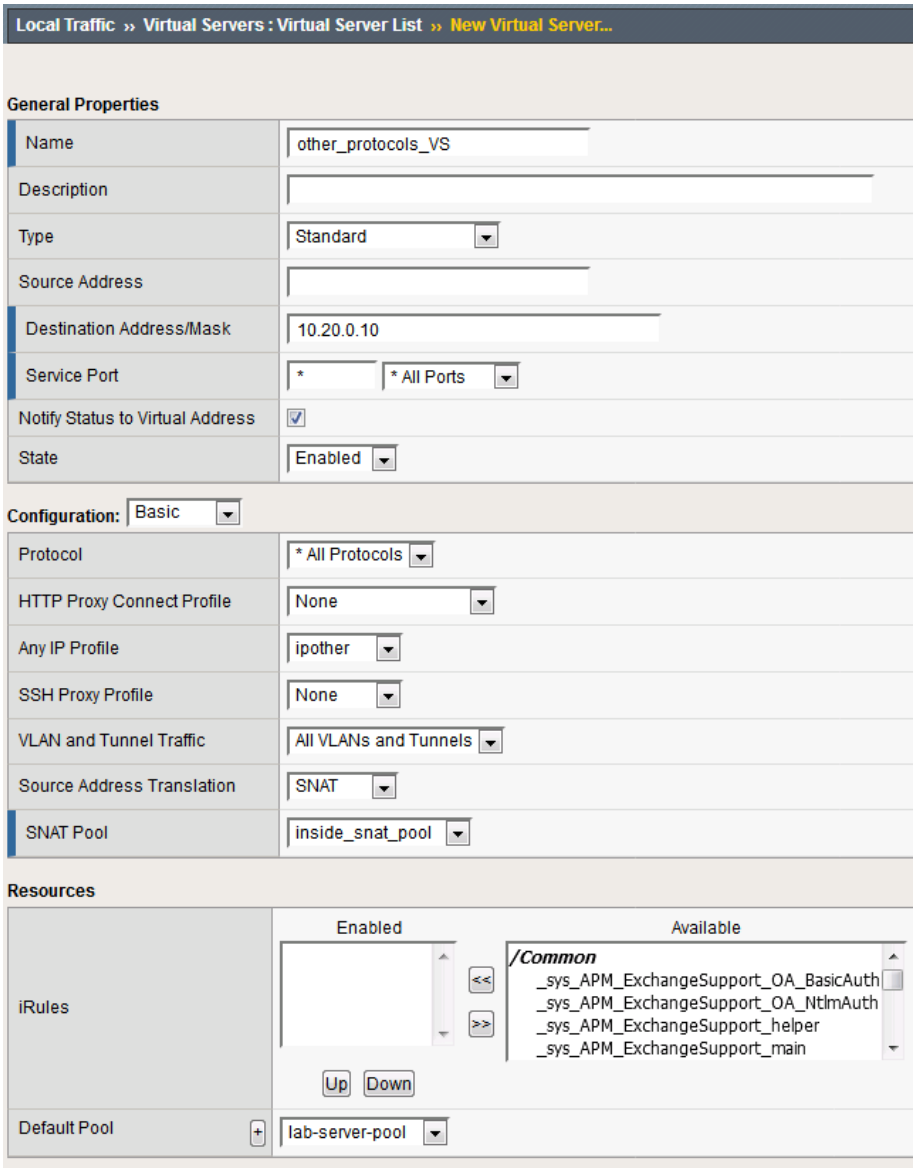
    d. Protocol: * All Protocols

    e. Any IP Profile: ipother

    f. Source Address Translation: SNAT

    g. SNAT Pool: inside_snat_pool

    h. Default Pool: lab-server-pool



13. Return to the Attack Host SSH session and attempt to SSH to the server using SSH 10.20.0.10. Simply verify that you are prompted for credentials and press CTRL+C to cancel the session. This verifies that non-DNS traffic is now flowing through the BIG-IP.

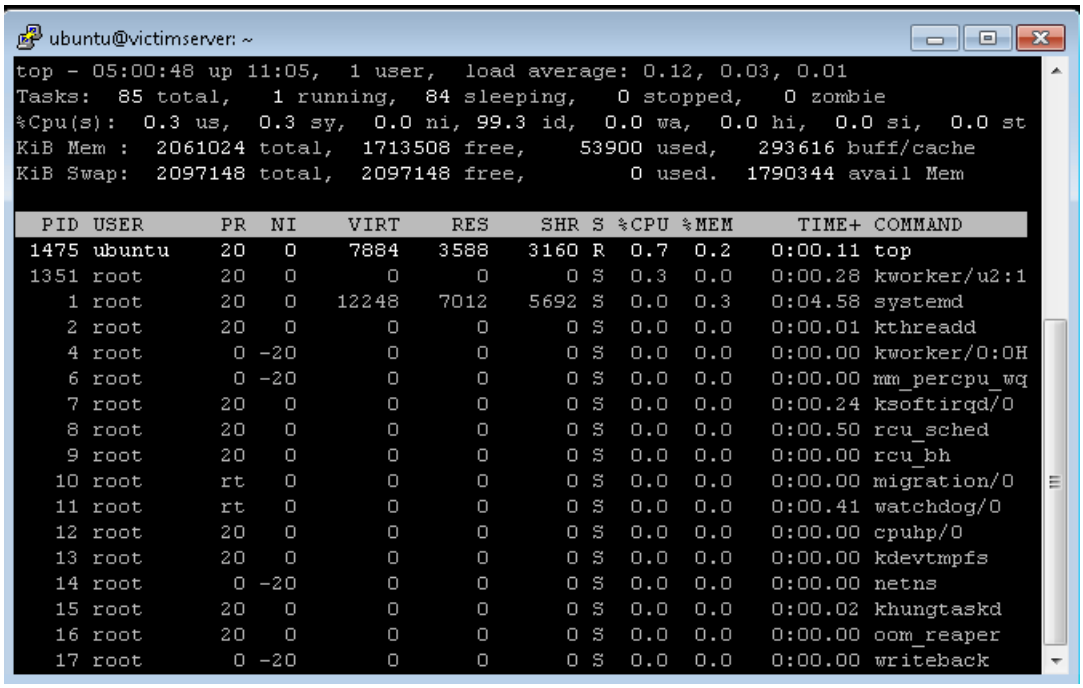
3.1.2 Detecting and Preventing DNS DoS Attacks on a Virtual Server

Establishing a DNS server baseline

Before we can attack our DNS server, we should establish a baseline for how many QPS our DNS server can handle. For this lab, let's find the magic number of QPS that causes 50% CPU utilization on the BIND process.

1. Connect to the Victim Server SSH session by double-clicking the **Victim Server (Ubuntu)** shortcut on the jump host desktop.
2. From the BASH prompt, enter **top** and press **Enter** to start the top utility.

3. You will see a list of running processes sorted by CPU utilization, like the output below:



4. Connect to the Attack Host SSH session by double-clicking the **Attack Host (Ubuntu)** shortcut on the jump host desktop.
5. Start by sending 500 DNS QPS for 30 seconds to the host using the following syntax:
   dnsperf -s 10.20.0.10 -d queryfile-example-current -c 20 -T 20 -l 30 -q 10000 -Q 500

Hint: There is a text file on the desktop of the jump host with all of the CLI commands used in the lab for cut/paste use.

---

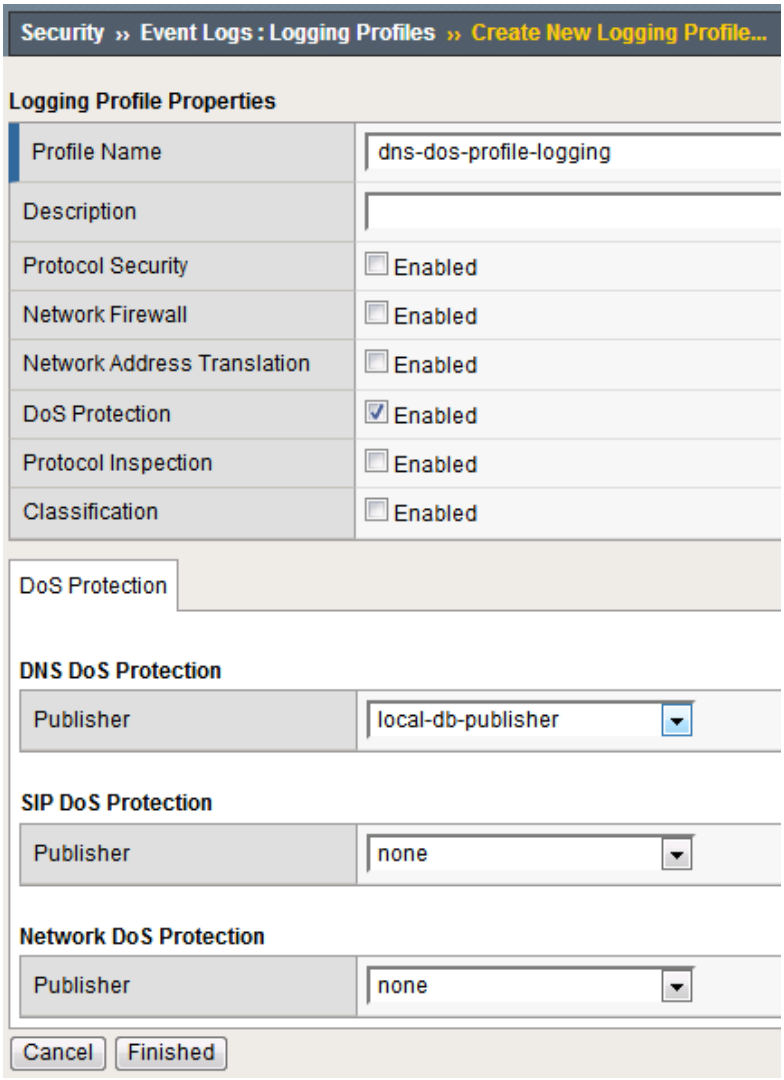
6. Observe CPU utilization over the 30 second window for the **named** process. If the CPU utilization is below 45%, increase the QPS by increasing the -Q value. If the CPU utilization is above 55%, decrease the QPS.
7. Record the QPS required to achieve a sustained CPU utilization of approximately 50%. Consider this the QPS that the server can safely sustain for demonstration purposes.
8. Now, attack the DNS server with 10,000 QPS using the following syntax:
   dnsperf -s 10.20.0.10 -d queryfile-example-current -c 20 -T 20 -l 30 -q 10000 -Q 10000
9. You'll notice that the CPU utilization on the victim server skyrockets, as well as DNS query timeout errors appearing on the attack server's SSH session. This shows your DNS server is overwhelmed.

Configuring a DoS Logging Profile

We'll create a DoS logging profile so that we can see event logs in the BIG-IP UI during attack mitigation.

1. On the BIG-IP web UI, navigate to **Security** > **Event Logs** > **Logging Profiles** and create a new profile with the following values, leaving unspecified attributes at their default value:
   a. Profile Name: dns-dos-profile-logging
   b. DoS Protection: Enabled

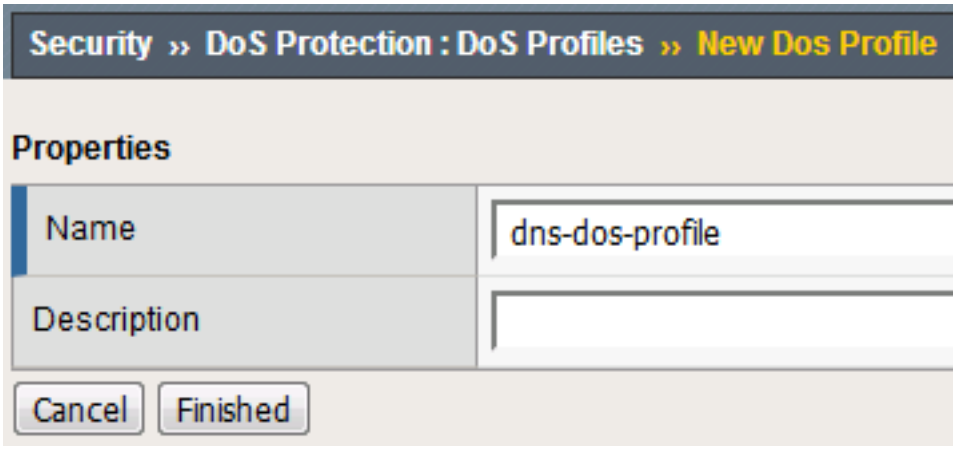
c. DNS DoS Protection Publisher: local-db-publisher



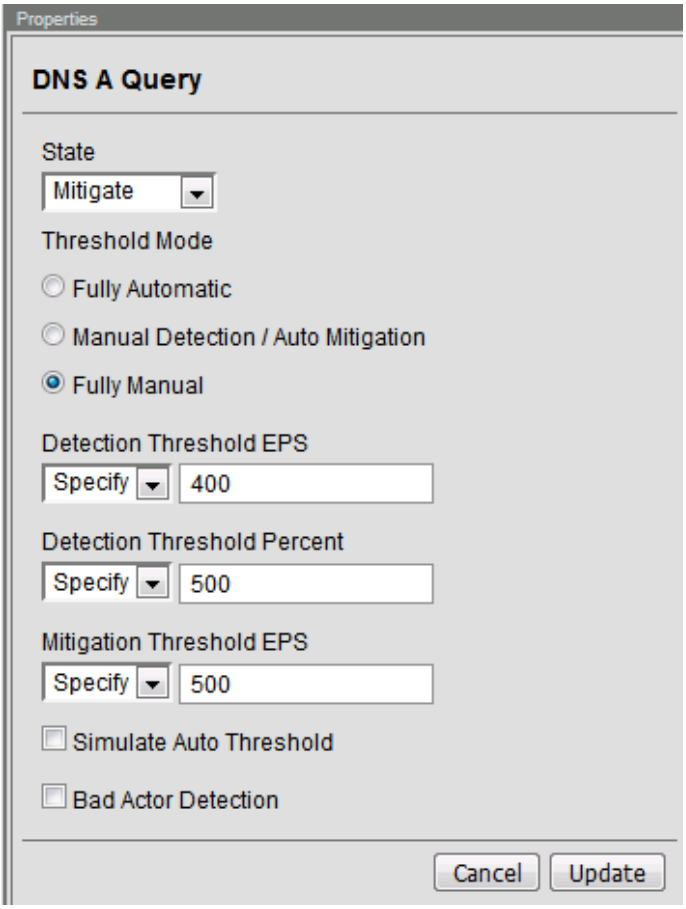
Configuring a DoS Profile

We'll now create a DoS profile with manually configured thresholds to limit the attack's effect on our server.

1. Navigate to **Security** > **DoS Protection** > **DoS Profiles** and create a new DoS profile with the name **dns-dos-profile**.



2. The UI will return to the DoS Profiles list. Click the **dns-dos-profile** name.
3. Click the **Protocol Security** tab and select **DNS Security** from the drop-down.
4. Click the **DNS A Query** vector from the Attack Type list.
5. Modify the **DNS A Query** vector configuration to match the following values, leaving unspecified attributes with their default value:
    a. State: Mitigate
    b. Threshold Mode: Fully Manual
    c. Detection Threshold EPS: (Set this at 80% of your safe QPS value)
    d. Mitigation Threshold EPS: (Set this to your safe QPS value)



6. Make sure that you click **Update** to save your changes.

Attaching a DoS Profile

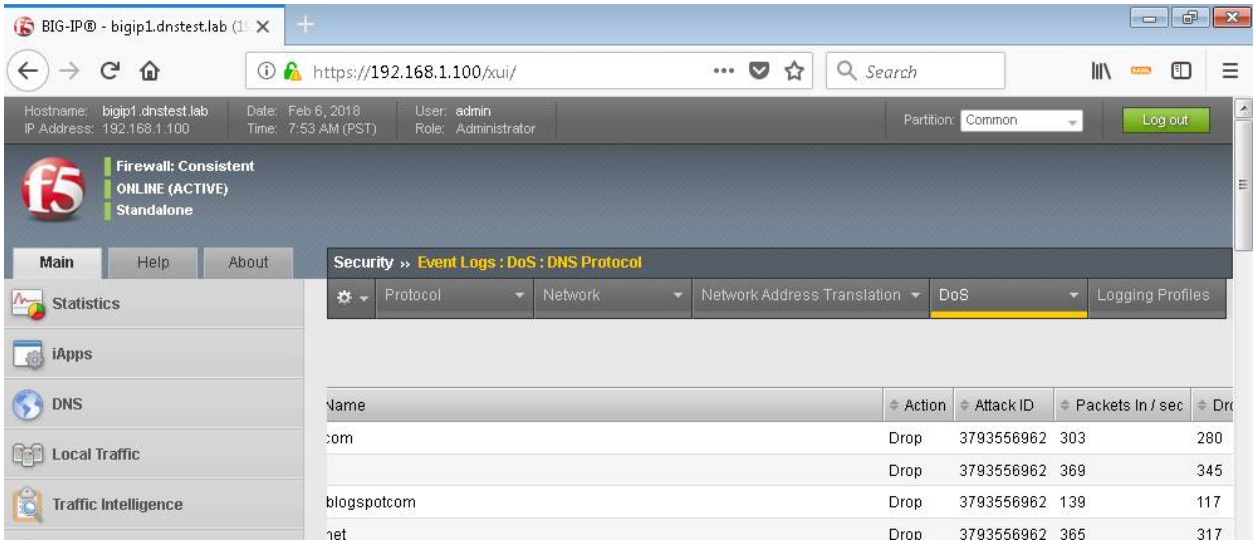
We'll attach the DoS profile to the virtual server that we configured to manage DNS traffic.

1. Navigate to **Local Traffic** > **Virtual Servers** > **Virtual Server List**.
2. Click on the **udp_dns_VS** name.
3. Click on the **Security** tab and select **Policies**.
4. In the **DoS Protection Profile** field, select **Enabled** and choose the **dns-dos-profile**.
5. In the **Log Profile**, select **Enabled** and move the **dns-dos-profile-logging** profile from **Available** to **Selected**.
6. Click **Update**.

Simulate a DNS DDoS Attack

1. Open the SSH session to the victim server and ensure the top utility is running.
2. Once again, attack your DNS server from the attack host using the following syntax:
   dnsperf -s 10.20.0.10 -d queryfile-example-current -c 20 -T 20 -l 30 -q 10000 -Q 10000
3. On the server SSH session running the top utility, notice the CPU utilization on your server remains in a range that ensures the DNS server is not overwhelmed.

4. After the attack, navigate to **Security** > **Event Logs** > **DoS** > **DNS Protocol**. Observe the logs to see the mitigation actions taken by the BIG-IP.



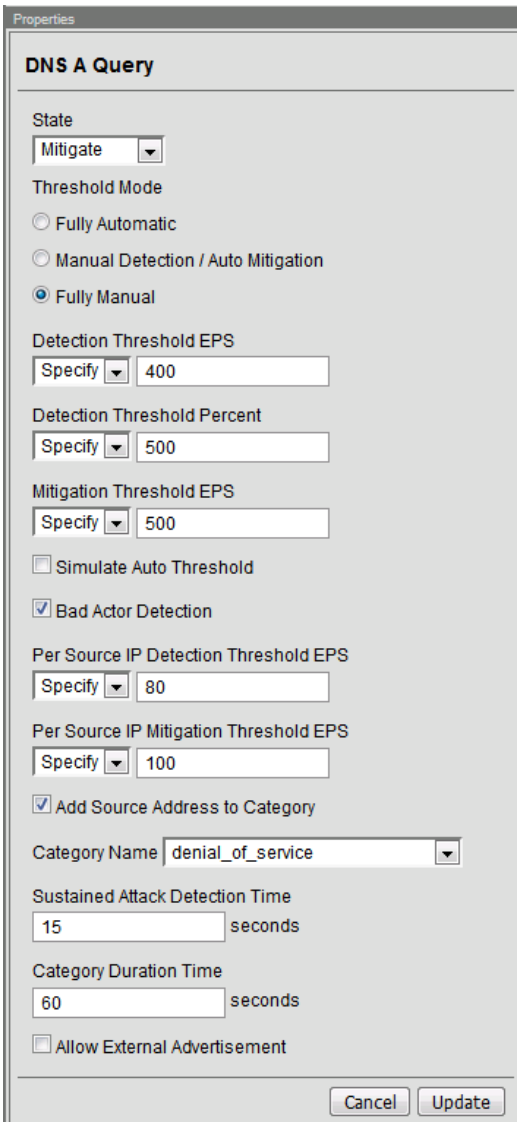
DNS DDoS Mitigations for Continued Service

At this point, you've successfully configured the BIG-IP to limit the amount of resource utilization on the BIG-IP. Unfortunately, even valid DNS requests can be caught in the mitigation we've configured. There are further steps that can be taken to mitigate the attack that will allow non-malicious DNS queries.

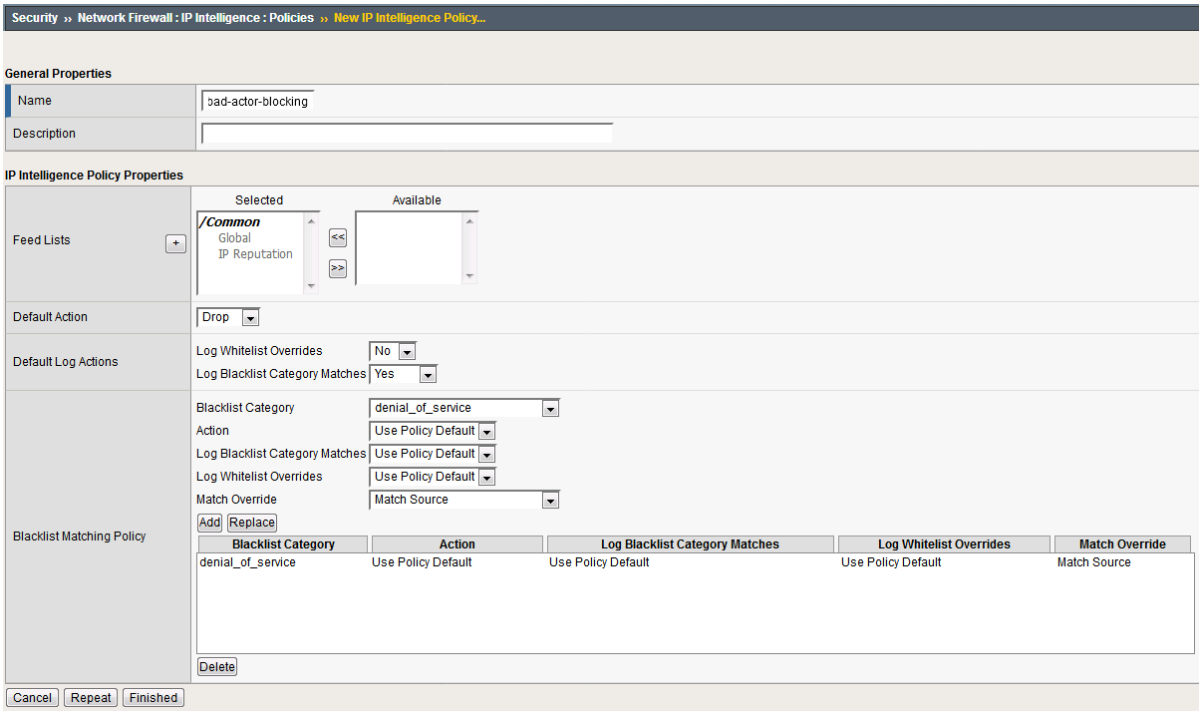
Bad Actor Detection

Bad actor detection and blacklisting allows us to completely block communications from malicious hosts at the BIG-IP, completely preventing those hosts from reaching the back-end servers. To demonstrate:

1. Navigate to **Security** > **DoS Protection** > **DoS Profiles**.
2. Click on the **dns-dos-profile** profile name.
3. Click on the **Protocol Security** tab then select **DNS Security**.
4. Click on the **DNS A Query** attack type name.
5. Modify the vector as follows:
    a. Bad Actor Detection: Checked
    b. Per Source IP Detection Threshold EPS: 80
    c. Per Source IP Mitigation Threshold EPS: 100
    d. Add Source Address to Category: Checked
    e. Category Name: denial_of_service
    f. Sustained Attack Detection Time: 15 seconds
    g. Category Duration Time: 60 seconds

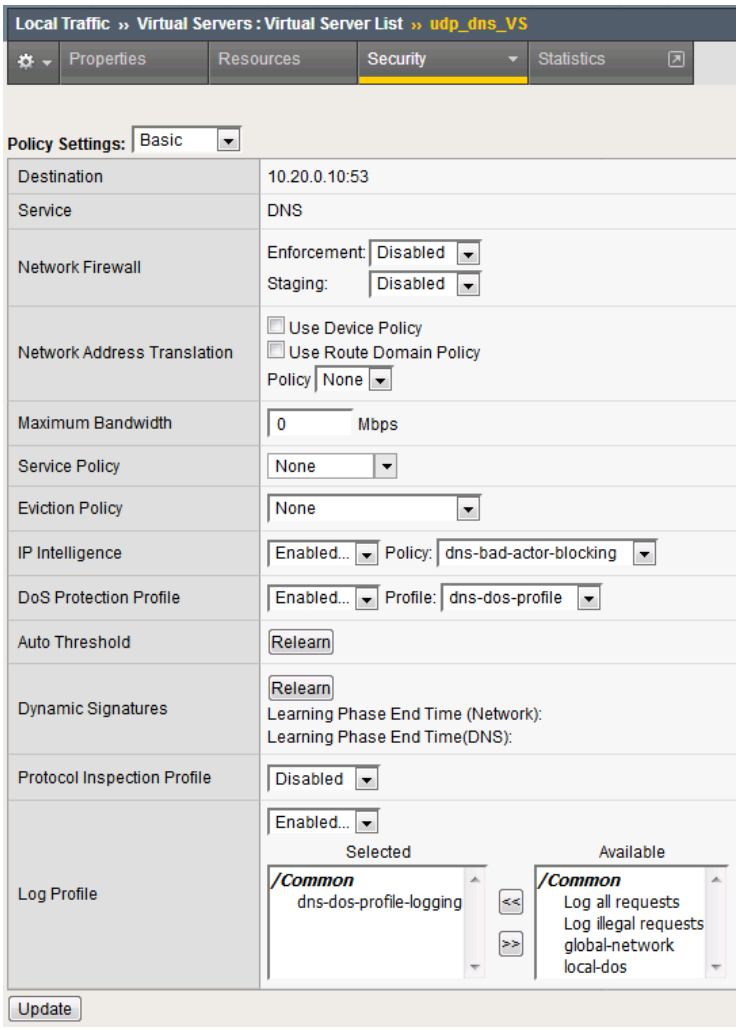


6. Make sure you click **Update** to save your changes.
7. Navigate to **Security** > **Network Firewall** > **IP Intelligence** > **Policies** and create a new IP Intelligence policy with the following values, leaving unspecified attributes at their default values:
    a. Name: dns-bad-actor-blocking
    b. Default Log Actions section:
        i. Log Blacklist Category Matches: Yes
    c. Blacklist Matching Policy
        i. Create a new blacklist matching policy:
            1. Blacklist Category: denial_of_service

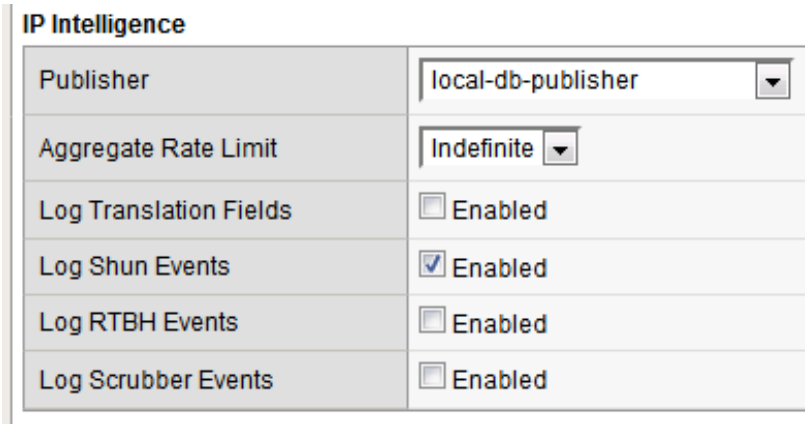


2. Click **Add** to add the policy.

8. Click **Finished**.
9. Navigate to **Local Traffic** > **Virtual Servers** > **Virtual Server List**.
10. Click on the **udp_dns_VS** virtual server name.
11. Click on the **Security** tab and select **Policies**.
12. Enable **IP Intelligence** and choose the **dns-bad-actor-blocking** policy.

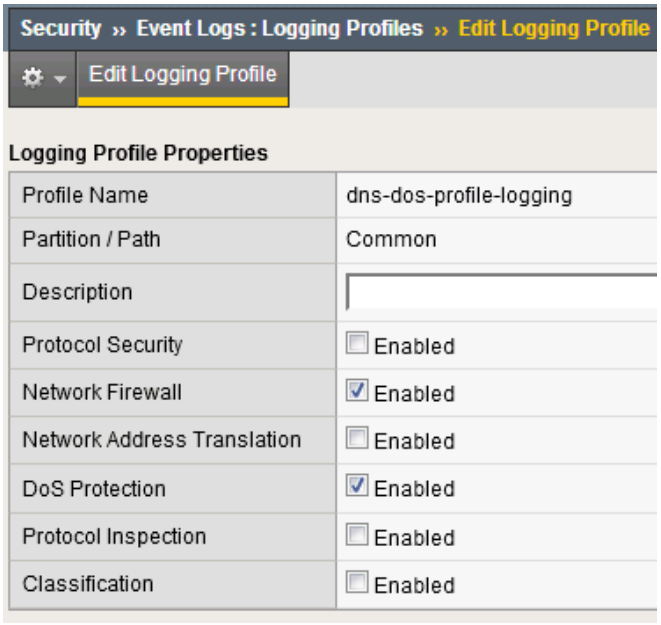


13. Make sure you click **Update** to save your changes.
14. Navigate to **Security** > **Event Logs** > **Logging Profiles**.
15. Click the **global-network** logging profile name.
16. Under the **Network Firewall** tab, set the IP Intelligence Publisher to **local-db-publisher** and check **Log Shun Events**.



17. Click **Update** to save your changes.

18. Click the **dns-dos-profile-logging** logging profile name.
19. Check **Enabled** next to **Network Firewall**.



20. Under the **Network Firewall** tab, change the **Network Firewall** and **IP Intelligence Publisher** to **local-db-publisher** and click **Update**.

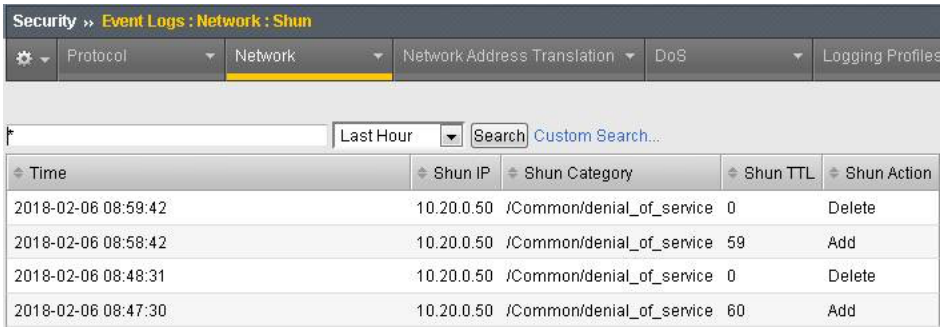
21. Bring into view the Victim Server SSH session running the top utility to monitor CPU utilization.
22. On the Attack Server host, launch the DNS attack once again using the following syntax:
dnsperf -s 10.20.0.10 -d queryfile-example-current -c 20 -T 20 -l 30 -q 10000 -Q 10000

23. You'll notice CPU utilization on the victim server begin to climb, but slowly drop. The attack host will show that queries are timing out as shown below. This is due to the BIG-IP blacklisting the bad actor.

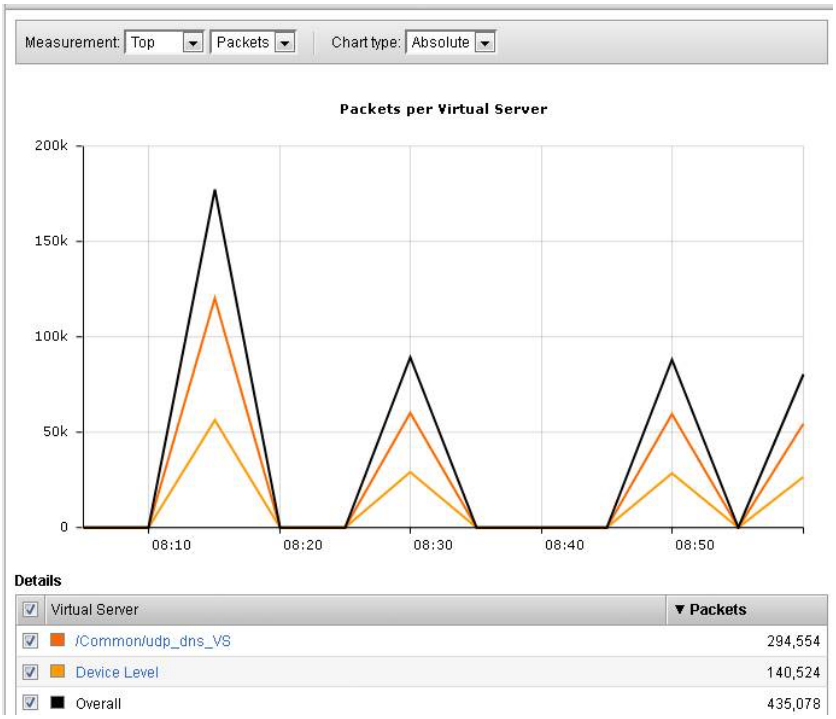
```
[Timeout] Query timed out: msg id 3466
[Timeout] Query timed out: msg id 3467
[Timeout] Query timed out: msg id 3468
[Timeout] Query timed out: msg id 3469
[Timeout] Query timed out: msg id 3470
[Timeout] Query timed out: msg id 3471
```

24. Navigate to **Security** > **Event Logs** > **Network** > **IP Intelligence**. Observe the bad actor blocking mitigation logs.

25. Navigate to **Security** > **Event Logs** > **Network** > **Shun**. This screen shows the bad actor being added to (and later deleted from) the shun category.

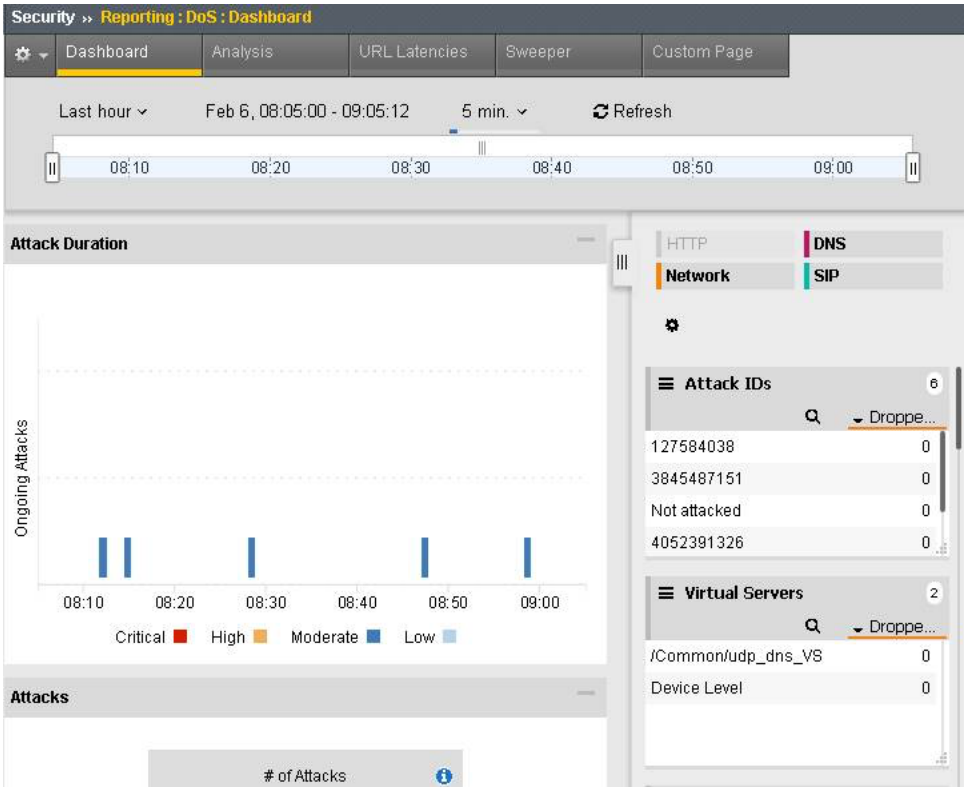


26. Navigate to **Security** > **Reporting** > **Protocol** > **DNS**. Change the **View By** drop-down to view various statistics around the DNS traffic and attacks.



27. Navigate to **Security** > **Reporting** > **Network** > **IP Intelligence**. The default view may be blank. Change the **View By** drop-down to view various statistics around the IP Intelligence handling of the attack traffic.

28. Navigate to **Security** > **Reporting** > **DoS** > **Dashboard** to view an overview of the DoS attacks and timeline. You can select filters in the filter pane to highlight specific attacks.



29. Finally, navigate to **Security** > **Reporting** > **DoS** > **Analysis**. View detailed statistics around each attack.

Remote Triggered Black Holing

The BIG-IP supports the advertisement of bad actor(s) to upstream devices via BGP to block malicious traffic closer to the source. This is accomplished by publishing a blacklist to an external resource. This is not demonstrated in this lab.

Silverline Mitigation

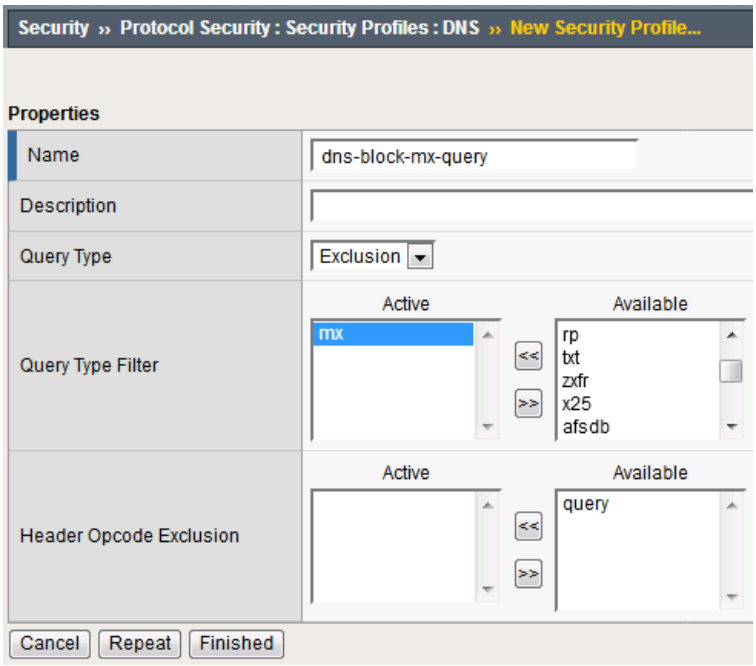
F5's cloud-based scrubbing service Silverline offers "always on" and "on demand" DDoS scrubbing that could assist in this scenario as well. This is not demonstrated in this lab.

3.1.3 Filtering specific DNS operations

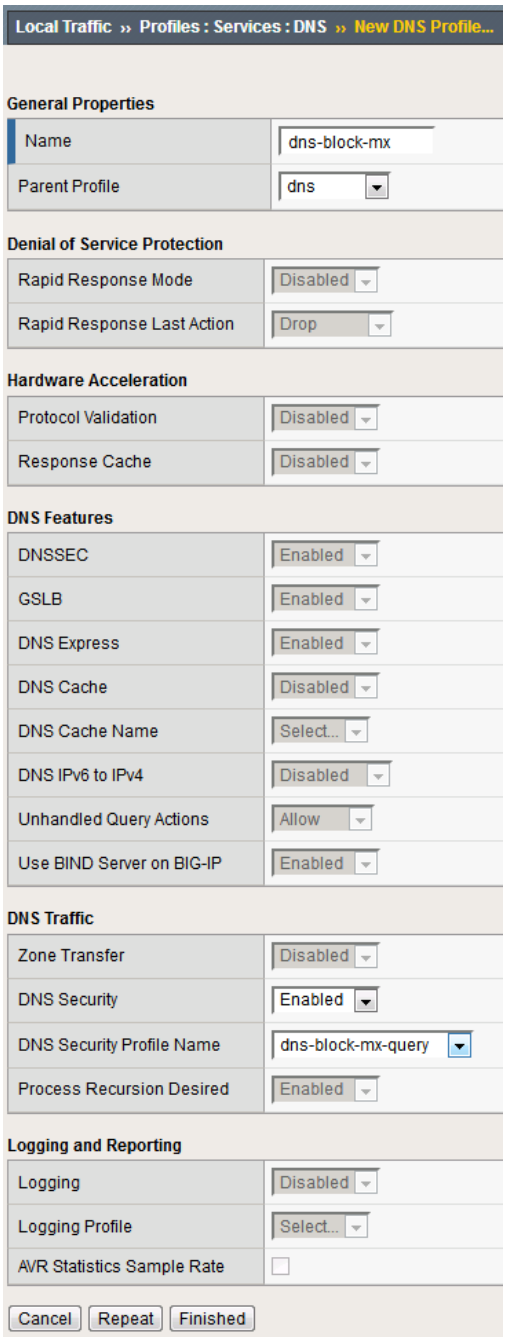
The BIG-IP offers the ability to filter DNS query types and header opcodes to act as a DNS firewall. To demonstrate, we will block MX queries from our DNS server.

1. Open the SSH session to the attack host.
2. Perform an MX record lookup by issuing the following command:
   dig @10.20.0.10 MX example.com
3. The server doesn't have a record for this domain. This server doesn't have MX records, so those requests should be filtered

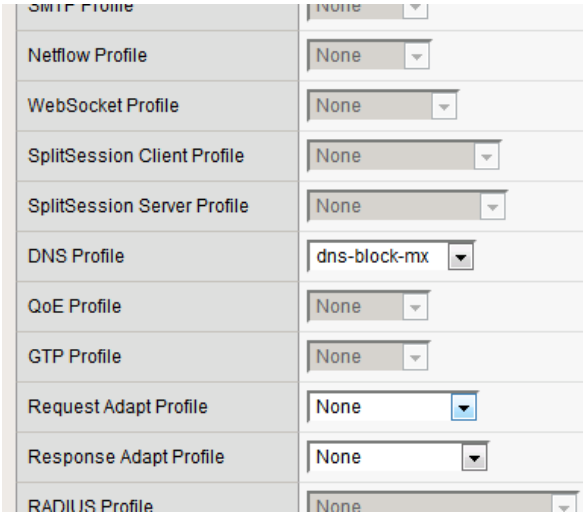
4. Navigate to **Security** > **Protocol Security** > **Security Profiles** > **DNS** and create a new DNS security profile with the following values, leaving unspecified attributes at their default value:
   a. Name: dns-block-mx-query
   b. Query Type Filter: move mx from Available to Active



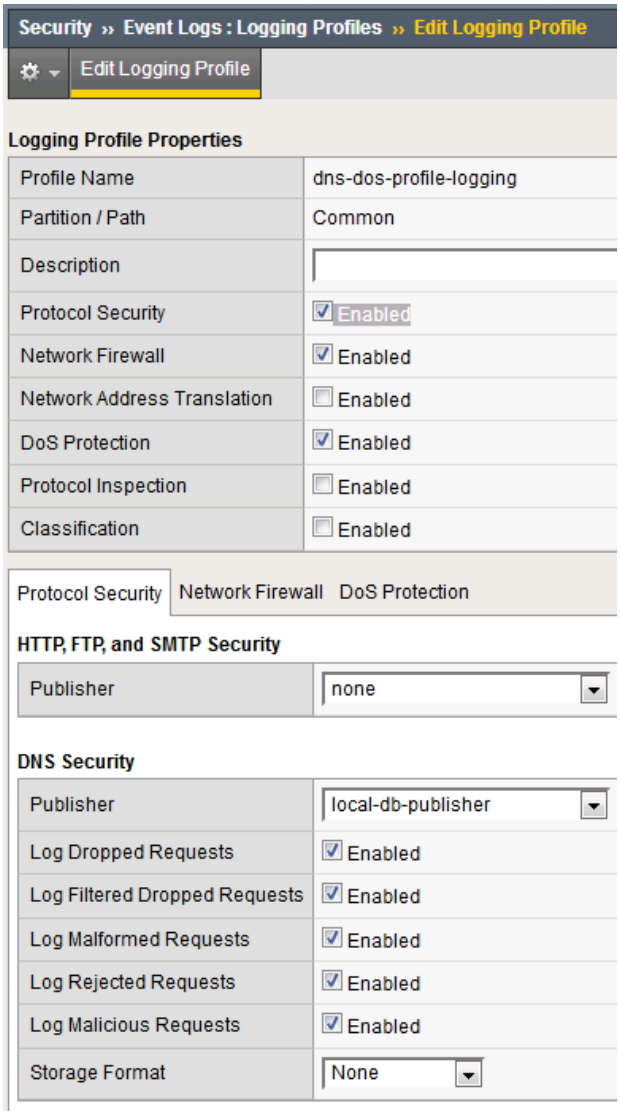
5. Navigate to **Local Traffic** > **Profiles** > **Services** > **DNS**. **NOTE:** if you are mousing over the services, DNS may not show up on the list. Select **Services** and then use the pulldown menu on services to select **DNS**.
6. Create a new DNS services profile with the following values, leaving unspecified values at their default values:
   a. Name: dns-block-mx
   b. DNS Traffic
      i. DNS Security: Enabled
      ii. DNS Security Profile Name: dns-block-mx-query



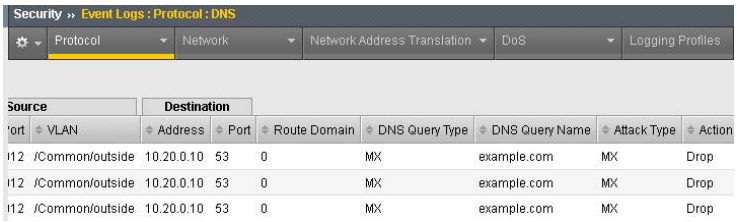
7. Navigate to **Local Traffic** > **Virtual Servers** > **Virtual Server List**.
8. Click on the **udp_dns_VS** virtual server name.
9. In the **Configuration** section, change the view to **Advanced**.
10. Set the **DNS Profile** to **dns-block-mx**.



11. Click **Update** to save your settings.
12. Navigate to **Security** > **Event Logs** > **Logging Profiles**.
13. Click on the **dns-dos-profile-logging** logging profile name.
14. Check **Enabled** next to **Protocol Security**.
15. In the **Protocol Security** tab, set the **DNS Security Publisher** to local-db-publisher and check all five of the request log types.



16. Make sure that you click **Update** to save your settings.
17. Return to the Attack Server SSH session and re-issue the MX query command:
    dig @10.20.0.10 MX example.com
18. The query hangs as the BIG-IP is blocking the MX lookup.
19. Navigate to **Security** > **Event Logs** > **Protocol** > **DNS**. Observer the MX query drops.



Attention: This concludes the DNS portion of the lab. On the victim server, stop the top utility by pressing **CTRL + C**.

3.2 Module 2 – Detecting and Preventing System DoS and DDoS Attacks

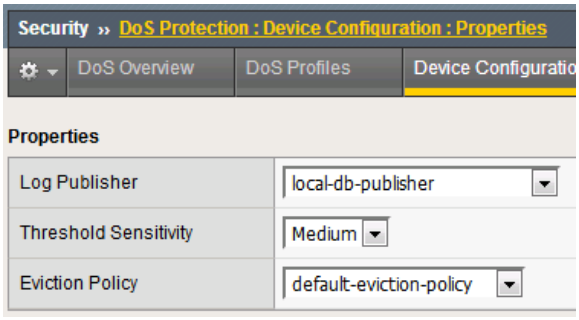
In this lab, you will launch attacks against the BIG-IP, configure mitigation and finally review the reports and logs.

3.2.1 Detecting and Preventing System DoS and DDoS Attacks

Configure Logging

Configuring a logging destination will allow you to verify the BIG-IPs detection and mitigation of attacks, in addition to the built-in reporting.

1. In the BIG-IP web UI, navigate to **Security** > **DoS Protection** > **Device Configuration** > **Properties**.
2. Under **Log Pubisher**, select **local-db-publisher**.
3. Click the **Commit Changes to System** button.

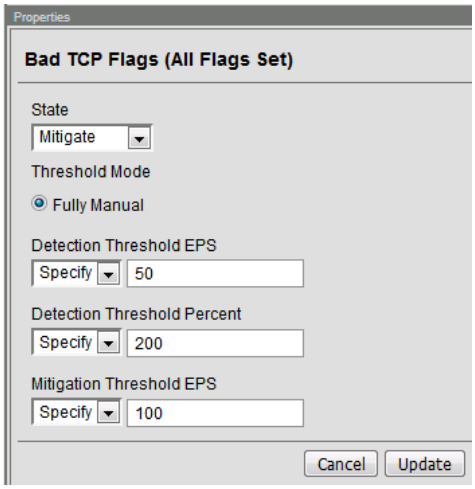


Simulating a Christmas Tree Packet Attack

In this example, we'll set the BIG-IP to detect and mitigate an attack where all flags on a TCP packet are set. This is commonly referred to as a Christmas tree packet and is intended to increase processing on in-path network devices and end hosts to the target.

We'll use the hping utility to send 25,000 packets to our server, with random source IPs to simulate a DDoS attack where multiple hosts are attacking our server. We'll set the SYN, ACK, FIN, RST, URG, PUSH, Xmas and Ymas TCP flags.

1. In the BIG-IP web UI, navigate to **Security** > **DoS Protection** > **Device Configuration** > **Network Security**.
2. Expand the **Bad-Header-TCP** category in the vectors list.
3. Click on the **Bad TCP Flags (All Flags Set)** vector name.
4. Configure the vector with the following parameters:
    a. State: Mitigate
    b. Threshold Mode: Fully Manual
    c. Detection Threshold EPS: Specify 50
    d. Detection Threshold Percent: Specify 200
    e. Mitigation Threshold EPS: Specify 100



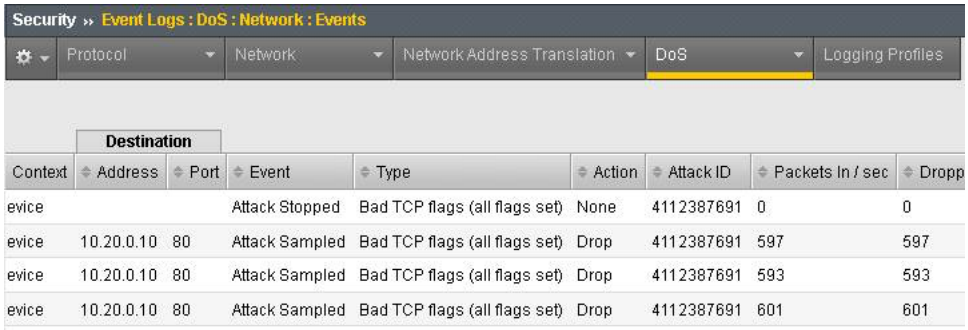
5. Click **Update** to save your changes.
6. Open the BIG-IP SSH session and scroll the ltm log in real time with the following command: tail -f /var/log/ltm
7. On the attack host, launch the attack by issuing the following command on the BASH prompt:
   sudo hping3 10.20.0.10 –flood –rand-source –destport 80 -c 25000 –syn –ack –fin –rst –push –urg –xmas –ymas
8. You'll see the BIG-IP ltm log show that the attack has been detected:

```
Feb  6 09:36:09 bigip1 err tmm[10663]: 01010252:3: A Enforced Device DOS attack
start was detected for vector Bad TCP flags (all flags set), Attack ID 411238769
1.
```

9. After approximately 60 seconds, press **CTRL+C** to stop the attack.

```
ubuntu@attackhost:~$ sudo hping3 10.20.0.10 --flood --rand-source --destport 80
-c 25000 --syn --ack --fin --rst --push --urg --xmas --ymas
HPING 10.20.0.10 (ens3 10.20.0.10): RSAFPUXY set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 10.20.0.10 hping statistic ---
361447 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
ubuntu@attackhost:~$
```

10. Return to the BIG-IP web UI. Navigate to **Security** > **Event Logs** > **DoS** > **Network** > **Events**. Observer the log entries showing the details surrounding the attack detection and mitigation.

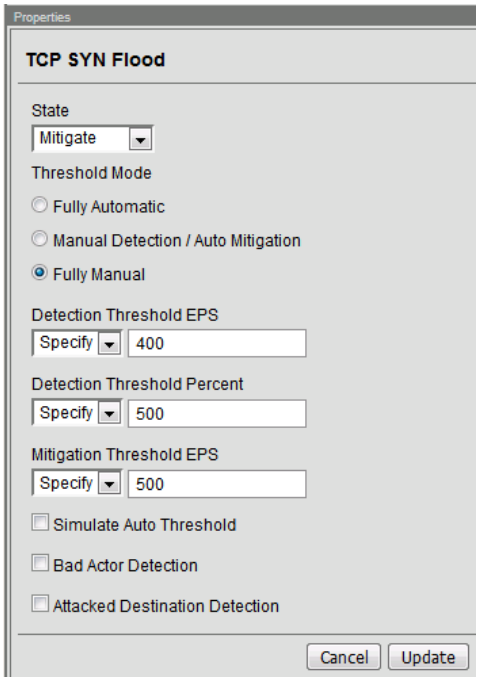


11. Navigate to **Security** > **Reporting** > **DoS** > **Analysis**. Single-click on the attack ID in the filter list to the right of the charts and observe the various statistics around the attack.

Simulating a TCP SYN DDoS Attack

In the last example, we crafted a packet that is easily identified as malicious, as its invalid. We'll now simulate an attack with traffic that could be normal, acceptable traffic. The TCP SYN flood attack will attempt to DDoS a host by sending valid TCP traffic to a host from multiple source hosts.

1. In the BIG-IP web UI, navigate to **Security** > **DoS Protection** > **Device Configuration** > **Network Security**.
2. Expand the **Flood** category in the vectors list.
3. Click on **TCP Syn Flood** vector name.
4. Configure the vector with the following parameters (use the lower values specified):
    a. State: Mitigate
    b. Threshold Mode: Fully Manual
    c. Detection Threshold EPS: 50
    d. Detection Threshold Percent: 200
    e. Mitigation Threshold EPS: 100



5. Click **Update** to save your changes.
6. Open the BIG-IP SSH session and scroll the ltm log in real time with the following command: tail -f /var/log/ltm
7. On the attack host, launch the attack by issuing the following command on the BASH prompt:
   sudo hping3 10.20.0.10 –flood –rand-source –destport 80 –syn -d 120 -w 64
8. After about 60 seconds, stop the flood attack by pressing **CTRL + C**.
9. Return to the BIG-IP web UI and navigate to **Security** > **Event Logs** > **DoS** > **Network** > **Events**. Observe the log entries showing the details surrounding the attack detection and mitigation.
10. Navigate to **Security** > **Reporting** > **DoS** > **Dashboard** to view an overview of the DoS attacks and timeline. You can select filters in the filter pane to highlight the specific attack.

11. Finally, navigate to **Security** > **Reporting** > **DoS** > **Analysis**. View detailed statistics around the attack.

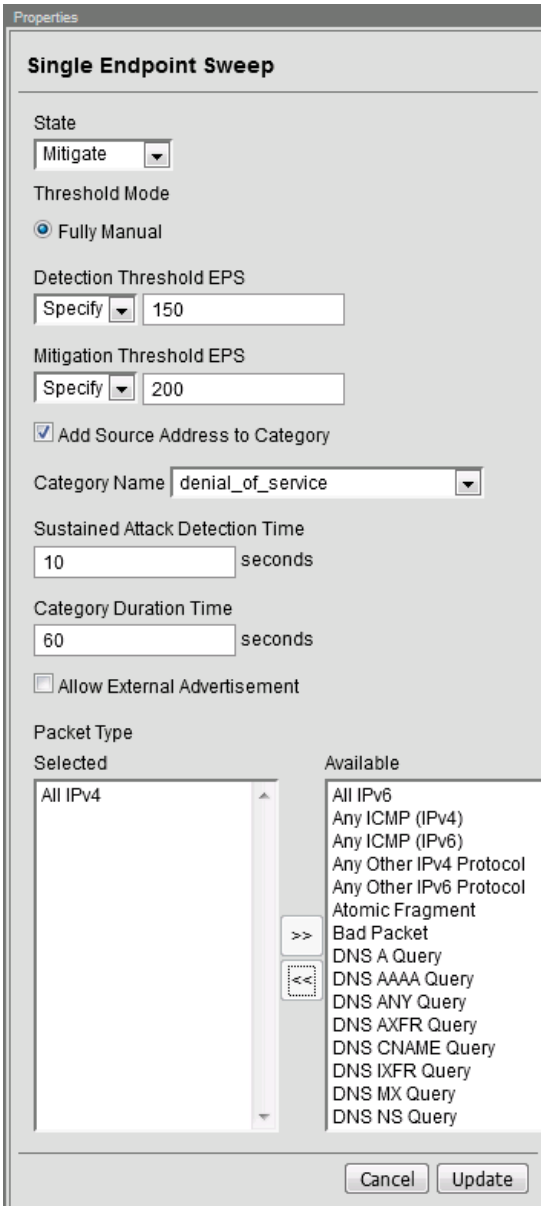
3.2.2 Preventing Global DoS Sweep and Flood Attacks

In the last section, the focus was on attacks originating from various hosts. In this section, we will focus on mitigating flood and sweep attacks from a single host.

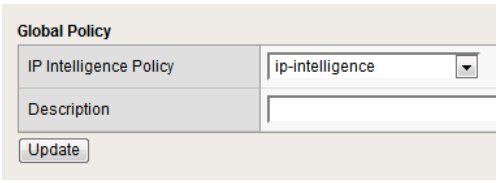
Single Endpoint Sweep

The single endpoint sweep is an attempt for an attacker to send traffic across a range of ports on the target server, typically to scan for open ports.

1. In the BIG-IP web UI, navigate to **Security** > **DoS Protection** > **Device Configuration** > **Network Security**.
2. Expand the **Single-Endpoint** category in the vectors list.
3. Click on **Single Endpoint Sweep** vector name.
4. Configure the vector with the following parameters:
    a. State: Mitigate
    b. Threshold Mode: Fully Manual
    c. Detection Threshold EPS: 150
    d. Mitigation Threshold EPS: 200
    e. Add Source Address to Category: Checked
    f. Category Name: denial_of_service
    g. Sustained Attack Detection Time: 10 seconds
    h. Category Duration Time: 60 seconds
    i. Packet Type: Move All IPv4 to Selected



5. Click **Update** to save your changes.
6. Navigate to **Security** > **Network Firewall** > **IP Intelligence** > **Policies**.
7. In the **Global Policy** section, change the **IP Intelligence Policy** to **ip-intelligence**.



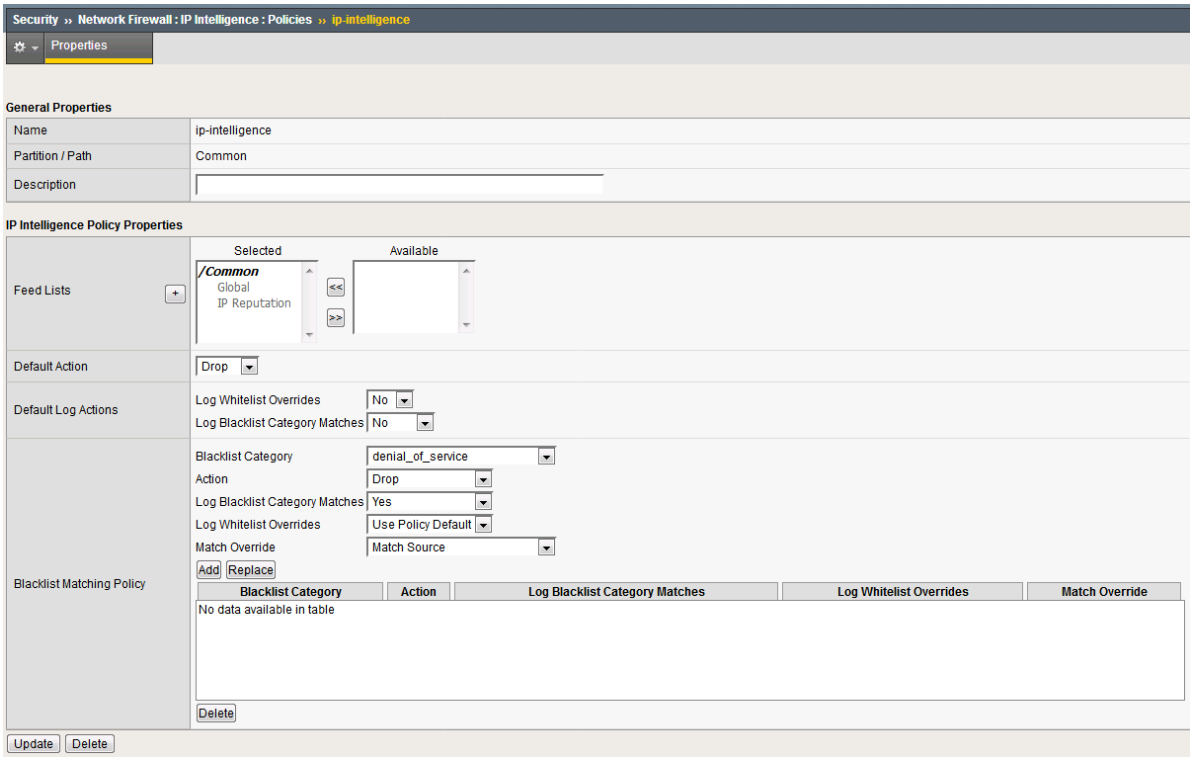
8. Click **Update**.
9. Click on the **ip-intelligence** policy in the policy list below.
10. Create a new Blacklist Matching Policy in the IP Intelligence Policy Properties section with the following attributes, leaving unspecified attributes with their default values:

a. Blacklist Category: denial-of-service

b. Action: drop

c. Log Blacklist Category Matches: Yes

11. Click **Add** to add the new Blacklist Matching Policy.



12. Click **Update** to save changes to the ip-intelligence policy.

13. Open the BIG-IP SSH session and scroll the ltm log in real time with the following command: tail -f /var/log/ltm

14. On the victim server, start a packet capture with an SSH filter by issuing sudo tcpdump -nn not port 22

15. On the attack host, launch the attack by issuing the following command on the BASH prompt:
sudo hping3 10.20.0.10 –flood –scan 1-65535 -d 128 -w 64 –syn

16. You will see the scan find a few open ports on the server, and the server will show the inbound sweep traffic. However, you will notice that the traffic to the server stops after a short time (10 seconds, the configured sustained attack detection time.) Leave the test running.

17. After approximately 60 seconds, sweep traffic will return to the host. This is because the IP Intelligence categorization of the attack host has expired. After 10 seconds of traffic, the bad actor is again blacklisted for another 60 seconds.

18. Stop the sweep attack on the attack host by pressing **CTRL + C**.

19. Return to the BIG-IP web UI and navigate to **Security** > **Event Logs** > **DoS** > **Network** > **Events**. Observe the log entries showing the details surrounding the attack detection and mitigation.

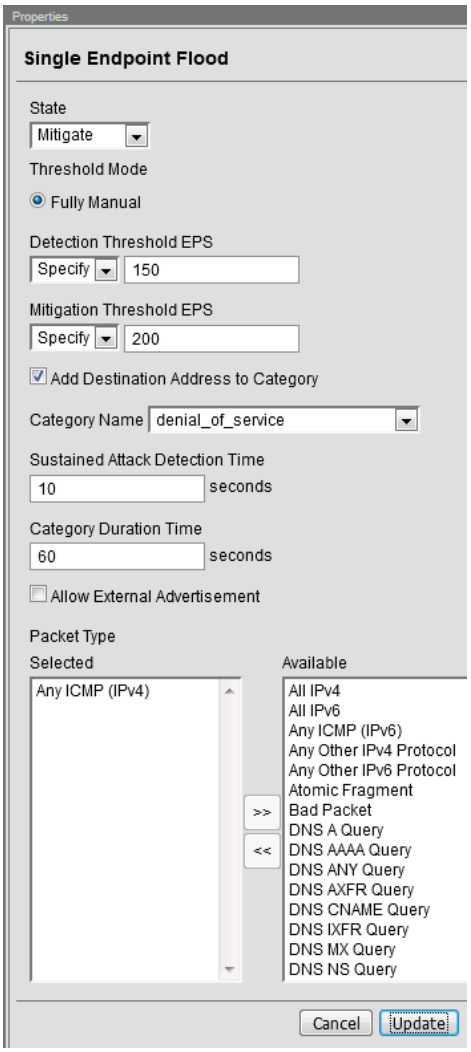
20. Navigate to **Security** > **Event Logs** > **Network** > **IP Intelligence**. Observe the log entries showing the mitigation of the sweep attack via the ip-intelligence policy.

21. Navigate to **Security** > **Event Logs** > **Network** > **Shun**. Observe the log entries showing the blacklist adds and deletes.
22. Navigate to **Security** > **Reporting** > **Network** > **IP Intelligence**. Observe the statistics showing the sweep attack and mitigation. Change the **View By** drop-down to view the varying statistics.
23. Navigate to **Security** > **Reporting** > **DoS** > **Dashboard** to view an overview of the DoS attacks and timeline. You can select filters in the filter pane to highlight the specific attack.
24. Finally, navigate to **Security** > **Reporting** > **DoS** > **Analysis**. View detailed statistics around the attack.

Single Endpoint Flood

The single endpoint flood attack is an attempt for an attacker to send a flood of traffic to a host in hopes of overwhelming a service to a point of failure. In this example, we'll flood the target server with ICMP packets.

1. In the BIG-IP web UI, navigate to **Security** > **DoS Protection** > **Device Configuration** > **Network Security**.
2. Expand the **Single-Endpoint** category in the vectors list.
3. Click on **Single Endpoint Flood** vector name.
4. Configure the vector with the following parameters:
    a. State: Mitigate
    b. Threshold Mode: Fully Manual
    c. Detection Threshold EPS: 150
    d. Mitigation Threshold EPS: 200
    e. Add Destination Address to Category: Checked
    f. Category Name: denial_of_service
    g. Sustained Attack Detection Time: 10 seconds
    h. Category Duration Time: 60 seconds
    i. Packet Type: Move Any ICMP (IPv4) to Selected



5. Click **Update** to save your changes.
6. Open the BIG-IP SSH session and scroll the ltm log in real time with the following command: tail -f /var/log/ltm
7. We'll run a packet capture on the victim server to gauge the incoming traffic. On the victim server, issue the following command: sudo tcpdump -nn not port 22
8. On the attack host, launch the attack by issuing the following command on the BASH prompt:
   sudo hping3 10.20.0.10 –faster -c 25000 –icmp
9. The attack host will begin flooding the victim server with ICMP packets. However, you will notice that the traffic to the server stops after a short time (10 seconds, the configured sustained attack detection time.)
10. After approximately 60 seconds, run the attack again. ICMP traffic will return to the host. This is because the IP Intelligence categorization of the attack host has expired.
11. Return to the BIG-IP web UI.
12. Navigate to **Security** > **Event Logs** > **DoS** > **Network** > **Events**. Observe the log entries showing the details surrounding the attack detection and mitigation.
13. Navigate to **Security** > **Event Logs** > **Network** > **IP Intelligence**. Observe the log entries showing

the mitigation of the sweep attack via the ip-intelligence policy.

14. Navigate to **Security** > **Reporting** > **Network** > **IP Intelligence**. Observe the statistics showing the sweep attack and mitigation.
15. Navigate to **Security** > **Reporting** > **DoS** > **Dashboard** to view an overview of the DoS attacks and timeline. You can select filters in the filter pane to highlight the specific attack.
16. Finally, navigate to **Security** > **Reporting** > **DoS** > **Analysis**. View detailed statistics around the attack.

3.2.3 Conclusion

Congratulations on finishing the lab!

This lab did not cover auto thresholds for protections, nor did we test dynamic signatures. Testing auto thresholds requires a more real-world environment. For suggested testing guidelines for auto thresholds and dynamic signatures, engage your F5 account team.

This concludes the DoS/DDoS portion of the lab. You may now close all sessions, log out of the jump host and log out of the training portal.

Thank you for your time.

3.3 Appendix

3.3.1 DNS Security vectors

The system tracks and rate limits all UDP DNS packets (excluding those whitelisted). TCP DNS packets are also tracked but only for the DNS requests that reach a virtual server that has a DNS profile associated with it.

NOTE: This information applies to 13.1.0.1.

For vectors where VLAN is <tunable>, you can tune this value in tmsh: modify sys db dos.dnsvlan value, where value is 0-4094.

DoS category	Attack name	Dos vector name	Information	Hardware accelerated
DNS	DNS A Query	dns-a-query	DNS Query, DNS Qtype is A_QRY, VLAN is <tunable> in tmsh usingdos.dnsvlan.	Yes
DNS	DNS AAAA Query	dns-aaaa-query	DNS Query, DNS Qtype is AAAA, VLAN is <tunable> in tmsh usingdos.dnsvlan.	Yes
DNS	DNS Any Query	dns-any-query	DNS Query, DNS Qtype is ANY_QRY, VLAN is <tunable> in tmsh usingdos.dnsvlan.	Yes
DNS	DNS AXFR Query	dns-axfr-query	DNS Query, DNS Qtype is AXFR, VLAN is <tunable> in tmsh usingdos.dnsvlan.	Yes
DNS	DNS CNAME Query	dns-cname-query	DNS Query, DNS Qtype is CNAME, VLAN is <tunable> in tmsh usingdos.dnsvlan.	Yes
DNS	DNS IXFR Query	dns-ixfr-query	DNS Query, DNS Qtype is IXFR, VLAN is <tunable> in tmsh usingdos.dnsvlan.	Yes
DNS	DNS Malformed	dns-malformed	Malformed DNS packet	Yes
DNS	DNS MX Query	dns-mx-query	DNS Query, DNS Qtype is MX, VLAN is <tunable> in tmsh usingdos.dnsvlan.	Yes
DNS	DNS NS Query	dns-ns-query	DNS Query, DNS Qtype is NS, VLAN is <tunable> in tmsh usingdos.dnsvlan.	Yes
DNS	DNS OTHER Query	dns-other-query	DNS Query, DNS Qtype is OTHER, VLAN is <tunable> in tmsh usingdos.dnsvlan.	Yes
DNS	DNS PTR Query	dns-ptr-query	DNS Query, DNS Qtype is PTR, VLAN is <tunable> in tmsh usingdos.dnsvlan.	Yes
DNS	DNS Question Items != 1	dns-qdcount-limit	DNS Query, DNS Qtype is ANY_QRY, the DNS query has more than one question.	Yes
DNS	DNS Response Flood	dns-response-flood	UDP DNS Port=53, packet and DNS header flags bit 15 is 1 (response), VLAN is <tunable> in tmsh using dos.dnsvlan.	Yes
DNS	DNS SOA Query	dns-soa-query	DNS Query, DNS Qtype is SOA_QRY, VLAN is <tunable> in tmsh usingdos.dnsvlan.	Yes
DNS	DNS SRV Query	dns-srv-query	DNS Query, DNS Qtype is SRV, VLAN is <tunable> in tmsh usingdos.dnsvlan.	Yes
DNS	DNS TXT Query	dns-txt-query	DNS Query, DNS Qtype is TXT, VLAN is <tunable> in tmsh usingdos.dnsvlan.	Yes

3.3.2 Network Security Vectors

DoS category	Attack name	Dos vector name	Information	Hardware accelerated
Flood	Ethernet Broadcast Packet	ether-brdcst-pkt	Ethernet broadcast packet flood	Yes

Continued on next page

Table 1 – continued from previous page

DoS category	Attack name	Dos vector name	Information	Hardware accelerated
Flood	Ethernet Multicast Packet	ether-multicst-pkt	Ethernet destination is not broadcast, but is multicast	Yes
Flood	ARP Flood	arp-flood	ARP packet flood	Yes
Flood	IP Fragment Flood	ip-frag-flood	Fragmented packet flood with IPv4	Yes
Flood	IGMP Flood	igmp-flood	Flood with IGMP packets (IPv4 packets with IP protocol number 2)	Yes
Flood	Routing Header Type 0	routing-header-type-0	Routing header type zero is present in flood packets	Yes
Flood	IPv6 Fragment Flood	ipv6-frag-flood	Fragmented packet flood with IPv6	No
Flood	IGMP Fragment Flood	igmp-frag-flood	Fragmented packet flood with IGMP protocol	Yes
Flood	TCP SYN Flood	tcp-syn-flood	TCP SYN flood	Yes
Flood	TCP SYN ACK Flood	tcp-synack-flood	TCP SYN/ACK flood	Yes
Flood	TCP RST Flood	tcp-rst-flood	TCP RST flood	Yes
Flood	TCP Window Size	tcp-window-size	The TCP window size in packets is above the maximum. To tune this value, in tmsh: modify sys db dos.tcplowwindowsize value, where value is <=128.	Yes
Flood	ICMPv4 Flood	icmpv4-flood	Flood with ICMP v4 packets	Yes
Flood	ICMPv6 Flood	icmpv6-flood	Flood with ICMP v6 packets	Yes
Flood	UDP Flood	udp-flood	UDP flood attack	Yes

Continued on next page

Table 1 – continued from previous page

DoS category	Attack name	Dos vector name	Information	Hardware accelerated
Flood	TCP SYN Oversize	tcp-syn-oversize	Detects TCP data SYN packets larger than the maximum specified by the dos.maxsynsize parameter. To tune this value, in tmsh: modify sys db dos.maxsynsize value. The default size is 64 and the maximum allowable value is 9216.	Yes
Flood	TCP Push Flood	tcp-push-flood	TCP push packet flood	Yes
Flood	TCP BADACK Flood	tcp-ack-flood	TCP ACK packet flood	No
Bad Header - L2	Ethernet MAC Source Address == Destination Address	ether-mac-sa-eq-da	Ethernet MAC source address equals the destination address	Yes
Bad Header - IPv4	Bad IP Version	bad-ver	The IPv4 address version in the IP header is not 4	Yes
Bad Header - IPv4	Header Length Too Short	hdr-len-too-short	IPv4 header length is less than 20 bytes	Yes
Bad Header - IPv4	Header Length > L2 Length	hdr-len-gt-l2-len	No room in layer 2 packet for IP header (including options) for IPv4 address	Yes
Bad Header - IPv4	L2 Length >> IP Length	l2-len-ggt-ip-len	Layer 2 packet length is much greater than the payload length in an IPv4 address header and the layer 2 length is greater than the minimum packet size	Yes
Bad Header - IPv4	No L4	no-l4	No layer 4 payload for IPv4 address	Yes
Bad Header - IPv4	Bad IP TTL Value	bad-ttl-val	Time-to-live equals zero for an IPv4 address	Yes

Continued on next page

Table 1 – continued from previous page

DoS category	Attack name	Dos vector name	Information	Hardware accelerated
Bad Header - IPv4	TTL <= <tunable>	ttl-leq-one	An IP packet with a destination that is not multicast and that has a TTL greater than 0 and less than or equal to a tunable value, which is 1 by default. To tune this value, in tmsh: modify sys db dos.iplowttli value, where value is 1-4.	Yes
Bad Header - IPv4	IP Error Checksum	ip-err-chksum	The header checksum is not correct	Yes
Bad Header - IPv4	IP Option Frames	ip-opt-frames	IPv4 address packet with option.db variable tm.acceptipsourceroute must be enabled to receive IP options.	Yes
Bad Header - IPv4	Bad Source	ip-bad-src	The IPv4 source IP = 255.255.255.255 or 0xe0000000U	Yes
Bad Header - IPv4	IP Option Illegal Length	bad-ip-opt	Option present with illegal length	No
Bad Header - IPv4	Unknown Option Type	unk-ipopt-type	Unknown IP option type	No
Bad Header - IGMP	Bad IGMP Frame	bad-igmp-frame	IPv4 IGMP packets should have a header >= 8 bytes. Bits 7:0 should be either 0x11, 0x12, 0x16, 0x22 or 0x17, or else the header is bad. Bits 15:8 should be non-zero only if bits 7:0 are 0x11, or else the header is bad.	Yes
Fragmentation	IP Fragment Too Small	ip-short-frag	IPv4 short fragment error	Yes
Fragmentation	IPv6 Fragment Too Small	ipv6-short-frag	IPv6 short fragment error	Yes

Continued on next page

Table 1 – continued from previous page

DoS category	Attack name	Dos vector name	Information	Hardware accelerated
Fragmentation	IPV6 Atomic Fragment	ipv6-atomic-frag	IPv6 Frag header present with M=0 and FragOffset =0	Yes
Fragmentation	ICMP Fragment	icmp-frag	ICMP fragment flood	Yes
Fragmentation	IP Fragment Error	ip-other-frag	Other IPv4 fragment error	Yes
Fragmentation	IPV6 Fragment Error	ipv6-other-frag	Other IPv6 fragment error	Yes
Fragmentation	IP Fragment Overlap	ip-overlap-frag	IPv4 overlapping fragment error	No
Fragmentation	IPv6 Fragment Overlap	ipv6-overlap-frag	IPv6 overlapping fragment error	No
Bad Header - IPv6	Bad IPV6 Version	bad-ipv6-ver	The IPv6 address version in the IP header is not 6	Yes
Bad Header - IPv6	IPV6 Length > L2 Length	ipv6-len-gt-l2-len	IPv6 address length is greater than the layer 2 length	Yes
Bad Header - IPv6	Payload Length < L2 Length	payload-len-ls-l2-len	Specified IPv6 payload length is less than the L2 packet length	Yes
Bad Header - IPv6	Too Many Extension Headers	too-many-ext-hdrs	For an IPv6 address, there are more than <tunable> extended headers (the default is 4). To tune this value, in tmsh: modify sys db dos.maxipv6exthdrs value, where value is 0-15.	Yes
Bad Header - IPv6	IPv6 duplicate extension headers	dup-ext-hdr	An extension header should occur only once in an IPv6 packet, except for the Destination Options extension header	Yes

Continued on next page

Table 1 – continued from previous page

DoS category	Attack name	Dos vector name	Information	Hardware accelerated
Bad Header - IPv6	IPv6 extension header too large	ext-hdr-too-large	An extension header is too large. To tune this value, in tmsh: modify sys db dos.maxipv6extsize value, where value is 0-1024.	Yes
Bad Header - IPv6	No L4 (Extended Headers Go To Or Past End of Frame)	l4-ext-hdrs-go-end	Extended headers go to the end or past the end of the L4 frame	Yes
Bad Header - IPv6	Bad IPV6 Hop Count	bad-ipv6-hop-cnt	Both the terminated (cnt=0) and forwarding packet (cnt=1) counts are bad	Yes
Bad Header - IPv6	IPv6 hop count <= <tunable>	hop-cnt-leq-one	The IPv6 extended header hop count is less than or equal to <tunable>. To tune this value, in tmsh: modify sys db dos.ipv6lowhopcnt value, where value is 1-4.	Yes
Bad Header - IPv6	IPv6 Extended Header Frames	ipv6-ext-hdr-frames	IPv6 address contains extended header frames	Yes
Bad Header - IPv6	IPv6 extended headers wrong order	bad-ext-hdr-order	Extension headers in the IPv6 header are in the wrong order	Yes
Bad Header - IPv6	Bad IPv6 Addr	ipv6-bad-src	IPv6 source IP = 0xff00::	Yes
Bad Header - IPv6	IPv4 Mapped IPv6	ipv4-mapped-ipv6	IPv4 address is in the lowest 32 bits of an IPv6 address.	Yes
Bad Header - TCP	TCP Header Length Too Short (Length < 5)	tcp-hdr-len-too-short	The Data Offset value in the TCP header is less than five 32-bit words	Yes
Bad Header - TCP	TCP Header Length > L2 Length	tcp-hdr-len-gt-l2-len		Yes

Continued on next page

Table 1 – continued from previous page

DoS category	Attack name	Dos vector name	Information	Hardware accelerated
Bad Header - TCP	Unknown TCP Option Type	unk-tcp-opt-type	Unknown TCP option type	Yes
Bad Header - TCP	Option Present With Illegal Length	opt-present-with-illegal-len	Option present with illegal length	Yes
Bad Header - TCP	TCP Option Overruns TCP Header	tcp-opt-overruns-tcp-hdr	The TCP option bits overrun the TCP header	Yes
Bad Header - TCP	Bad TCP Checksum	bad-tcp-chksum	The TCP checksum does not match	Yes
Bad Header - TCP	Bad TCP Flags (All Flags Set)	bad-tcp-flags-all-set	Bad TCP flags (all flags set)	Yes
Bad Header - TCP	Bad TCP Flags (All Cleared)	bad-tcp-flags-all-clr	Bad TCP flags (all cleared and SEQ#=0)	Yes
Bad Header - TCP	SYN && FIN Set	syn-and-fin-set	Bad TCP flags (SYN and FIN set)	Yes
Bad Header - TCP	FIN Only Set	fin-only-set	Bad TCP flags (only FIN is set)	Yes
Bad Header - TCP	TCP Flags - Bad URG	tcp-bad-urg	Packet contains a bad URG flag, this is likely malicious	Yes
Bad Header - ICMP	Bad ICMP Checksum	bad-icmp-chksum	An ICMP frame checksum is bad. Reuse the TCP or UDP checksum bits in the packet	Yes

Continued on next page

Table 1 – continued from previous page

DoS category	Attack name	Dos vector name	Information	Hardware accelerated
Bad Header - ICMP	Bad ICMP Frame	bad-icmp-frame	The ICMP frame is either the wrong size, or not of one of the valid IPv4 or IPv6 types. Valid IPv4 types: • 0 Echo Reply • 3 Destination Unreachable • 4 Source Quench • 5 Redirect • 8 Echo • 11 Time Exceeded • 12 Parameter Problem • 13 Timestamp • 14 Timestamp Reply • 15 Information Request • 16 Information Reply • 17 Address Mask Request • 18 Address Mask Reply Valid IPv6 types: • 1 Destination Unreachable • 2 Packet Too Big • 3 Time Exceeded • 4 Parameter Problem • 128 Echo Request • 129 Echo Reply	Yes

Table 1 – continued from previous page

DoS category	Attack name	Dos vector name	Information	Hardware accelerated
Bad Header - ICMP	ICMP Frame Too Large	icmp-frame-too-large	The ICMP frame exceeds the declared IP data length or the maximum datagram length. To tune this value, in tmsh: modify sys db dos.maxicmpframesize value, where value is <=65515.	Yes
Bad Header - UDP	Bad UDP Header (UDP Length > IP Length or L2 Length)	bad-udp-hdr	UDP length is greater than IP length or layer 2 length	Yes
Bad Header - UDP	Bad UDP Checksum	bad-udp-chksum	The UDP checksum is not correct	Yes
Other	Host Unreachable	host-unreachable	Host unreachable error	Yes
Other	TIDCMP	tidcmp	ICMP source quench attack	Yes
Other	LAND Attack	land-attack	Source IP equals destination IP address	Yes
Other	IP Unknown protocol	ip-unk-prot	Unknown IP protocol	No
Other	TCP Half Open	tcp-half-open	The number of new or untrusted TCP connections that can be established. Overrides the Global SYN Check threshold in Configuration > Local Traffic > General.	No
Other	IP uncommon proto	ip-uncommon-proto	Sets thresholds for and tracks packets containing IP protocols considered to be uncommon. By default, all IP protocols other than TCP, UDP, ICMP, IPV6-ICMP, and SCTP are on the IP uncommon protocol list.	Yes

Continued on next page

Table 1 – continued from previous page

DoS category	Attack name	Dos vector name	Information	Hardware accelerated
Bad Header - DNS	DNS Oversize	dns-oversize	Detects oversized DNS headers. To tune this value, in tmsh: modify sys db dos.maxdnssize value, where value is 256-8192.	Yes
Single Endpoint	Single Endpoint Sweep	sweep	Sweep on a single endpoint. You can configure packet types to check for, and packets per second for both detection and rate limiting.	No
Single Endpoint	Single Endpoint Flood	flood	Flood to a single endpoint. You can configure packet types to check for, and packets per second for both detection and rate limiting.	No
Bad Header-SCTP	Bad SCTP Checksum	bad-sctp-checksum	Bad SCTP packet checksum	No

4 Flowmon Integrated Out-of-path DDoS Solution

4.1 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.

4.1.1 Lab Topology

The following components have been included in your lab environment:

- 1 x F5 BIG-IP AFM VE (v13.1.0.6)
- 2 x vyOS routers (v1.1.8)
- 1 x Flowmon Collector (v9.01.04)/DDoS Defender (v4.01.00)
- 1 x Webserver (Ubuntu 16.04)
- 1 x Jumphost (Windows 7)
- 1 x Attacker (Ubuntu 16.04)

Lab Components

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

Component	VLAN/IP Address(es)	Connection Type, Credentials
Jumphost	• **Management:** 10.1.1.199 • **Users:** 10.1.10.30 • **Internal:** 10.1.20.30 • **Servers:** 10.1.30.30	RDP `external_user/P@ssw0rd!`
BIG-IP AFM	• **Management:** 10.1.1.7 • **Internal:** 10.1.20.245	TMUI `admin/admin`
Flowmon Collector/DDoS Defender	• **Management:** 10.1.1.9 • **Internal:** 10.1.20.10	TMUI `admin/admin`
Router 1	• **Management:** 10.1.1.10 • **Users:** 10.1.10.243 • **Internal:** 10.1.20.243	ssh `vyos/vyos`
Router 2	• **Management:** 10.1.1.11 • **Users:** 10.1.10.244 • **Internal:** 10.1.20.244	ssh `vyos/vyos`
Attacker	• **Management:** 10.1.1.4 • **Users:** 10.1.10.100	ssh `f5admin/f5admin`
Webserver	• **Management:** 10.1.1.6 • **Servers:** 10.1.30.252	ssh `f5admin/f5admin`

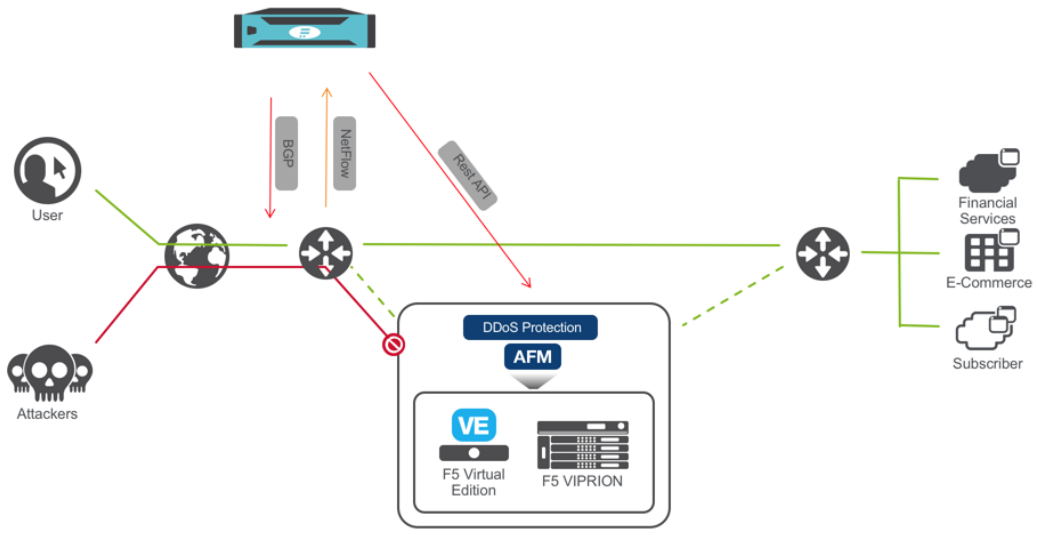
4.2 Module – Deployment use case and Lab diagram

In this module you will learn about common use-case for AFM/DHD + Flowmon out-of-path DDoS protection solution and explore Lab diagram.

4.2.1 Deployment use case

A Joint F5 + Flowmon solution is deployed “out-of-path” and provides an out-of-band DDoS mitigation of L3-4 volumetric DDoS attacks. It’s a simple and convenient solution that leverages the existing IT infrastructure to provide traffic flow information.

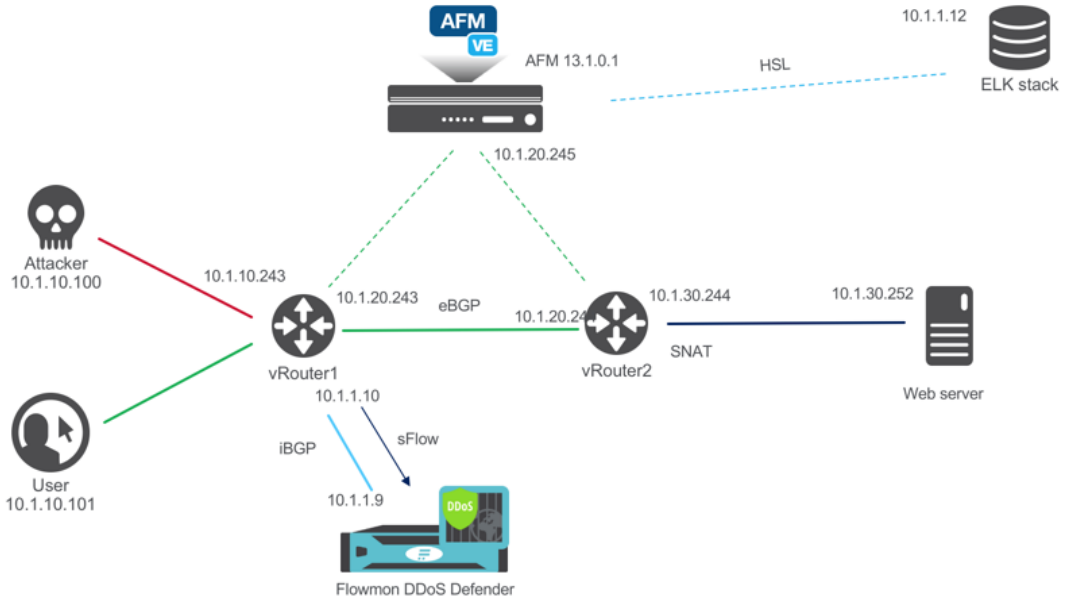
Flowmon Collector appliance receives NetFlow/sFlow/IPFIX from edge routers while Flowmon DDoS Defender uses i/eBGP/Flowspec to route the traffic to F5 DHD/AFM appliance. F5 DHD/AFM DDoS profile, VS and other parameters provisioned dynamically through iControl REST.



Pic.1 Solution Diagram

4.2.2 Lab blueprint setup

Lab blueprint is deployed in Oracle Ravello cloud with access from F5 UDF portal. All Flowmon elements are pre-configured, F5 AFM VE resources are provisioned and network is configured.



Pic.2 Lab blueprint

4.2.3 Licensing

BIG-IP is licensed automatically.

Evaluation license has been applied to Flowmon Collector/DDoS Defender. Please contact Lab admin if there are issues with any lab elements.

4.2.4 Other considerations

Note: Router1 is configured to export sFlow with sampling rate of 1

Note: Learn about sFlow:

https://sflow.org

4.3 Module – DDoS Attack

In this module you will prepare for and launch a SYN flood DoS attack. You will need an active RDP connection to a Linux Jumphost to perform all necessary prerequisites

4.3.1 Prepare traffic visualization and monitoring

- Connect to Windows jumphost using RDP
- Open SSH connections to Router1 and Router2
- **Verify Router1 BGP configuration. Protected subnet `10.1.30.0/24` should have a Next Hop defined as Route**

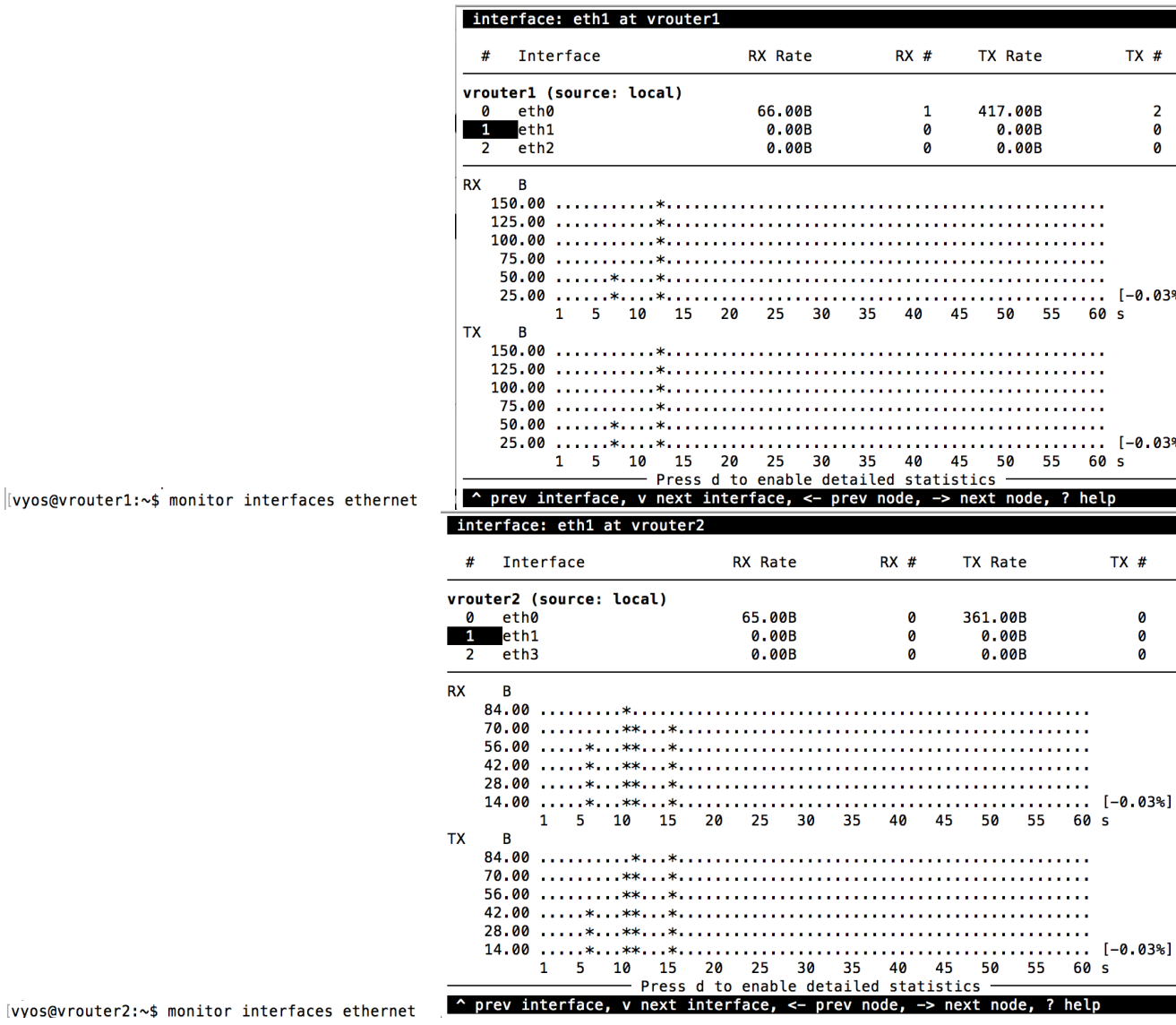
  `show ip bgp`

```
vyos@vrouter1:~$ show ip bgp
BGP table version is 0, local router ID is 10.1.10.243
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.10.0/24     0.0.0.0                  1         32768 i
*  10.1.30.0/24     10.1.20.244                            0 3 2 i
*>                  10.1.20.244              1             0 2 i

Total number of prefixes 2
```

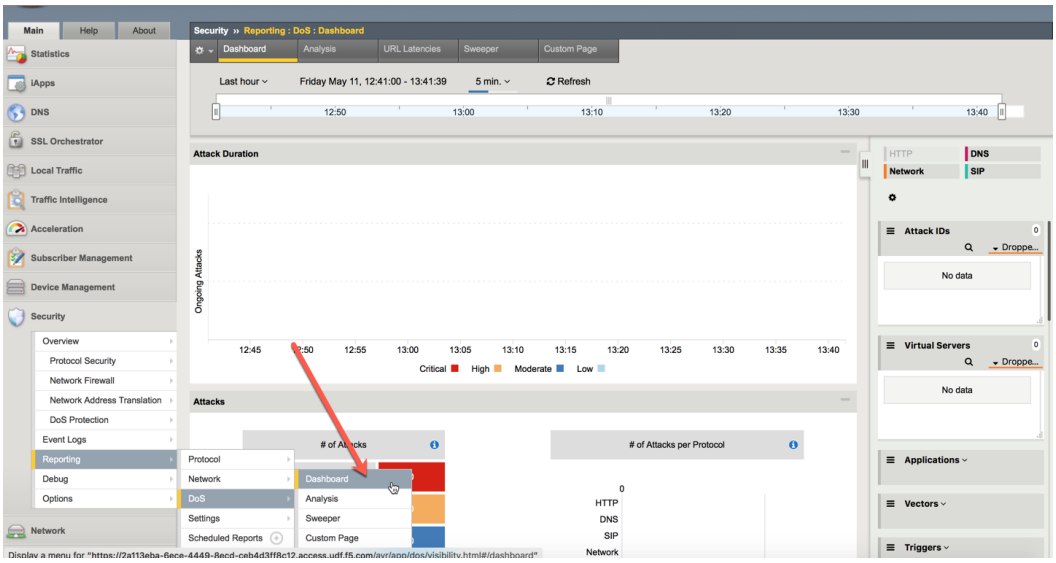
- Start interface monitoring in Router1 and Router2 `monitor interfaces ethernet`



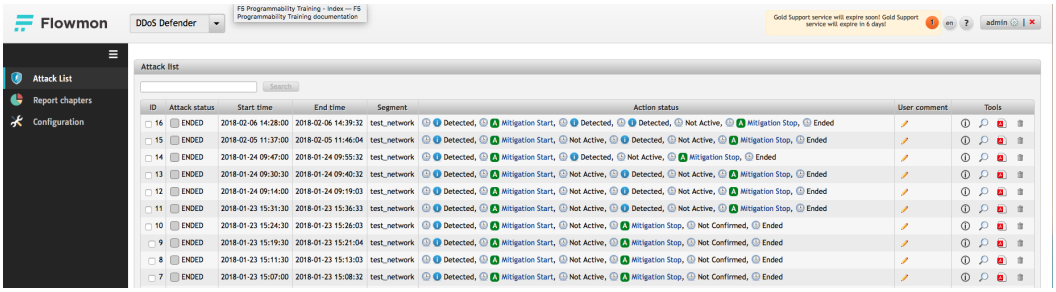
- Select *eth1* and press `g` to enable graphical statistics

Note: You may need to expand terminal window for graphs to appear

- Open Web Browser and click on *BIG-IP AFM* bookmark, then login into BIG-IP TMUI using `admin` credentials
- Open **DoS Visibility Dashboard** in AFM TMUI



- In a new Browser tab click on *Flowmon Web interface* bookmark. Once Flowmon main menu opens, click on *Flowmon DDoS Defender* icon and login using `admin` credentials
- Open **Attack List** in Flowmon DDoS Defender WebUI



Note: Disregard any active alarms Flowmon may show in the upper right screen corner. These are artifcts of this lab environment

4.3.2 Initiate DDoS attack

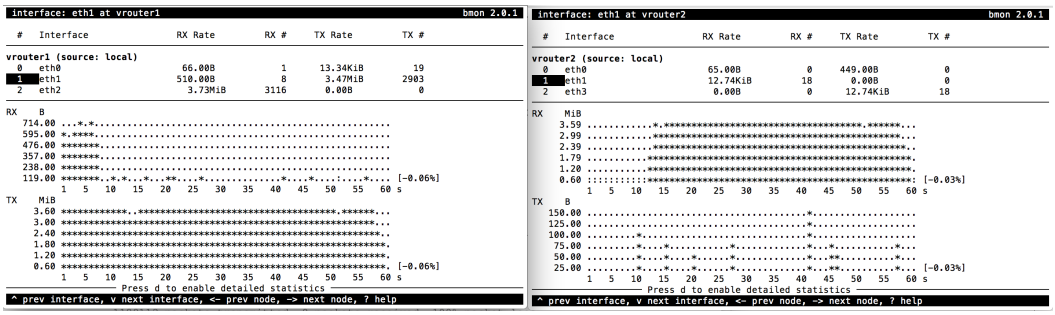
Run SYN flood (hping3) from Attacker VM

- Click on **Attacker SSH** icon to open `Attacker VM` ssh session
- From Attacker VM run SYN flood towards Web server

```
./syn_flood
```

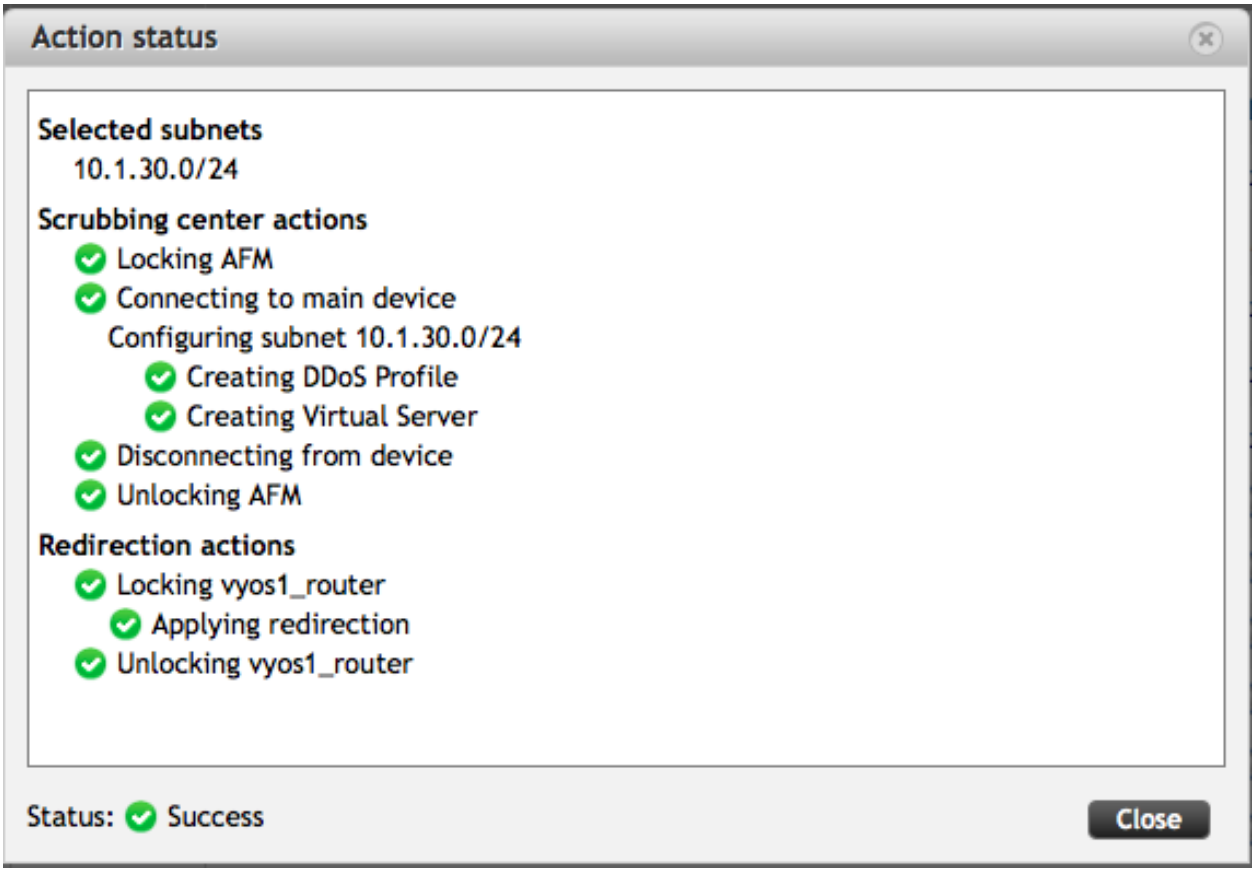
```
[f5admin@attacker:~$ ./syn_flood
[[sudo] password for f5admin:
HPING 10.1.30.252 (ens3 10.1.30.252): S set, 40 headers + 1200 data bytes
hping in flood mode, no replies will be shown
```

- Observe traffic growth in both Router1 and Router2. After **15-45 seconds** traffic will drop in Router2 due to DDoS detection and mitigation start



DDoS mitigation start

An *ACTIVE* attack with the new ID will appear in Flowmon DDoS defender 'Active attacks' screen. Flowmon dynamically provisions AFM DDoS profile and VS, and initiates traffic diversion to AFM using BGP advertisement



BGP route change and traffic drop

- Router1 shows new route to protected `10.1.30.0/24` subnet

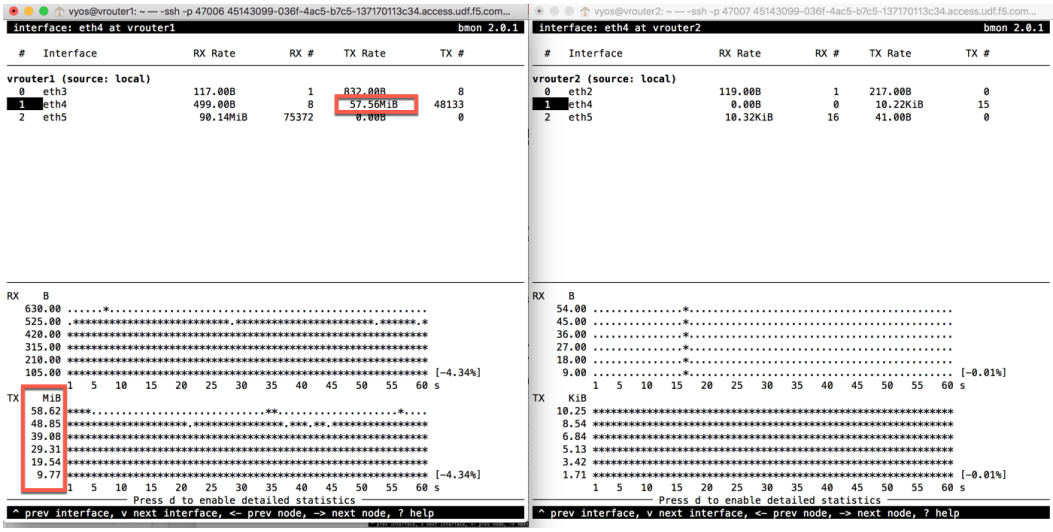
```
show ip bgp
```

```
vyos@vrouter1:~$ show ip bgp
BGP table version is 0, local router ID is 10.1.10.243
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 0.0.0.0          10.1.1.1                 0             32768 ?
*> 10.1.10.0/24     0.0.0.0                  1             32768 i
*>i10.1.30.0/24     10.1.20.245                    100      0 i
*                   10.1.20.244              1             0 2 i

Total number of prefixes 3
```

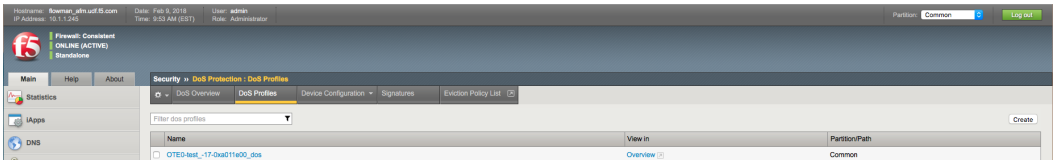
- As traffic is being routed through AFM, Router2 shows no significant network activity while Router1 still experiences high traffic load



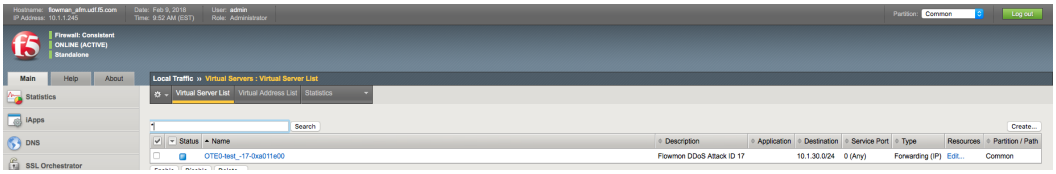
AFM DDoS profile and virtual server

Note: Flowmon uses iControl REST interface to provision necessary parameters in AFM

- In AFM TMUI Navigate to **Security –> DoS protection –> DoS profiles** and confirm that the DoS profile has been provisioned for the protected subnet

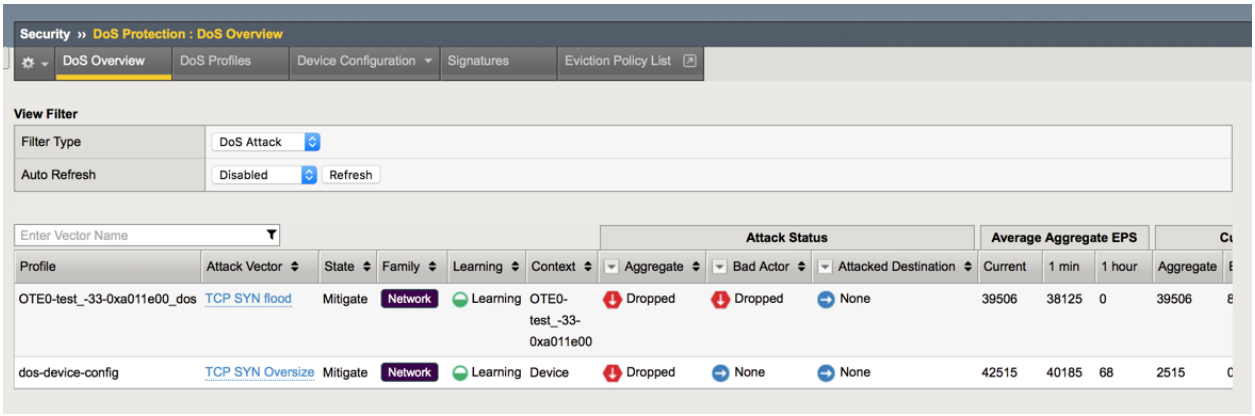


- In **Local Traffic –> Virtual Servers –> Virtual Server List** confirm that VS with corresponding Attack ID has been created



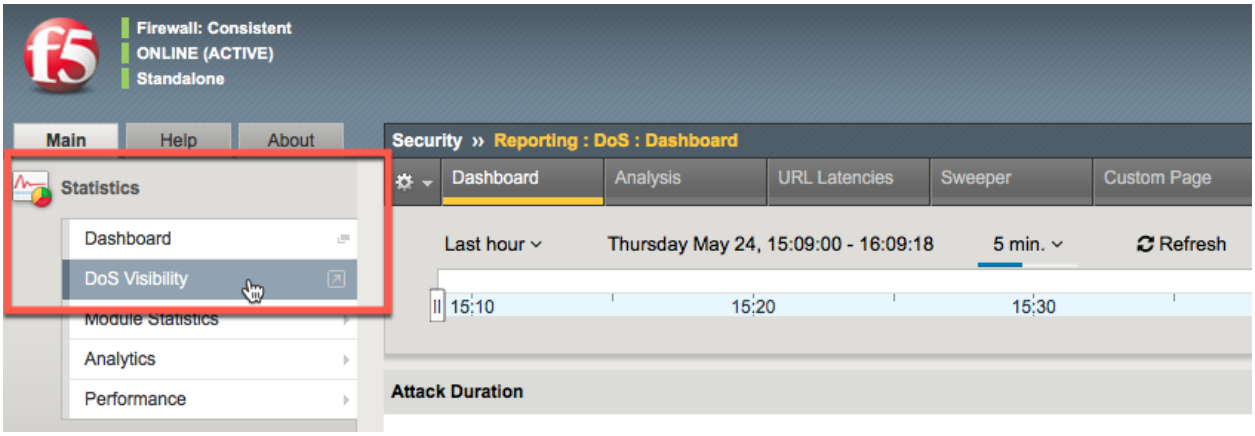
AFM DDoS mitigation

In AFM TMUI navigate to **Security –> DoS Protection –> DoS Overview** and confirm that AFM is performing DoS mitigation using the provisioned DoS profile



Note: *Statistics -> DoS Visibility* TMUI menu provides graphical attack data

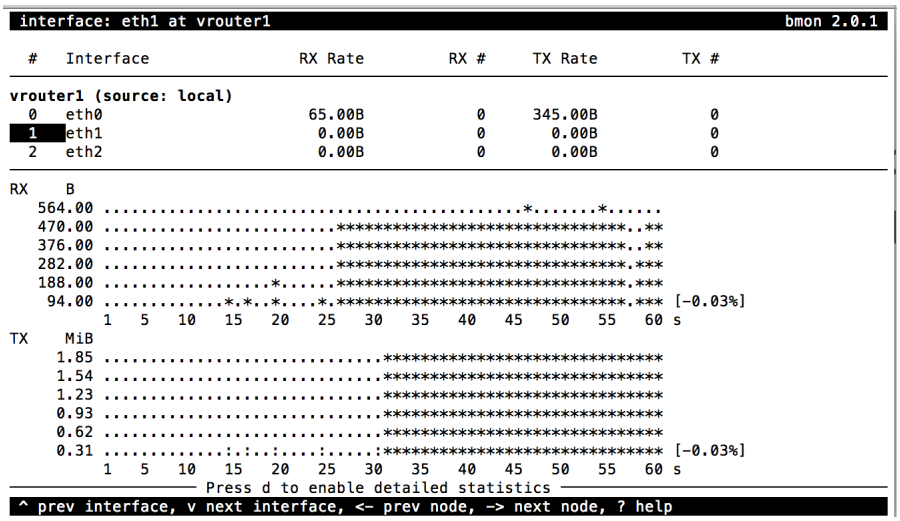
It may take up to ~5 minutes for DoS Visibility Dashboard to show our simulated DDoS attack. You may need to click *Refresh* for data to appear



4.3.3 Attack stop

Stop SYN flood

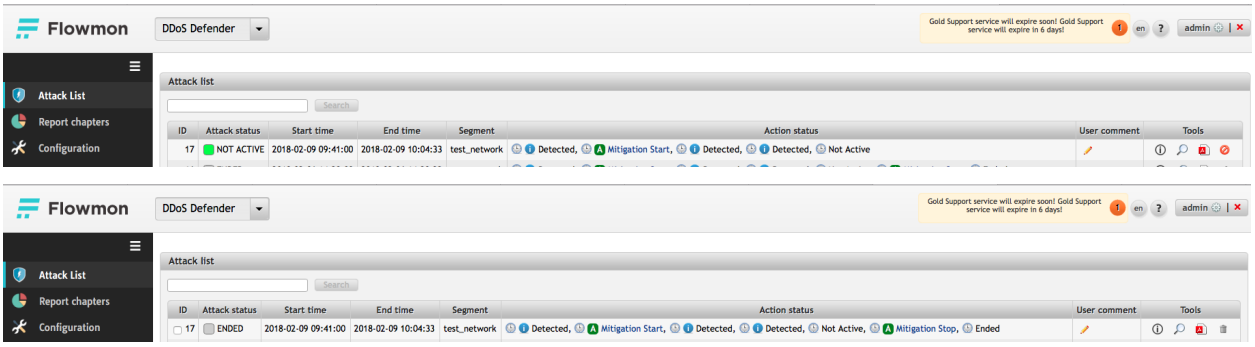
Press (`Ctrl-C`) to finish the attack. Traffic will drop on Router1

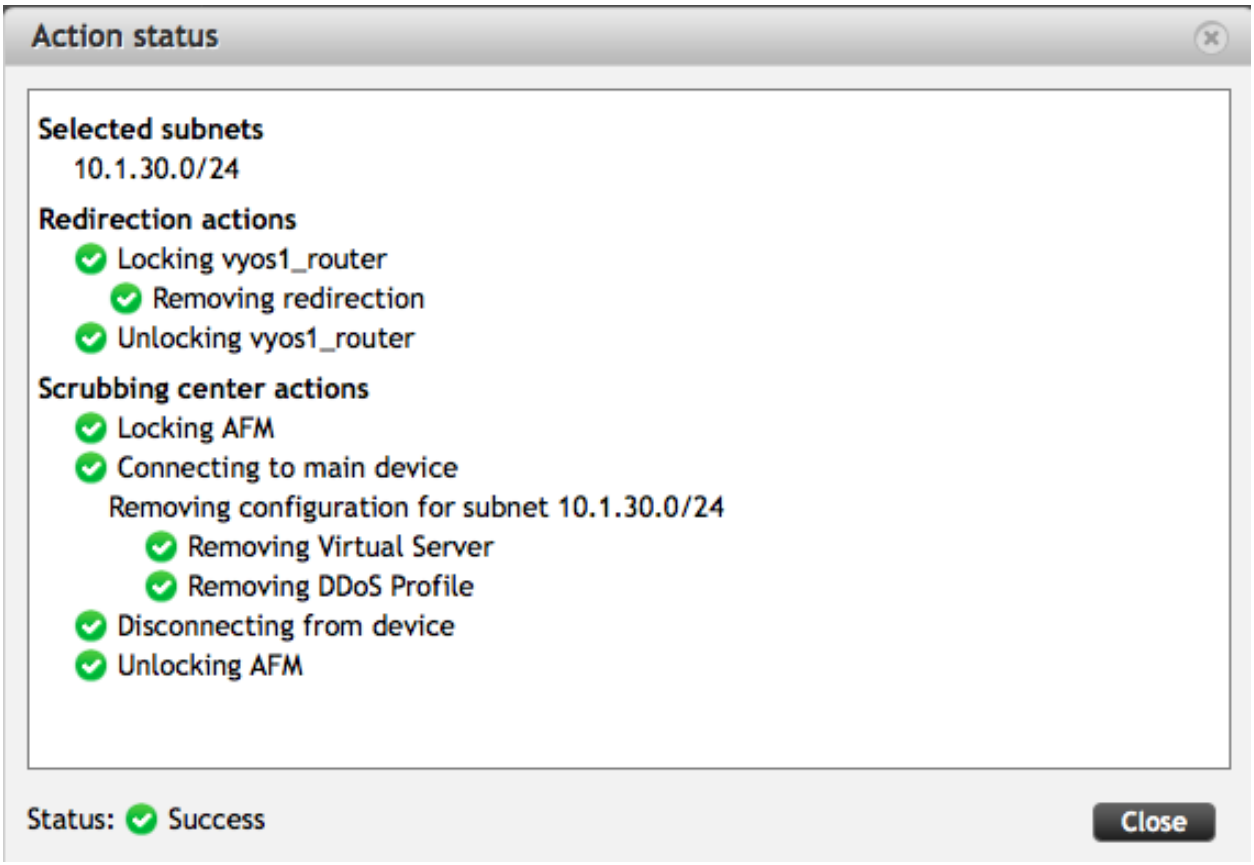


Note: STOP HERE. It will take 5-10 minutes for Flowmon to mark the attack as *NOT ACTIVE*. This is done in order to avoid 'flip-flop' effect in repeated attack situation

Mitigation stop

Flowmon DDoS Defender Attack List screen shows the current attack with status *NOT ACTIVE*. Attack will transition to *ENDED* state when Flowmon performs *Mitigation Stop* routine





**It typically takes ~ 5min for Flowmon DDoS Defender to update attack status*

AFM configuration, BGP route removal

As part of *Mitigation Stop* routine Flowmon removes BGP route from Router1 and Virtual Server and DDoS Profile from AFM

```
show ip bgp
```

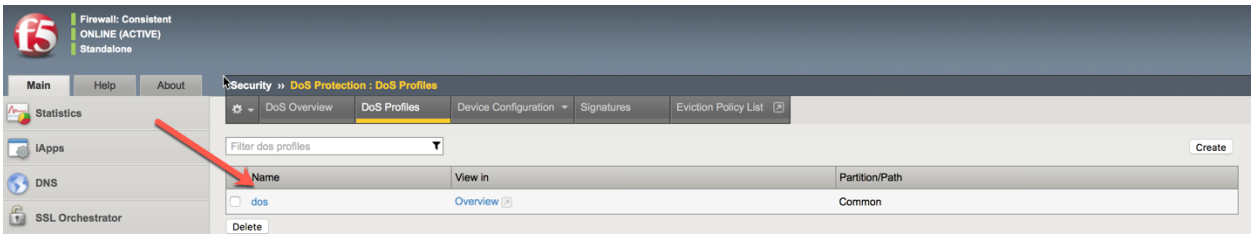
```
vyos@vrouter1:~$ show ip bgp
BGP table version is 0, local router ID is 10.1.10.243
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 0.0.0.0          10.1.1.1                 0         32768 ?
*> 10.1.10.0/24     0.0.0.0                  1         32768 i
*> 10.1.30.0/24     10.1.20.244              1             0 2 i

Total number of prefixes 3
```

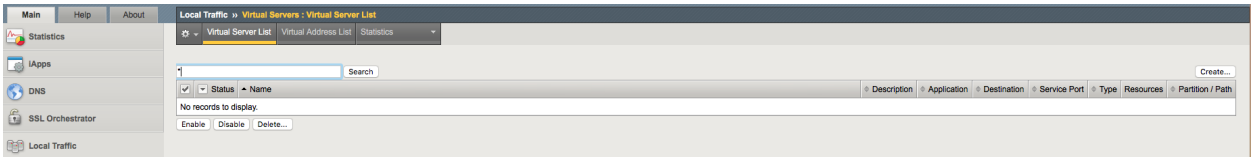
In AFM TMUI navigate to Security –> DoS Protection –> DoS Profiles

Verify that only default “dos” profile present



In AFM TMUI navigate to Local Traffic –> Virtual Servers –> Virtual Server List

Verify that Virtual Server matching Attack ID has been removed



Congratulations! You have successfully completed the lab!