Deploying the BIG-IP Dual-Stack Data Center Firewall With F5 Advanced Firewall Manager

Welcome to the F5 BIG-IP data center firewall Deployment Guide. This document provides guidance on configuring BIG-IP with AFM (Advanced Firewall Manager) and LTM (Local Traffic Manager) as a high-security, high-availability, high-performance dual-stack data center network firewall and IPv6/IPv4 gateway.

BIG-IP AFM is an ICSA Labs-certified network firewall which provides advanced network-protection capabilities meeting the needs of all organizations, including those with PCIDSS, FIPS, and HIPAA compliance requirements.

BIG-IP AFM and LTM provide superior security and functionality for organizations integrating IPv6 into their network architecture and operations.

This Deployment Guide includes extensive design information to help you bring BIG-IP security and performance to your existing networks.

For more information on the F5 BIG-IP platform, see *http://www.F5.com/products/big-ip*.

For more information on the F5 BIG-IP Advanced Firewall Manager (AFM), see *https://F5.com/products/modules/advanced-firewall-manager*.

Products and Versions

Product	Version
BIG-IP system (LTM, AFM)	11.6 - 12.0
BIG-IP iApp template	f5.datacenter_firewall_dg_quick_start.v1.0.0rc1
Deployment Guide version	1.2 (see *Document Revision History on page 74*)
Last updated	02-05-2016

Important: *Make sure you are using the most recent version of this deployment guide, available at http://f5.com/pdf/deployment-guides/f5-data-center-firewall-dg.pdf.*

To provide feedback on this Deployment Guide or other F5 solution documents, contact us at *solutionsfeedback@F5.com*.

Microsoft Partner Network

Contents

The BIG-IP Application Delivery Firewall

For years, sophisticated organizations have relied upon the F5 BIG-IP product family to manage and deliver application network traffic reliably, securely, and quickly. Very often this has included managing and balancing network traffic to and through network firewall devices, many of which cannot approach the performance of the BIG-IP system itself. Now the BIG-IP Advanced Firewall Manager (AFM) module provides a comprehensive, ICSA Labs-certified network firewall solution to protect the data center. BIG-IP AFM is integrated with BIG-IP Local Traffic Manager (LTM) and other BIG-IP modules.

During the long incubation period of IPv6 the F5 BIG-IP has gained recognition as the most effective tool for building transparent, secure, high-performance dual-stack corporate networks. The BIG-IP system's IPv6 fluency simplifies network security planning, operations, and management.

The F5 BIG-IP constitutes an Application Delivery Firewall platform which provides a unified view of Layers 3 through 7+ for both general and ICSA-mandatory reporting and alerting as well as integration with SIEM systems. The BIG-IP system's full-proxy architecture negates so called "advanced evasion technique" attacks which bypass many common firewalls.

The BIG-IP AFM module operates chiefly at OSI Layers 2 through 4, with significant functions at Layers 5 to 7. Additional BIG-IP modules—the Application Security Manager (ASM) and Access Policy Manager (APM)—provide application-layer firewall and AAA gateway services at Layers 7+ as well as TLS VPN services (L3+).

Combining the BIG-IP APM with AFM and/or ASM is particularly valuable, because APM enables identity-based security. With APM you may tune policy for authorized users rather than subject everyone to least-common-denominator firewall rules. More critically, you may track security alerts and issues back to individuals in many cases. You may also resolve advanced-persistent-threat, credential or identity theft, and even malfeasance problems much more effectively when you associate user identity to network activity.)

Because L2–7 network firewall security is foundational for information system protection, this Deployment Guide shows how to implement key policies to protect a typical data center network architecture using BIG-IP with AFM and LTM. You may replicate and adjust these policies to protect your own network and the information systems using it.

Prerequisites and configuration notes

The following are general prerequisites, assumptions, and notes about the configuration described in this guide.

- For the maximum benefit from this Deployment Guide, you should be familiar with network security concepts and the basics of the BIG-IP platform.

  The material in this Guide is not a substitute for the product documentation available at: *https://support.F5.com/kb/en-us/products/big-ip-afm/versions.11-6-0.html*.

- You must have at least one data center with an Internet link. The configuration in this guide supports public-facing and internal applications.

- We assume your users access applications in the data center as well as external services via the data center's Internet link. You may also have VPN links to branch-office, cloud-infrastructure, or business-partner networks with independent Internet connectivity.

- This guide describes an implementation in which the F5 BIG-IP with AFM and LTM is deployed between your primary ISP link and your intranet to secure inbound and outbound traffic against intrusions and DoS attacks (the BIG-IP system supports single and multiple ISP links with or without BGP routing). The BIG-IP AFM constitutes the primary *edge firewall*. You may also use AFM as an interior firewall.

- This guide shows you how to manage and secure both IPv6 and IPv4 traffic simultaneously. Both types of traffic are active on nearly all LANs today.

- Network security deployments implement specific rules and practices to mitigate the challenges of a threat model. For purposes of this Deployment Guide we do not explore the subject very deeply. Rather, we draw guidance from the *F5 DDoS Protection Reference Architecture* along with NIST Special Publications *800 41 rev1—Guidelines on Firewalls and Firewall Policy* and *800-119—Guidelines for the Secure Deployment of IPv6*, and the *Payment Card Industry (PCI) Data Security Standard Version 3.0 Requirements*. We also used the *BIG-IP Systems Network Firewall Guide for ICSA Certification* and other F5 reference and guidance documents, in addition to *RFC7123*, *RFC4890*, and other sources.

- Data center firewalls support both North-South (NS) and East-West (EW) use cases. *North-South* refers generally to network traffic between the Internet and the organizational intranet. *East-West* refers generally to network traffic between portions of the intranet which constitute different trust environments. Trust boundaries are defined by security policy (much of which is expressed in firewall rules) and need not mirror physical network layout.
- The BIG-IP AFM includes a number of additional features that are not a part of this guide, such as SNMP trap configuration, connection-eviction policy, SIP Protocol and DoS protection, and IPFIX setup.
- You must have access to the BIG-IP web-based Configuration utility (GUI) and command line.
- There is an iApp template available to make much of the initial configuration easier. See *https://devcentral.f5.com/codeshare/data-center-firewall-quick-start-iapp-template*. The iApp is not required, but will save configuration time.

This deployment guide covers the following topics:

For more detail on the concepts and configuration presented in this guide, see *Appendix A: Supplemental Information on page 49*.

Deployment Guide Model Network Architecture

Figure 1 shows key elements of the model network architecture which is the basis for the examples in this Deployment Guide.

In this diagram, the BIG-IP system is the gateway to the Internet for the data center and much of the intranet so the default routes (IPv6 and IPv4) on most devices point toward it (mostly via intermediate routers/L3 switches). However, branch office networks may have independent Internet access.

Network-management (e.g., SNMP) tools, system administrators' workstations, and infrastructure devices' management ports are homed on management network subnets.

The model network supports split-horizon DNS (separate Internet and intranet views) though we do not specify implementation details. IPv6 devices learn DNS server addresses from DHCPv6. A spam-filtering mail server handles both incoming and outgoing SMTP traffic.

Third-party security audits review static configuration files so this guide shows how to enforce explicit firewall policies on BIG-IP listeners. You will discourage sloppy configuration practices by running AFM in *Firewall mode*. In Firewall mode, no virtual server or self IP object will process any traffic until you enforce a named firewall policy on it (having at least one rule with an action of "Accept").

In the configuration described in this guide, the example organization is forbidden to do business with entities in certain countries listed by the US State Department under the International Traffic in Arms Regulations (ITAR regime) codified at *22 CFR §126.1*. The firewall policy must exclude traffic from those countries.

Finally, there is a log server which will accept logs as fast as the BIG-IP AFM can send them.

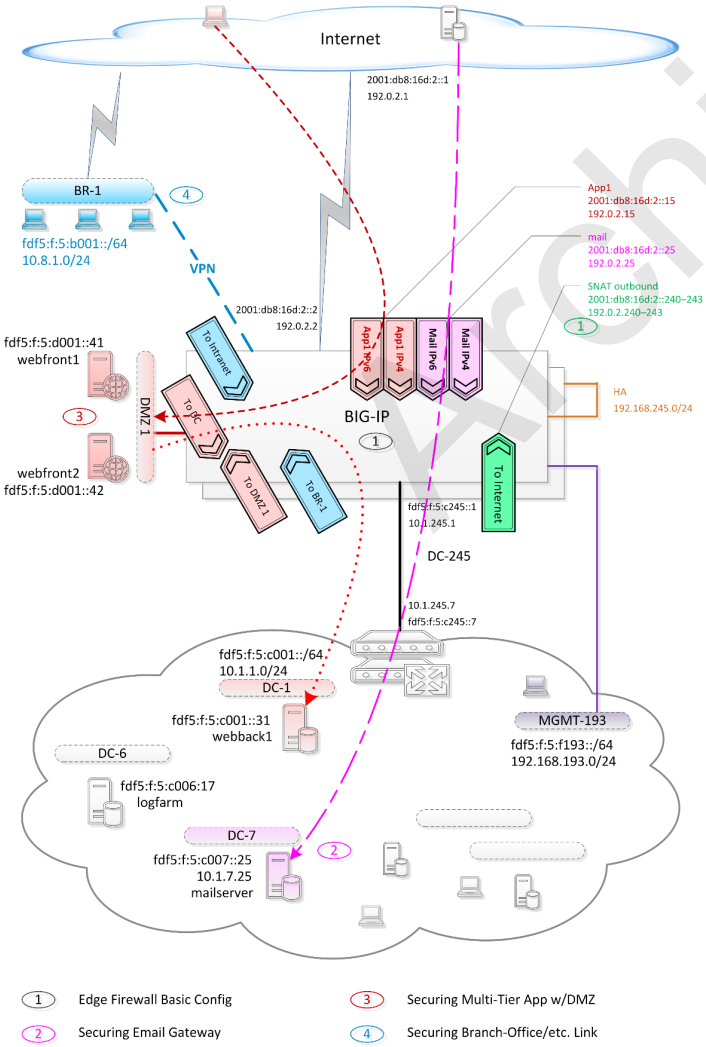

Figure 1: *Data Center Firewall Deployment Guide Model Network Architecture*

Allocating IP Addresses

With IPv6 as with IPv4 you will have two main blocks of addresses: public addresses reachable from the Internet and private addresses used in your intranet. All traffic between the Internet and your intranet requires address translation; this greatly aids security filtering and minimizes the cost of switching ISPs. Public addresses of outside (Internet) network correspondents will be visible in the intranet as source-address on inbound traffic and destination-address on outbound traffic. This averts any hassle with DNS and makes your logs easy to search and analyze.

Commonly, your ISP will give you a PA (Provider Assigned) IPv6 public address space and you will use a ULA (Unique Local Address) IPv6 private address space in your intranet. If you have PI (Provider-Independent) IPv6 address space, you may subnet it into public and private blocks or use it just for public addresses.

The examples in this document illustrate one way to align IPv4 and IPv6 private subnet numbers for the convenience of network administrators. For instance, one of the data center VLANs carries the paired subnets fdf5:f:5:c001::/64 and 10.1.1.0/24. The same digits are used for the third octet of the IPv4 subnet number and the last three columns of the fourth chunk of the IPv6 subnet number. The first column in the fourth chunk is a mnemonic code for a region of the intranet ('c' for data center—elsewhere we use 'd' for the DMZ); the intranet region also selects the first two octets of the IPv4 subnet number (10.1 for data-center—compare 172.30 for DMZ). While the explanation of this scheme may seem complicated, it is easier to put into practice.

Much network firewall policy is concerned with packet addressing. When you adapt examples from this document to create your own AFM configuration, you may find the following table useful. Note that you won't replace addresses in lists of standard subnets like *island-nets* (defined later in this document).

Description	Deployment Guide Example	Your value
Public IPv4 subnet	**192.0.2.0/24**	
Public IPv6 prefix + subnet	**2001:db8:16d:2::/64**	
Private IPv4 interior subnets	**10.W.X.0/24**	
Private IPv4 "management network" block of subnets	**192.168.192.0/18**	
Private IPv4 DMZ subnets	**172.30.X.0/24**	
Private IPv6 prefix + subnets	**fdf5:f:5:ZZZZ::/64**	
Private IPv6 "management network" block of subnets	**fdf5:f:5:f000/54**	
BIG-IP management subnet	**192.168.193.0/24**	
BIG-IP HA subnet	**192.168.245.0/24**	

Performing the initial configuration of the BIG-IP system for AFM

In this section, we provide guidance on the initial setup of the BIG-IP system for the Advanced Firewall Manager module.

Important *In the following section, and throughout the guide, we use example values and object names based on our configuration. Change these to the appropriate values and names for your configuration.*

1. ***Install your BIG-IP devices according to F5 documentation***
   It is outside the scope of this document to provide instructions on installing the BIG-IP system in your network. Refer to the BIG-IP documentation available on the F5 technical support web site: *https://support.f5.com*.

2. ***Use a redundant pair of BIG-IP devices (recommended)***
   This guide is written with the assumption you have a pair of BIG-IP devices for redundancy. In our examples, our BIG-IP devices are named **big-s1** and **big-s2**. F5 ScaleN deployments with additional devices are also feasible, but not detailed in this guide.

3. ***Configure the management addresses***
   Each BIG-IP must have an IPv4 or IPv6 management-port address and management default route. Typically you configured these addresses during the initial BIG-IP configuration. You can find the settings in **System > Platform**. In the User Administration area, leave **SSH Access** set to **Enabled** and **SSH IP Allow** set to *** All Addresses**. We show how to secure management SSH access later. The following table shows our example devices and their associated management IP addresses.

BIG-IP device	Management IP address (default route)
big-s1 (big-s1.example.net)	192.168.193.21/24 (192.168.193.1)
big-s2 (big-s1.example.net)	192.168.193.22/24 (192.168.193.1)

4. ***Ensure AFM is provisioned on your BIG-IP system***
   Go to **System > Resource Provisioning > Configuration**, and make sure **Advanced Firewall (AFM)** is set to **Nominal**. If it is not, select **Nominal** from the list, click **Submit** and then wait for the BIG-IP system to update.

5. **Configure VLANs on your BIG-IP system**
   On each BIG-IP device, configure the following VLANs. For VLAN configuration, go to **Network > VLANs**.

VLAN	Remarks
external	link to Internet gateway [standard]
internal	link to intranet/data-center gateway (router/L3-switch) [standard]
HA	links redundant BIG-IP devices [standard]. You may use a (link aggregation) trunk to downmux BIG-IP HA (ConfigSync and Mirroring) traffic across multiple L2 ports for added performance. Consult *BIG-IP TMOS Concepts* for details.

6. **Configure the BIG-IP self IP addresses on each device**
   On the first BIG-IP device (**big-s1** in our example) use the following table for guidance on creating self IP addresses, using the addresses appropriate for your implementation. All self IP addresses should be created in the **/Common** partition.
   For Self IP configuration, go to **Network > Self IPs**.

Name	IP Address	Netmask	VLAN	Port Lockdown	Traffic Group
self-ext-s1-4	192.0.2.3	255.255.255.0	external	Allow Default	traffic-group-local-only
self-ext-s1-6	2001:db8:16d:2::3	ffff:ffff:ffff:ffff::	external	Allow Default	traffic-group-local-only
self-int-s1-4	10.1.245.2	255.255.255.0	internal	Allow Default	traffic-group-local-only
self-int-s1-6	fdf5:f:5:c245::2	ffff:ffff:ffff:ffff::	internal	Allow Default	traffic-group-local-only
self-ha-s1	192.168.245.21	255.255.255.0	HA	Allow Default	traffic-group-local-only

   On the second BIG-IP device (**big-s2** in our example) create self IP addresses using the following table as guidance. All self IP addresses should be created in the **/Common** partition.

Name	IP Address	Netmask	VLAN	Port Lockdown	Traffic Group
self-ext-s2-4	192.0.2.4	255.255.255.0	external	Allow Default	traffic-group-local-only
self-ext-s2-6	2001:db8:16d:2::4	ffff:ffff:ffff:ffff::	external	Allow Default	traffic-group-local-only
self-int-s2-4	10.1.245.3	255.255.255.0	internal	Allow Default	traffic-group-local-only
self-int-s2-6	fdf5:f:5:c245::3	ffff:ffff:ffff:ffff::	internal	Allow Default	traffic-group-local-only
self-ha-s2	192.168.245.22	255.255.255.0	HA	Allow Default	traffic-group-local-only

On both devices create floating self IP addresses using the following table as guidance.
All self IP addresses should be created in the **/Common** partition.

Name	IP Address	Netmask	VLAN	Port Lockdown	Traffic Group
self-ext-float-4	192.0.2.2	255.255.255.0	external	Allow Default	traffic-group-1
self-ext-float-6	2001:db8:16d:2::2	ffff:ffff:ffff:ffff::	external	Allow Default	traffic-group-1
self-int-float-4	10.1.245.1	255.255.255.0	internal	Allow Default	traffic-group-1
self-int-float-6	fdf5:f:5:c245::1	ffff:ffff:ffff:ffff::	internal	Allow Default	traffic-group-1

7. **Configure NTP, DNS, and named.conf**
On each device: configure *NTP (Network Time Protocol) (SOL3122)*. Also *configure DNS Lookup Server List plus BIND Forwarder Server List (SOL13205)*. Then *correct each BIG-IP's named.conf file per SOL12224*. Optionally configure a remote syslog server in System / Logs / Configuration / Remote Logging.

8. **Update the IP Geolocation database**
Download the latest IP Geolocation database update per SOL11176. Install it on each device.

9. **Configure High Availability**
You can add both BIG-IP devices to a high availability Sync-Failover group using the Configuration utility on one of them.

   The BIG-IP HA link in our example crosses only one VLAN so we just use IPv4 for HA communications. Use of IPv6 on BIG-IP HA links is supported but slightly less efficient. If you decide to use IPv6 for BIG-IP high availability traffic, review *SOL15816*.

   Log in to the first BIG-IP device, big-s1 in our example.

   a. On the Main tab, click **Device Management > Devices**.
      - Click **big-s1.example.net (Self)**.
      - On the Menu bar, click Device **Connectivity > Config Sync**.
      - From the Local Address list, select **192.168.245.21 (HA)**.
      - Click **Update**.

   b. On the Menu bar, click **Device Connectivity > Failover Network**.
      - Ensure the **Failover Unicast List** contains 192.168.193.21 (VLAN Management Address) and 192.168.245.21 (VLAN HA). If these addresses do not appear in the list, click the **Add** button to add the missing address(es).
      - Leave the **Use Failover Multicast Address** unchecked (do NOT check this box).
      - Click **Update**.

   c. On the Menu bar, click **Device Connectivity > Mirroring**.
      - From the **Primary Local Mirror Address** list, select 192.168.245.21 (HA).
      - From the **Secondary Local Mirror Address** list, select 10.1.245.2 (internal).
      - Click **Update**.

   d. On the Main tab, click **Device Management > Device Trust > Peer List**.
      - Click **Add**.
      - In the **Device IP Address** field, type **192.168.99.22** (the management IP address of big-s2).
      - In the **Administrator Username** field, type **admin**.
      - In the **Administrator Password** field, type the associated password.
      - Click **Retrieve Device Information**. The other BIG-IP device's host name (big-s2.example.net in our example) should appear in the **Device Properties Name** field.
      - Click **Finished**.

   e. On the Main tab, click **Device Management > Device Groups**.
      - Click **Create**.

- In the **Name** field, type **ha-group-1**.
- From the **Group Type** list, select **Sync-Failover**.
- In the Members area, use the Add button (**<<**) to move both **big-s1.example.net** and **big-s2.example.net** from **Available** to **Includes**.
- Check the **Network Failover** box.
- Leave **Automatic Sync** unchecked (NOT checked).
- Click **Finished**.

f. On the Main tab, click **Device Management > Overview**.

- In the **Device Groups** area, click **ha-group-1**.
- In the **Devices** area, click **big-s1.example.net (Self)**.
- In the Sync Options area that appears, click Sync Device to Group.
- Check the **Overwrite Configuration** box.
- Click **Sync**.

This completes the high availability configuration.

F5 DevCentral offers a good tutorial the BIG-IP HA setup: *ScaleN: A Network Architect-Engineer's Unofficial Guide To ScaleN Clustering*.

10. **Add the IP routes to the BIG-IP system**
On each BIG-IP device, add the following IP routes in the **/Common** partition. Note that if you encounter an error message similar to *01070712:3: Cannot create static route: ::/0 gw 2001:db8:16d:2::1 on interface '' in rd0 - netlink error: 113 (No route to host)* when using TMOS 11.6, upgrade to 11.6 Hotfix 4 or later.

Name	Destination	Netmask	Resource	Gateway
default-4	0.0.0.0	0.0.0.0	Use Gateway…	192.0.2.1
default-6	::	::	Use Gateway…	2001:db8:16d:2::1
intranet-4-pA	10.0.0.0	255.0.0.0	Use Gateway…	10.1.245.7
intranet-4-pB	172.16.0.0	255.240.0.0	Use Gateway…	10.1.245.7
intranet-4-pC	192.168.0.0	255.255.0.0	Use Gateway…	10.1.245.7
intranet-6	fdf5:f:5::	ffff:ffff:ffff::	Use Gateway…	fdf5:f:5:c245::7

11. On each BIG-IP device, use the virtual console or an SSH client such as PuTTY to access the BIG-IP command line and execute the following TMSH commands to enable IPv6 Neighbor Discovery on the intranet VLAN only. For more details see *SOL13580: Configuring neighbor discovery for IPv6*. Leave the 'A' flag set in RA's.
In the model network architecture, servers learn DNS settings from DHCPv6:

```
create /net router-advertisement ra-internal vlan internal enabled prefixes add { /Common/pfx-internal { prefix
 fdf5:f:5:c245:: prefix-length 64 } router } managed other-config
```

If you set up a network without DHCPv6, substitute this command:

```
create /net router-advertisement ra-internal vlan internal enabled prefixes add { /Common/pfx-internal { prefix
fdf5:f:5:c245:: prefix-length 64 } router }
```

Save your changes:

```
save /sys config
```

Configuring a DHCPv6 Relay

IPv6 emphasizes dynamic configuration. Still, as with IPv4, devices using IPv6 may be given static IP addresses[1], default routes, and/or DNS settings. However, in many intranets even devices with static IPv6 addresses use DHCPv6 to get DNS settings. When you operate DHCPv6 servers you should make the BIG-IP device a DHCPv6 relay. If necessary, you can also *configure a IPv4 DHCP relay*.

In the model network, the DHCPv6 server address is **fdf5:f:5:c006::9**. Additional DHCPv6/DHCP servers are supported; you can simply add them to the pool(s). Use the following table to configure the appropriate objects. For specific instructions, see the Help tab or the product documentation.

Tip: *While your IP addresses will almost always be different from our examples, you may find it easier to follow the configuration in this guide if you use the same object names as in our examples. Some objects are called from other objects, which in turn are called from another object (and so on). Our guidance always refers to our example names.*

Pools (*Navigate to Local Traffic > Pools*)	
Name	**DHCPv6-pool**
Health Monitor	Select **gateway_icmp**.
New Members	Node Name: **dhcpv6-server1** Node Address: **fdf5:f:5:c006::9** Service Port: **547**
Virtual Servers (*Navigate to Local Traffic > Virtual Servers*)	
Name	**DHCPv6-relay-vs**
Type	**DHCP**
Destination Address	**ff02::1:2** (IPv6 Default)
DHCP Profile	From the first list, select **DHCPv6**. From the second list, under /Common select **dhcpv6**.
VLAN and Tunnel Traffic	Select **Enabled On**, and then select only the **Internal** VLAN you created.
Default Pool	Select the DHCPv6 pool you created

Note: You may download an iApp to perform much of the following configuration from: *https://devcentral.f5.com/codeshare/data-center-firewall-quick-start-iapp-template*

Use the virtual console or an SSH client such as PuTTY to access the BIG-IP command line on each device and execute the following TMSH commands:

```
modify /sys syslog iso-date enabled
modify /sys db tm.icsastricttcpforwarding value enable
modify /sys db dos.dropv4mapped value true
modify /ltm global-settings general snat-packet-forward enabled
save /sys config
```

From this point, unless otherwise noted you may add configuration changes to one BIG-IP device and then propagate them using configsync.

[1] It is customary for servers in LTM load balancing pools (members, nodes) to have static IP addresses, though *LTM will let you identify nodes and members by FQDN mapped to IP address by DNS*

Performing the Basic AFM Configuration for Edge Firewall Use

The first job of a data center firewall is to block everything coming in until you decide what to allow. Even after that, the firewall must block unwanted traffic trying to get in through any ports you open. The firewall's second job is to let traffic go out, except for suspicious traffic trying to reach destinations you don't approve of. Think of your data center firewall as a fortress with eagle-eyed sentries. You can get out; you can let your friends in; but it frustrates your enemies and their spies.

With BIG-IP AFM, you layer application-specific security policy onto a base of a global policy aimed at generic threats. This section shows how to establish that base layer.

For a review of AFM structure and configuration, see *BIG-IP AFM Overview on page 50* and subsequent topics.

Note: *You can download an iApp to perform much of the following configuration. See https://devcentral.f5.com/codeshare/data-center-firewall-quick-start-iapp-template*

Creating the Network Firewall Address Lists

Reusable elements make firewall rules and policies more efficient and maintainable. For instance, *address lists* let you replace blocks of numbers with mnemonic names and streamline updates to addresses used in multiple rules.

Use the following guidance to create all of the following address lists, adding subnets (CIDR notation) individually. Instead of clicking **Finished** after adding each list, you can click **Repeat** to save time.

Address Lists (*Navigate to Security > Network Firewall > Address Lists*)				
Name	**island-nets**			**Notes (not a part of the configuration)**
Description	Standard *RFC6890*			
Addresses/Regions	Type or copy and paste the following addresses into the Add new entry box:			May not be src or dst of packet on the wire. (::/96 catches IPv6 self and loopback.) ::ffff:0:0/96 is purposefully omitted until allowed by a future F5 TMOS update[1].
	100.64.0.0/10	**::/96**	**2001:10::/28**	
	127.0.0.0/8	**100::/64**	**2001:20::/28**	
	169.254.0.0/16	**2001:2::/48**	**2001:db8::/32**	
	240.0.0.0/4			
Name	**unicast-nets**			As of 2015, covers all public IPv6 unicast. (We "notch out" some IPv4 nets.)
Description	Standard			
Addresses/Regions	**0.0.0.0/1**	**192.0.0.0/3**		
	128.0.0.0/2	**2000::/3**		
Name	**mcast-nets**			Always invalid as src address
Description	Standard			
Addresses/Regions	**224.0.0.0/4**	**ff00::/8**		
Name	**nonpub-nets**			Not allowed from or to public Internet. We treat TEST-NET-n as private-use.
Description	Standard *RFC6890*			
Addresses/Regions	**0.0.0.0/8**	**192.0.2.0/24**	**203.0.113.0/24**	
	10.0.0.0/8	**192.168.0.0/16**	**64:ff9b::/96**	
	172.16.0.0/12	**198.18.0.0/15**	**fc00::/7**	
	192.0.0.0/24	**198.51.100.0/24**		
Name	**bad-6to4-nets**			
Description	Standard			
Addresses/Regions	**2002::/24**	**2002:ac10::/28**	**2002:c612::/31**	
	2002:0a00::/24	**2002:c000::/40**	**2002:c633:6400::/40**	
	2002:6440::/26	**2002:c000:0200::/40**	**2002:cb00:7100::/40**	
	2002:7f00::/24	**2002:c0a8::/32**	**2002:e000::/19**	
	2002:a9fe::/32			

[1] Refer to f5 ID456376. This Guide sets TMOS db variable dos.dropv4mapped to avert any security issue.

Address Lists (*Navigate to Security > Network Firewall > Address Lists*)		
Name	**intranet-nets**	**Notes (not a part of the configuration)**
Description	Example	
Addresses/Regions	These addresses are our examples. Adjust for your network if necessary. **10.0.0.0/8** | **192.168.0.0/16** **172.16.0.0/12** | **fdf5:f:5::/48**	All intranet subnets (includes all-dmz-nets; specific rules control DMZ access)
Name	**net-mgmt-nets**	
Description	Example	
Addresses/Regions	These addresses are our examples. Adjust for your network if necessary. **192.168.192.0/18** | **fdf5:f:5:f000::/54**	Device management, SNMP tools, RADIUS servers, etc.
Name	**all-dc-nets**	
Description	Example	
Addresses/Regions	These addresses are our examples. Adjust for your network if necessary. **10.1.0.0/16** | **fdf5:f:5:c000::/54**	All Data Center subnets
Name	**our-pub-nets**	
Description	Example	
Addresses/Regions	These addresses are our examples. Adjust for your network if necessary. **192.0.2.0/24** | **2001:db8:16d:2::/64**	Public Internet-accessible subnets
Name	**outbound-snat-nets**	
Description	Example	
Addresses/Regions	These addresses are our examples. Adjust for your network if necessary. **192.0.2.240/30** | **2001:db8:16d:2::240/126**	Addresses used to SNAT outbound connections from intranet to Internet
Name	**mailserver**	
Description	Example	
Addresses/Regions	These addresses are our examples. Adjust for your network if necessary. **10.1.7.25/32** | **fdf5:f:5:c007::25/128**	To see "mailserver" in AFM rules rather than the addresses
Name	**ITAR-countries**	
Description	From *22 CFR 126.1* updated 2015–03–27	
Addresses/Regions	**Belarus (BY)** | **Korea, Democratic People's Republic of (KP)** **Congo, The Democratic Republic of the (CD)** | **Lebanon (LB)** **Liberia (LR)** **Cote D'Ivoire (CI)** | **Libyan Arab Jamahiriya (LY)** **Cuba (CU)** | **Somalia (SO)** **Eritrea (ER)** | **Sudan (SD)** **Iran, Islamic Republic of (IR)** | **Syrian Arab Republic (SY)** **Iraq (IQ)** | **Venezuela (VE)**	This is an example of how you can use geolocation, and is optional. When you really implement ITAR restrictions, update this list and the date in the Description.

Creating the Network Firewall Port Lists

Use the following table for guidance to create all of the following port lists, adding port numbers individually. Again, because there are so many lists, instead of clicking **Finished** after adding each list, you can click **Repeat** to save time.

Port Lists (*Navigate to Security > Network Firewall > Port Lists*)		
Name	***Port***	**Notes (not a part of the configuration)**
dns	**53**	
ftp	**21**	FTP uses additional ports—ALG required
http	**80**	
https	**443**	
imap-tls	**993**	
kerberos	**88, 464**	
ldap	**389**	
ldaps	**636**	
ms-ad-gc	**3268, 3269**	Active Directory Global Catalog (LDAP on the wire)
ms-sql	**1433**	
ntp	**123**	
smtp-mta	**25**	
smtp-mua	**587**	
smtp-all	**25, 587**	
snmp-mgr	**161, 10161**	Only network managers may send *to* these ports.
smtp-trap	**162, 10162**	10162 for DTLS/TLS
ssh	**22**	
syslog-udp	**514**	
syslog-tls	**6514**	
tripwire	**1169**	
udp-dhcp	**67, 68, 546, 547**	
udp-EW-basic	**dns, ntp, snmp-trap, syslog-udp, tripwire**	Once you have created a Port list, you can use that list in a new list. Begin typing and the Port list will appear in the box, preceded by /Common. DMZ servers typically access self-secured critical intranet services on these ports. If you do not use Tripwire you may omit the port.
tcp-EW-basic	**dns, smtp-mua, syslog-tls, tripwire**	If you do not use Tripwire you may omit the port. DMZ servers typically access self-secured critical intranet services on these ports.
udp-EW-generic	**kerberos**	Kerberos is often needed with tcp-EW-generic and tcp-rdp-smb.
tcp-EW-generic	**http, https, imap-tls, kerberos, ldap, ldaps, ms-ad-gc, smtp-mta, ssh**	basic+generic+rdp+smb is a good starting point for screening branch-office or workstation nets. When DMZ servers must access specific services (e.g., backend resources) on these ports you should create corresponding BIG-IP virtual servers, or at least add specific firewall rules (e.g., src dmz:any dst w.x.y.z:https). However, when that is too difficult, you might permit basic+generic and rely on (destination) host/service security.
tcp-rdp-smb	**445, 3389**	Remote desktop, file server access, and so on.
u-t-tsp-ayiya	**3653, 5072**	TSP or AYIYA tunnels (VPN/policy evasion)

This completes the Network Firewall Port list configuration.

Creating the Network Firewall Rule

The next task is to create the following iRule. This iRule is tied to the global IP Intelligence policy you define later, but you create it now so you can reference it in network firewall rules.

From the Main tab, expand **Local Traffic** and then click **iRules**. Click the **Create** button. Give the iRule a unique name (we use **afm-no-evil-dst**), and then use the following code in the Definition section, omitting the line numbers:

```
1   when FLOW_INIT {
2      set okay [list "cloud_provider_networks"]
3
4      set r [IP::intelligence [IP::local_addr]]
5      if {[lsearch -exact $r "_WHITELIST_"] >= 0} { return }
6      foreach {cat} $r {
7         if {[lsearch -exact $okay $cat] < 0} {
8            #log local0.info "denied outbound to [IP::local_addr] = ${cat}"
9            ACL::action reset ; #reject
10           return
11        }
12     }
13  }
```

Creating the Network Firewall Rule list

The next task is to create the Firewall rule lists. You first create the Rule list with a name and optional description, and then add rules to it. As you edit each rule, extra fields appear when you change a list from **Any** to **Specify**.

For the first list, the following table details the configuration. For the subsequent rule lists, use the second table of values for guidance.

Rule Lists (*Navigate to Security > Network Firewall > Rule Lists*)		
Rule List Name	**global-forbid-nets** Click **Finished** and then from the Rule Lists table, click the Rule List you just created.	
Rules	Click the **Add** button and add the following rules.	
	Name	**no-src-island-mcast**
	Protocol	**Any**
	Source > Address/Region	Add **island-nets** and **mcast-nets**
	Source > Port	**Any**
	Source > VLAN/Tunnel	**Any**
	Destination > Address/Region	**Any**
	Destination > Port	**Any**
	Action	**drop** Click **Repeat** and then add the following rule.
	Name	**no-dst-island-4lnk**
	Protocol	**Any**
	Source > Address/Region	**Any**
	Source > Port	**Any**
	Source > VLAN/Tunnel	**Any**
	Destination > Address/Region	Add **island-nets** and **0.0.0.0/8**
	Destination > Port	**Any**
	Action	**drop** Click **Finished**

Rule List Name	Rule Name	Protocol	Source address	Src. Port	Src. VLAN	Dst. Address	Dst. Port	Action
control-from-Internet	ok-ipv6-link	Any	fe80::/10	Any	external	fe80::/10 ff02::/16	Any	Accept
	no-fwd-link	Any	fe80::/10	Any	external	Any	Any	Drop
	no-src-nonpub-etc	Any	nonpub-nets bad-6to4-nets intranet-nets	Any	external	Any	Any	Drop
	no-dst-nonpub-etc	Any	Any	Any	external	nonpub-nets intranet-nets	Any	Drop
	ok-mcast-link-global	Any	unicast-nets	Any	external	224.0.0.0/4 ff02::/16 ff0e::/16	Any	Accept
	no-other-mcast	Any	Any	Any	external	mcast-nets	Any	Drop
	ok-valid-src	Any	unicast-nets	Any	external	Any	Any	Accept
	no-other-src	Any	Any	Any	external	Any	Any	Drop

Rule List Name	Rule Name	Protocol	Source address	Src. Port	Src. VLAN	Dst. Address	Dst. Port	Action
block-selfIP-mgmt	no-https	TCP	Any	Any	Any	Any	HTTPS	Drop
	no-ssh	TCP	Any	Any	Any	Any	SSH	Drop
	no-snmp-u	UDP	Any	Any	Any	Any	snmp-mgr	Drop
	no-snmp-t	TCP	Any	Any	Any	Any	snmp-mgr	Drop
permit-dhcp	ok-dhcp	UDP	Any	udp-dhcp	Any	Any	udp-dhcp	Accept
filter-outbound	no-dst-nonpub-etc	Any	Any	Any	Any	nonpub-nets intranet-nets 192.88.99.0/24	Any	Reject
	no-send-mcast	Any	Any	Any	Any	mcast-nets	Any	Reject
	no-tsp-ayiya-t	TCP	Any	Any	Any	Any	u-t-tsp-ayiya	Reject
	no-tsp-ayiya-u	UDP	Any	Any	Any	Any	u-t-tsp-ayiya	Reject
	ok-smtp-mailserver	TCP	mailserver	Any	Any	Any	smtp-all	Accept
	For this Rule, you also need to attach the iRule you created in *Creating the Network Firewall Rule on page 13*							
	no-smtp-others	TCP	Any	Any	Any	Any	smtp-all	Reject
	ok-tcp	TCP	intranet-nets	Any	Any	Any	Any	Accept
	For this Rule, you also need to attach the iRule you created in *Creating the Network Firewall Rule on page 13*							
	ok-udp	UDP	intranet-nets	Any	Any	Any	Any	Accept
	For this Rule, you also need to attach the iRule you created in *Creating the Network Firewall Rule on page 13*							
	no-other	Any	Any	Any	Any	Any	Any	Reject

Creating the ICMP Rule list

The next task is to create a global ICMP control rule list as described in the following table. For each rule in the table, after you select the Protocol type you must choose ICMP type-and-code combinations and Add them to the message list.

Rule Lists (*Navigate to Security > Network Firewall > Rule Lists*)

global-icmp-control Rule list

Rule List Name	**global-icmp-control** Click **Finished** and then from the Rule Lists table, click the Rule List you just created.	
Rules	Click the **Add** button and add the following rules.	
	Name	**no-ping6-snats**
	Protocol	**ICMPv6** Make sure **58** appears in the box on the right.
	ICMPv6 Message	***Type*** / ***Code***: **Echo Request (128)** / **any**
	Source > Address/Region	**any**
	Source > Port	**any**
	Source > VLAN/Tunnel	**any**
	Destination > Address/Region	Add **outbound-snat-nets**
	Destination > Port	**any**
	Action	**drop** Click **Repeat** and then add the following rule.
	Name	**ok-always-icmp6**
	Protocol	**ICMPv6** Make sure **58** appears in the box on the right.
	ICMPv6 Message	***Type*** / ***Code*** *(click Add after each)*: **Destination Unreachable (1)** / **Any**; **Packet Too Big (2)** / **Any**; **Time Exceeded (3)** / **Hop limit exceeded in transit (0)**; **Time Exceeded (3)** / **Fragment reassembly time exceeded (1)**; **Parameter Problem (4)** / **Erroneous header field encountered (0)**; **Parameter Problem (4)** / **Unrecognized Next Header type encountered (1)**; **Parameter Problem (4)** / **Unrecognized IPv6 option encountered (2)**; **Echo Request (128)** / **Any**; **Echo Reply (129)** / **Any**

Rule Lists (*Navigate to Security > Network Firewall > Rule Lists*)

Rules continued

Source > Address/Region	**Any**
Source > Port	**Any**
Source > VLAN/Tunnel	**Any**
Destination > Address/Region	**Any**
Destination > Port	**Any**
Action	**Accept Decisively** Click **Repeat** and then add the following rule.

Name	**ok-local-icmp6**
Protocol	**ICMPv6** Make sure **58** appears in the box on the right.
ICMPv6 Message	

Type	***Code*** *(then click Add)*
Multicast Listener Query (130)	**Any**
Multicast Listener Report (131)	**Any**
Multicast Listener Done (132)	**Any**
Router Solicitation (133)	**Any**
Router Advertisement (134)	**Any**
Neighbor Solicitation (135)	**Any**
Neighbor Advertisement (136)	**Any**
Inverse Neighbor Discovery Solicitation Message (141)	**Any**
Inverse Neighbor Discovery Advertisement Message (142)	**Any**
Version 2 Multicast Listener Report (143)	**Any**
Certification Path Solicitation Message (148)	**Any**
Certification Path Advertisement Message (149)	**Any**
Multicast Router Advertisement (151)	**Any**
Multicast Router Solicitation (152)	**Any**
Multicast Router Termination (153)	**Any**

Source > Address/Region	**fe80::/10** *(If needed in your network, add destination prefixes like site-local multicast ff05::/16)*
Source > Port	**Any**
Source > VLAN/Tunnel	**Any**
Destination > Address/Region	Add **fe80::/10** and **ff02::/16**
Destination > Port	**Any**
Action	**Accept Decisively** Click **Repeat** and then add the following rule.

Name	**no-unk-icmp6**
Protocol	**ICMPv6** Make sure **58** appears in the box on the right.
ICMPv6 Message	

Type	***Code***
Any	**Any**

Source > Address/Region	**Any**
Source > Port	**Any**
Source > VLAN/Tunnel	**Any**
Destination > Address/Region	**Any**
Destination > Port	**Any**
Action	**Drop** Click **Repeat** and then add the following rule.

Name	**ok-snats-icmp4**
Protocol	**ICMP** Make sure **1** appears in the box on the right. Note the change from ICMPv6 to ICMP.
ICMP Message	

Type	***Code*** *(then click Add)*
Echo Reply (0)	**No Code (0)**
Destination Unreachable (3)	**Net Unreachable (0)**
Destination Unreachable (3)	**Host Unreachable (1)**
Destination Unreachable (3)	**Fragmentation needed and Don't Fragment… (4)**
Destination Unreachable (3)	**Communication Administratively Prohibited (13)**
Time Exceeded (11)	**Time To Live exceeded in transit (0)**
Time Exceeded (11)	**Fragment Reassembly Time Exceeded (1)**

Rule Lists (*Navigate to Security > Network Firewall > Rule Lists*)		
Rules Continued	***Source > Address/Region***	**Any**
	Source > Port	**Any**
	Source > VLAN/Tunnel	**Any**
	Destination > Address/Region	**outbound-snat-nets**
	Destination > Port	**Any**
	Action	**Accept Decisively** Click **Repeat** and then add the following rule.
	Name	**no-snats-icmp4**
	Protocol	**ICMP** Make sure **1** appears in the box on the right.
	ICMP Message	***Type*** / ***Code***: **Any** / **Any**
	Source > Address/Region	**Any**
	Source > Port	**Any**
	Source > VLAN/Tunnel	**Any**
	Destination > Address/Region	**outbound-snat-nets**
	Destination > Port	**Any**
	Action	**Drop** Click **Repeat** and then add the following rule.
	Name	**ok-others-icmp4**
	Protocol	**ICMP** Make sure **1** appears in the box on the right.
	ICMP Message	***Type*** / ***Code*** *(then click Add)*: **Echo Reply (0)** / **No Code (0)**; **Destination Unreachable (3)** / **Net Unreachable (0)**; **Destination Unreachable (3)** / **Host Unreachable (1)**; **Destination Unreachable (3)** / **Fragmentation needed and Don't Fragment… (4)**; **Destination Unreachable (3)** / **Communication Administratively Prohibited (13)**; **Echo Request (8)** / **No Code (0)**; **Time Exceeded (11)** / **Time To Live exceeded in transit (0)**; **Time Exceeded (11)** / **Fragment Reassembly Time Exceeded (1)**
	Source > Address/Region	**Any**
	Source > Port	**Any**
	Source > VLAN/Tunnel	**Any**
	Destination > Address/Region	**Any**
	Destination > Port	**Any**
	Action	**Accept Decisively** Click **Repeat** and then add the following rule.
	Name	**no-unk-icmp4**
	Protocol	**ICMP** Make sure **1** appears in the box on the right.
	ICMP Message	***Type*** / ***Code***: **Any** / **Any**
	Source > Address/Region	**Any**
	Source > Port	**Any**
	Source > VLAN/Tunnel	**Any**
	Destination > Address/Region	**Any**
	Destination > Port	**Any**
	Action	**Drop** Click **Finished**.

This completes the Network Firewall Rule list configuration.

Creating the Network Firewall Policies

The next task is to create the Network Firewall policies on the BIG-IP AFM. Use the following tables for guidance on configuring Network Firewall policies.

Network Firewall Policies (*Navigate to Security > Network Firewall > Policies*)		
Accept All Policy		
Description of policy	When AFM is in *Firewall mode* you must *enforce* a permissive named firewall policy while you *stage* a candidate selective policy on any object (otherwise the implicit policy for that object will discard all packets). The following is the permissive policy.	
Policy Name	**accept-all** Click **Finished** and then from the Policies table, click the Policy you just created.	
Rules	Click the **Add** button and add the following rule.	
	Name	**ok-all**
	Protocol	**Any**
	Source > Address/Region	**Any**
	Source > Port	**Any**
	Source > VLAN/Tunnel	**Any**
	Destination > Address/Region	**Any**
	Destination > Port	**Any**
	Action	**Accept** Click **Finished**.
Global Policy		
Description of policy	The global policy disposes of many erroneous, spoofed, fuzzed, and garbage packets. It also deals with ICMP (In TMOS 11.6 only global and route-domain firewall policies may filter ICMP traffic). Lacking a "VLAN context" you will filter all packets from the Internet using rules keyed to VLAN *external* in the global context. This saves adding multiple rules to every public-facing virtual server's policy.	
Policy Name	**global** Click **Finished** and then from the Policies table, click the Policy you just created.	
Rules	Click the **Add** button and add the following rules	
	Name	**g-icmp**
	Type	**Rule List**
	Rule List	From the list, under /Common, select the **global-icmp-control** Rule List you created
	Click **Repeat** and then add the following rule	
	Name	**g-bad-nets**
	Type	**Rule List**
	Rule List	From the list, under /Common, select the **global-forbid-nets** Rule List you created
	Click **Repeat** and then add the following rule	
	Name	**g-ctrl-inet**
	Type	**Rule List**
	Rule List	From the list, under /Common, select the **control-from-Internet** Rule List you created
	Click **Finished**	
Self IP Policy		
Description of policy	BIG-IP systems operate best when Port Lockdown is set to Default on all self IP addresses. However, you should apply a firewall policy to restrict interactive management traffic to the management port. The global policy also protects self IP's.	
Policy Name	**global** Click **Finished** and then from the Policies table, click the Policy you just created.	
Rules	Click the **Add** button and add the following rules	
	Name	**no-mgmt**
	Type	**Rule List**
	Rule List	From the list, under /Common, select the **block-selfIP-mgmt** Rule List you created
	Click **Repeat** and then add the following rule	
	Name	**ok-others**
	Protocol	**Any**
	Source > Address/Region	**Any**
	Source > Port	**Any**
	Source > VLAN/Tunnel	**Any**
	Destination > Address/Region	**Any**
	Destination > Port	**Any**
	Action	**Accept** Click **Finished**.

Network Firewall Policies (*Navigate to Security > Network Firewall > Policies*)		
DHCP relay Policy		
Description of policy	Like other listeners, DHCP relay virtual servers must be secured	
Policy Name	**ok-dhcp** Click **Finished** and then from the Policies table, click the Policy you just created.	
Rules	Click the **Add** button and add the following rule.	
	Name	**no-mgmt**
	Type	**Rule List**
	Rule List	From the list, under /Common, select the **permit-dhcp** Rule List you created
	Click **Finished**	
Outbound filter Policy		
Description of policy	A single policy will suffice for several "outbound SNAT" virtual servers. The Rule list used in this policy (filter-outbound) includes a rule to prevent intranet devices sending packets to public multicast addresses. You might have to insert exceptions for some applications (e.g., a streaming-media server). See the no-send-mcast entry in *Creating the Network Firewall Rule list on page 14*.	
Policy Name	**outbound** Click **Finished** and then from the Policies table, click the Policy you just created.	
Rules	Click the **Add** button and add the following rule.	
	Name	**ob-filter**
	Type	**Rule List**
	Rule List	From the list, under /Common, select the **filter-outbound** Rule List you created
	Click **Finished**	

This completes the Network Firewall Policy configuration.

IP Intelligence

The model network architecture uses IP geolocation with network firewall rules to exclude connections from countries on the ITAR list. Other uses for IP geolocation include blocking regional script kiddies and enforcing copyright license boundaries. As a backstop to IP geolocation, in this example we configure a global IP Intelligence policy to exclude connections from proxy services that might be hiding unwelcome source addresses. An F5 IP Intelligence subscription is strongly recommended but not required.

As a key DoS defense we exclude other bad-reputation source addresses, including networks blacklisted by Spamhaus.org, as well as bogons with help from Team Cymru. To reduce the load on your Internet link and BIG-IP AFM, try persuading your ISP to drop bogons upstream.

The best practice is to whitelist the public addresses of business partners plus your own public addresses, especially those of site-to-site VPN endpoints. Initially the whitelist will contain 192.0.2.0/24 and 2001:db8:16d::/48 in our example.

Creating the iRule to retrieve address lists

First, we create this iRule to help fetch certain address lists.

From the Main tab, expand **Local Traffic** and then click **iRules**. Click the **Create** button. Give the iRule unique name (we use **afm-feedlist-helper**), and then use the following code in the Definition section, omitting the line numbers:

```
# Team Cymru's bogon list and the Spamhaus.com E+DROP
# lists include comments and use addr/masklen rather
# than the CSV addr,masklen format the v11.6 Feed List
# parser wants.  To read those or similar lists (any
# with lines that begin "addr/masklen") into AFM:
#
# Create a virtual server with HTTP Profile on a non-
# routeable IP like 198.18.0.1:80.  Attach this iRule.
# No pool.  Use a Feed URL source URL like:
#   http://198.18.0.1/www.spamhaus.org/drop/drop.txt
# with the real hostname in the first part of the URL
# path.  This iRule will proxy requests to the real
# host and reformat the response to suit AFM.
#
# Ensure BIG-IP DNS resolver is setup per SOL12224.

when HTTP_REQUEST {
 if {([HTTP::method] ne "GET") ||
     ![regexp {^/([^/]+)(/.*)$} [HTTP::uri] x h u] ||
     ([set ip [lindex [RESOLV::lookup $h] 0]] eq "")} {
  log local0.err "unable to resolve ${h}"
  HTTP::respond 502 ; return
 }
 HTTP::uri $u
 HTTP::version "1.0" ; #avert response-chunking
 HTTP::header replace Host $h
 HTTP::header replace Accept-Encoding identity
 HTTP::header replace Connection close
 node $ip 80
}

when HTTP_RESPONSE {
 if {[HTTP::status] != 200} { return }
 if {[set clen [HTTP::header Content-Length]] eq ""} {
  set clen 2097152
 }
 HTTP::collect $clen
}

when HTTP_RESPONSE_DATA {
 #discard all but addresses, make those CSV
 set r {(?w)(?:(?:(^[0-9A-Fa-f:.]+/[0-9]{1,3})[^\n]*(\n))}
 append r {|(?:(?!^[0-9A-Fa-f:.]+/)[^\n]*\n))}
 regsub -all $r [HTTP::payload] {\1\2} buf
 set buf [string map {/ ,} $buf]
 HTTP::payload replace 0 [HTTP::payload length] $buf
 HTTP::release
}
```

Creating a virtual server for the iRule

The next task is to create a helper virtual server to run the iRule. Create this virtual server separately on each BIG-IP device (**big-s1** and **big-s2** in our example) and move its virtual address to traffic-group none.

Virtual Servers (*Navigate to Local Traffic > Virtual Servers*)	
Name	Type a unique name, such as **feedlist-helper-vs**
Type	**Standard**
Destination Address	**198.18.0.1**
HTTP Profile	From the list, under /Common, select **http**.
VLAN and Tunnel Traffic	Select **Enabled On**, and then select only the **Internal** VLAN you created.
iRules	Enable the **afm-feedlist-helper** iRule you created. Do not select a Default pool for this virtual server.

Modifying the Traffic Group

Once you have created the virtual server, you must modify the Traffic Group setting. If necessary, click **Local Traffic > Virtual Servers**. On the menu bar, click **Virtual Address List**. Click the IP address of the virtual server you just created (198.18.0.1 in our example). From the **Traffic Group** list, select **None**, and then click **Update**.

Creating blacklist categories and feed lists

The next task is to create blacklist categories and then Feed Lists to populate them. Use the following table for guidance.

Black List Categories (*Navigate to Security > Network Firewall > IP Intelligence > Black List Categories*)	
Bogons category	
Black List Category Name	**bogons**
Description	Unallocated or invalid IPv6 prefixes and IPv4 blocks. Click **Repeat** and then create the following category.
DROP_EDROP category	
Black List Category Name	**DROP_EDROP**
Description	Malevolent source networks. Click **Finished**

Next, create the associated feed lists using the following table as guidance.

Feed Lists (*Navigate to Security > Network Firewall > IP Intelligence > Feed Lists*)		
Bogons Feed List		
Name	**bogons-feed**	
Description	bogons courtesy of Team Cymru	
Feed List Properties: Feed URLs	***Name***	**bogons-6**
	URL	**http://198.18.0.1/www.team cymru.org/Services/ Bogons/fullbogons-ipv6.txt**
	List Type	**Black List**
	Blacklist Category	**bogons**
	Poll Interval	**14400** Click **Add** and then create the following Feed URL.
	Name	**bogons-4**
	URL	**http://198.18.0.1/www.team cymru.org/Services/ Bogons/fullbogons-ipv4.txt**
	List Type	**Black List**
	Blacklist Category	**bogons**
	Poll Interval	**14400** Click **Add** and then click **Finished**.
DROP_EDROP feed Feed List		
Name	**DROP_EDROP**	
Description	Don't Route Or Peer courtesy of Spamhaus.org	
Feed List Properties: Feed URLs	***Name***	**DROP**
	URL	**http://198.18.0.1/www.spamhaus.org/drop/drop.txt**
	List Type	**Black List**
	Blacklist Category	**DROP_EDROP**
	Poll Interval	**86400** Click **Add** and then create the following Feed URL.

Feed Lists (*Navigate to Security > Network Firewall > IP Intelligence > Feed Lists*)		
Feed List Properties: Feed URLs	***Name***	**EDROP**
	URL	**http://198.18.0.1/www.spamhaus.org/drop/edrop.txt**
	List Type	**Black List**
	Blacklist Category	**DROP_EDROP**
	Poll Interval	**86400** Click **Add** and then click **Finished**.

Creating the Local whitelist feed

Another Feed List is required to populate the IP Intelligence policy's whitelist. Before you create it, identify a web or FTP server to provide the whitelist addresses and note the corresponding URL so you can point a Feed URL at it. If you lack a suitable server, see *Appendix B: Storing IP Intelligence Address Lists in BIG-IP Data Groups on page 62* for a workaround .

Feed Lists (*Navigate to Security > Network Firewall > IP Intelligence > Feed Lists*)		
Name	**bogons-feed**	
Description	Trusted addresses for our company and our business partners.	
Feed List Properties: Feed URLs	***Name***	**local-whitelist-feed**
	URL	URL of the Web or FTP server. If you do not have one, see Appendix X on *page 62*
	List Type	**White List**
	Blacklist Category	**spam_sources** (What you choose here does not matter; it can be anything)
	Poll Interval	**600**

Creating an IP Intelligence policy

The final task for IP Intelligence is to create a policy that uses the objects you created in this section. Even if your BIG-IP system does not have an F5 IP Intelligence subscription, you can still use custom Feed Lists and black list categories.

IP Intelligence Policy (*Navigate to Security > Network Firewall > IP Intelligence > Policies*)		
Name	**global-IPI-policy**	
Description	Prevents connections from bogus and malicious source addresses	
IP Intelligence Properties	***Feed Lists***	Use the Add arrows (<<) to move the three Feed Lists you created (bogons-feed, DROP_EDROP-feed, and local-whitelist-feed) to the **Selected** box.
	Default Action	**Drop**
	Default Log Actions	Check the **Log Blacklist Category Matches** and **Log Whitelist Overrides** boxes.
	Blacklist Matching Policy	From the **Blacklist Category** list, select each of the following categories and then click **Add** (leave the Action and both log lists at the default (**Use Policy Default**)
		bogons
		botnets
		denial_of_service
		DROP_EDROP
		illegal_websites
		infected_sources
		phishing
		proxy
		scanners
		spam_sources
		web_attacks
		windows_exploits

Configuring AFM High-Speed Logging

The AFM sends log messages at high speed to a log server. Use this section to configure the objects for high-speed logging.

Pools (*Navigate to Local Traffic > Pools*)		
Name	**afm-log-pool**	
Health Monitor	Select **gateway_icmp**.	
New Members	Node Name	**logfarm**
	Node Address	**fdf5:f:5:c006::17**
	Service Port	**514**
Log Destination (*Navigate to System > Logs > Configuration > Log Destinations*)		
Remote High-Speed log destination		
Name	**afm-log-dest**	
Type	**Remote High-Speed Log**	
Pool Name	Select the pool you created (**afm-log-pool** in our example)	
Protocol	**UDP**	
Remote Syslog destination		
Name	**afm-syslog-dest**	
Type	**Remote Syslog**	
Syslog Format	**BSD Syslog**	
High-Speed Log Destination	**afm-log-dest**	
Log Publisher (*Navigate to System > Logs > Configuration > Log Publishers*)		
Name	**afm-log-pub**	
Destinations	From the Available box, select the **afm-syslog-dest** destination and then use the Add arrows (<<) to move it to the Selected box.	
Logging Profile (*Navigate to Security > Event Logs > Logging Profiles*)		
Profile Name	**afm-log-pfl**	
Protocol Security	Check the box to **Enable** Protocol Security.	
Network Firewall	Check the box to **Enable** Network Firewall.	
DoS Protection	Check the box to **Enable** DoS Protection.	
Protocol Security Tab	Click the Protocol Security tab.	
	HTTP, FTP, and SMTP Security: Publisher	Select the Log Publisher you created (**afm-log-pub** in our example).
Network Firewall Tab	Click the Network Firewall tab.	
	Network Firewall Publisher	Select the Log Publisher you created (**afm-log-pub** in our example).
	Log Rule Matches	Click the **Drop** and **Reject** boxes. Leave the Rate Limit set to **Indefinite**
	Log Translation Fields	Check the box to **Enable** Log Translation Fields
	Storage Format	Select **Field List**, and leave **Delimiter** set to a comma (**,**) From the **Available Items** list, select the following formats and use the Add (**<<**) button to move them to the **Selected Items** list in the following order: **date_time** **bigip_hostname** **context_name** **acl_policy_name** **acl_rule_name** **action** **protocol** **vlan** **src_ip** **src_port** **dest_ip** **dest_port** **drop_reason** **source_user** **translated_vlan** **translated_src_port** **translated_src_ip** **translated_dest_port** **translated_dest_ip**
	IP Intelligence Publisher	Select the Log Publisher you created (**afm-log-pub** in our example).
DoS Protection Tab	Click the DoS Protection tab.	
	Network DoS Protection: Publisher	Select the Log Publisher you created (**afm-log-pub** in our example).

Global DoS Protection

The Dos Protection Device Configuration (global) policy is always enforced. As explained in *Understanding AFM DoS Protection on page 59*, you must analyze your application traffic before adjusting this policy. Here you simply enable logging to permit data to accumulate for analysis.

DoS Protection Log Publisher (*Navigate to Security > DoS Protection > Device Configuration*)	
Log Publisher	From the list, select the Log Publisher you created (**afm-log-publisher** in our example). Click **Update**.

Preparing for outbound traffic

For outbound access through the data center firewall, use the following tables for guidance on configuring the following objects.

Creating the SNAT Pool

A SNAT pool represents a pool of translation addresses that you configure on the BIG-IP system. Use the following table for guidance on configuring a SNAT pool.

SNAT Pool (*Navigate to Local Traffic > Address Translation > SNAT Pool List*)	
Name	Type a unique name. In our example, we use **outbound-snatpool**
Member List: IP Address	Type the appropriate SNAT addresses, clicking the **Add** button after each. In our example, we type: **192.0.2.240** **192.0.2.241** **192.0.2.242** **192.0.2.243** **2001:db8:16d:2::240** **2001:db8:16d:2::241** **2001:db8:16d:2::242** **2001:db8:16d:2::243**

Configuring the BIG-IP system for outbound FTP

Use the following table for guidance on configuring the BIG-IP system for outbound FTP.

Pools (*Navigate to Local Traffic > Pools*)	
IPv4 Gateway pool	
Name	Type a unique name. In our example, we use **internet-gw-4-pool**
Health Monitor	Select **gateway_icmp**.
New Members	Node Name: **internet-gw-4** Node Address: **192.0.2.1** Service Port: ***All Services (0)**
IPv6 Gateway pool	
Name	Type a unique name. In our example, we use **internet-gw-6-pool**
Health Monitor	Select **gateway_icmp**.
New Members	Node Name: **internet-gw-6** Node Address: **2001:db8:16d:2::1** Service Port: ***All Services (0)**
Virtual Servers (*Navigate to Local Traffic > Virtual Servers*)	
IPv4 FTP Gateway virtual server	
Name	**ftp-outbound-4-vs**
Type	**Standard**
Destination Address	**0.0.0.0/0**
Service Port	**21 (FTP)**
FTP Profile	From the list, under /Common select **ftp**.
VLAN and Tunnel Traffic	Select **Enabled On**, and then select only the **Internal** VLAN you created.
Connection Mirroring	Check the box to **Enable** Connection Mirroring. You must select Advanced from the list at the top of the Configuration area.
Source Address Translation	**SNAT**
SNAT Pool	Select the SNAT pool you created (**outbound-snatpool** in our example).
Default Pool	Select the IPv4 gateway pool you created (**internet-gw-4-pool** in our example).

IPv6 FTP Gateway virtual server	
Name	**ftp-outbound-6-vs**
Type	**Standard**
Destination Address	**::/0**
Service Port	**21 (FTP)**
FTP Profile	From the list, under /Common select **ftp**.
VLAN and Tunnel Traffic	Select **Enabled On**, and then select only the **Internal** VLAN you created.
Connection Mirroring	Check the box to **Enable** Connection Mirroring. Select Advanced from Configuration list (top of pane) for this option to appear.
Source Address Translation	**SNAT**
SNAT Pool	Select the SNAT pool you created (**outbound-snatpool** in our example).
Default Pool	Select the IPv4 gateway pool you created (**internet-gw-6-pool** in our example).

Configuring the BIG-IP system for outbound TCP

Use the following table for guidance on configuring the BIG-IP system for outbound TCP traffic.

Rather than mirror outbound TCP connections, we enable Loose Initiation and Loose Close, and then disable Reset on Timeout in a Fast L4 Profile (see *SOL7595*). This averts loss of end-user connectivity in the event of a BIG-IP Traffic Group failover. Note that you typically use connection mirroring with application layer gateways (ALGs) like the outbound FTP virtual servers.

Fast L4 Profiles (*Navigate to Local Traffic > Profiles > Protocol > FastL4*)	
TCP Loose Fast L4 Profile	
Name	Type a unique name. In our example, we use **tcp-loose-fastl4**
Reset on Timeout	Clear the box to **Disable** Reset on Timeout
TCP Handshake Timeout	Leave **Specify** selected, and then in the **Seconds** box, type **10**
Loose Initiation	Check the box to **Enable** Loose Initiation
Loose Close	Check the box to **Enable** Loose Close
Virtual Servers (*Navigate to Local Traffic > Virtual Servers*)	
IPv4 FTP Gateway virtual server	
Name	**tcp-outbound-4-vs**
Type	**Forwarding (IP)**
Destination Address	**0.0.0.0/0**
Service Port	*** (* All ports)**
Protocol	**TCP**
Protocol Profile (Client)	Select the TCP Loose Fast L4 profile you created (**tcp-loose-fastl4** in our example).
VLAN and Tunnel Traffic	Select **Enabled On**, and then select only the **Internal** VLAN you created.
Source Address Translation	**SNAT**
SNAT Pool	Select the SNAT pool you created (**outbound-snatpool** in our example).
IPv6 FTP Gateway virtual server	
Name	**tcp-outbound-6-vs**
Type	**Forwarding (IP)**
Destination Address	**::/0**
Service Port	*** (* All ports)**
Protocol	**TCP**
Protocol Profile (Client)	Select the TCP Loose Fast L4 profile you created (**tcp-loose-fastl4** in our example).
VLAN and Tunnel Traffic	Select **Enabled On**, and then select only the **Internal** VLAN you created.
Source Address Translation	**SNAT**
SNAT Pool	Select the SNAT pool you created (**outbound-snatpool** in our example).

Configuring the BIG-IP system for outbound ICMP and UDP

To *pass ICMP through the BIG-IP* (so IPv6 will work and so intranet users can ping Internet resources) you need to create virtual servers that support the protocol ***All Protocols**. The simplest approach is to let those handle UDP as well, and block other protocols using the AFM network firewall policy.

Use the following table for guidance on configuring the BIG-IP system for outbound ICMP traffic.

Fast L4 Profiles (*Navigate to Local Traffic > Profiles > Protocol > FastL4*)	
Name	Type a unique name. In our example, we use **udp-outbound-fastl4**
Idle Timeout	Leave **Specify** selected, and then in the **Seconds** box, type **15**
Virtual Servers (*Navigate to Local Traffic > Virtual Servers*)	
IPv4 ICMP Outbound virtual server	
Name	**udp-icmp-outbound-4-vs**
Type	**Forwarding (IP)**
Destination Address	**0.0.0.0/0**
Service Port	*** (* All ports)**
Protocol	*** All Protocols**
Protocol Profile (Client)	Select the Fast L4 profile you created (**udp-outbound-fastl4** in our example).
VLAN and Tunnel Traffic	Select **Enabled On**, and then select only the **Internal** VLAN you created.
Source Address Translation	**SNAT**
SNAT Pool	Select the SNAT pool you created (**outbound-snatpool** in our example).
IPv6 FTP Gateway virtual server	
Name	**udp-icmp-outbound-6-vs**
Type	**Forwarding (IP)**
Destination Address	**::/0**
Service Port	*** (* All ports)**
Protocol	**TCP**
Protocol Profile (Client)	Select the Fast L4 profile you created (**udp-outbound-fastl4** in our example).
VLAN and Tunnel Traffic	Select **Enabled On**, and then select only the **Internal** VLAN you created.
Source Address Translation	**SNAT**
SNAT Pool	Select the SNAT pool you created (**outbound-snatpool** in our example).

Activating BIG-IP AFM Firewall Mode

When AFM is in *Firewall mode*, no virtual server or self IP processes traffic unless a sufficiently-permissive network firewall policy is enforced on it. Use the following table for guidance on activating Firewall mode. When you activate Firewall mode, remind all BIG-IP administrators to consider the presence or absence of firewall policies when they troubleshoot problems with BIG-IP virtual servers.

Default Firewall Action (*Navigate to Security > Options > Network Firewall*)	
Virtual Server & Self IP Contexts	Select **Drop**
Global Context	Select **Drop**, and then click **Update**.

Applying Fundamental Firewall Protection

The next task is to apply fundamental firewall protection on the BIG-IP AFM.

First, navigate to **Security > Network Firewall > Active Rules**. Click the word **Global** at the start of the first line of the list. A new page opens.

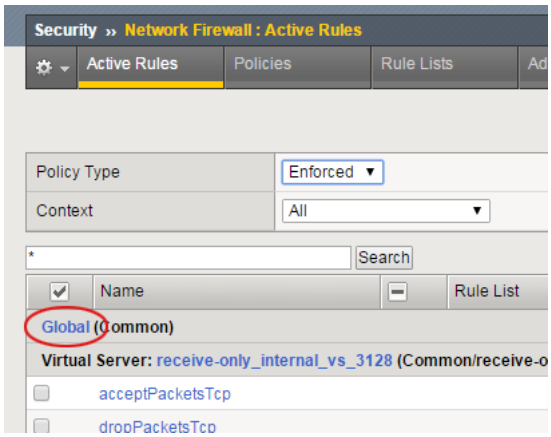

Figure 2: *Global link on the Active Rules page*

Use the following table for guidance on configuring the global firewall rules.

Global Firewall Rules (*Navigate to Security > Network Firewall > Active Rules > 'Global' link*)	
Network Firewall: Enforcement	Click **Enabled** From the **Policy** list that appears, select the global policy you created in *Creating the Network Firewall Policies on page 18* (**global** in our example).
Network Firewall: Staging	Leave Staging set to **Disabled**. Click **Update**.

The next task is to attach a firewall policy to the self IP addresses in the local-only traffic group. This needs to be performed on each BIG-IP device (big-s1 and big-s2) in our example. Then, on the primary device, attach the same policy to the self IPs in traffic-group-1 (floating), and use config sync to update the other device.

You may need to briefly enforce a different policy such as /Common/accept-all on one self IP while you configure a management-port firewall policy. See *Protecting the BIG-IP Management Port with AFM on page 46*.

Self IPs (*Navigate to Network > Self IPs*)	
Navigation help	Click Network > Self IPs. Click one of the self IPs in the traffic-group-local-only (use the Traffic Group column for reference). Once on the self IP properties page, click the Security tab.
Network Firewall: Enforcement	Select **Enabled**. From the **Policy** list that appears, select the self IP policy you created in *Creating the Network Firewall Policies on page 18* (**self IP** in our example).
Network Firewall: Staging	Leave Staging set to **Disabled**. Click **Update**.
	Repeat this guidance for each of the self IPs in traffic-group-local-only. Then repeat all on the second BIG-IP system. Finally, repeat this guidance on the primary device to the self IPs in traffic-group1, and then use configsync to update the other device. See https://support.f5.com/kb/en-us/products/big-ip_ltm/manuals/product/bigip-device-service-clustering-admin-11-6-0/5.html#unique_9477342 for specific information on syncing the BIG-IP configuration.

Next, on the Security tab of your DHCP relay virtual server(s), attach the firewall policy as shown in the following table. After testing and log analysis you can move the policy from staging to enforcement.

DHCP virtual server (*Navigate to Local Traffic > Virtual Servers > your-DHCP-relay-virtual-server > Security > Policies*)	
Navigation help	Navigate to **Local Traffic > Virtual Servers**. From the Virtual Server list, click the DHCP relay virtual server you created in *Configuring a DHCPv6 Relay on page 10* (or similar for IPv4 DHCP). On the Virtual Server Properties page, from the Menu bar, click **Security > Policies**.
Network Firewall: Enforcement	Select **Enabled**. From the **Policy** list that appears, select the Accept All policy you created in *Creating the Network Firewall Policies on page 18* (**accept-all** in our example).
Network Firewall: Staging	Select **Enabled**. From the **Policy** list that appears, select the DHCP relay policy you created in *Creating the Network Firewall Policies on page 18* (**ok-dhcp)** in our example).
IP Intelligence	Leave IP Intelligence set to **Disabled**.
DoS Protection Profile	Leave DoS Protection Profile set to **Disabled**.
Log Profile	Select **Enabled**. From the **Available** list, select the Log profile you created in *Configuring AFM High-Speed Logging on page 23* (**afm-log-pfl** in our example).
	If you also created an IPv4 DHCP relay virtual server, repeat this guidance for the IPv4 virtual server.

Applying a global IP Intelligence policy

The next task is to apply a global IP Intelligence policy to the configuration.

Global IP Intelligence Policy (*Navigate to Security > Network Firewall > IP Intelligence > Policies*)	
IP Intelligence Policy	From the list, select the IP Intelligence policy you created in *Creating an IP Intelligence policy on page 22* (**global-IPI-policy** in our example) Click **Update**.

Securing Outbound Traffic

This section provides guidance on securing outbound traffic through the BIG-IP system. In this example, we show how to stage the initial policies. After you generate some test traffic and make any adjustments suggested by log analysis you can move your policies from staging to enforcement.

Virtual Server (*Navigate to Local Traffic > Virtual Servers > your-outbound-virtual-servers > Security > Policies*)	
Navigation help	Click **Local Traffic > Virtual Server**. Click one of the virtual servers you created for outbound traffic. On the Menu bar, click **Security > Policies**. In our example, the outbound virtual servers are: ftp-outbound-4-vs, ftp-outbound-6-vs, tcp-outbound-4-vs, tcp-outbound-6-vs, udp-icmp-outbound-4-vs, and udp-icmp-outbound-6-vs.
Network Firewall: Enforcement	Select **Enabled**. From the **Policy** list that appears, select the Accept All policy you created in *Creating the Network Firewall Policies on page 18* (**accept-all** in our example).
Network Firewall: Staging	Select **Enabled**. From the **Policy** list that appears, select the outbound filter policy you created in *Creating the Network Firewall Policies on page 18* (**outbound** in our example).
IP Intelligence	Leave IP Intelligence set to **Disabled**.
DoS Protection Profile	Leave DoS Protection Profile set to **Disabled**.
Log Profile	Select **Enabled**. From the **Available** list, select the Log profile you created in *Configuring AFM High-Speed Logging on page 23* (**afm-log-pfl** in our example).
	Repeat this guidance for each of the outbound virtual servers.

Don't forget to set the BIG-IP floating self IP addresses fdf5:f:5:c245::1 and 10.1.245.1 (VLAN "internal") as the default gateway addresses on the data center routers (or L3 switches).

Additional information: Remote AFM Logging

You must provide a remote log server to accept log messages from AFM. The BIG-IP system supports several log formats. Our example uses the most common—syslog.

Most organizations capture detailed logs to support alerting, event correlation, incident management, security forensics, policy debugging, and so forth. AFM can emit many log messages very quickly. AFM uses the *BIG-IP High Speed Logging (HSL)* feature. In TMOS v11 High Speed Logging messages from AFM can only leave the BIG-IP system on traffic-handling VLAN's (see *SOL14459*). However, TMOS v12 added the ability to send High Speed Logging messages out the management port, though at a serious performance cost. You must create a Log Destination of type "Management Port" then enter the IP address, port, and protocol of the destination log server. After you create the Log Destination you can reference it in a Log Publisher.

(In default of a suitable logserver, you can simply disable logging most of the time but use the BIG-IP device's internal log storage to capture short bursts of log messages while testing new policies or troubleshooting network problems. That is not a recommended posture.)

Consider connecting your BIG-IP devices to a special logging VLAN/subnet to avoid mixing application and logging traffic. Mixing traffic can lead to congestion and delays. Another reason to segregate logging traffic is to bypass router and EW firewall hops in your intranet: if you send log messages directly onto the subnet where your log/SIEM server(s) live you may gain a lot of performance.

Additional Information: Internet Access from the Intranet

All traffic from the intranet to the Internet (NS) requires address translation as well as security filtering. You SNAT the private (intranet) source addresses of outbound connections using a pool of public addresses. To make your firewall rules more efficient use SNAT addresses you can summarize in CIDR blocks, as shown in our example. SNAT also helps hide details of the intranet from malicious users.

To protect intranet devices and users against various threats and to frustrate APT's trying to "phone home" you should prevent outbound connections to addresses blacklisted by the IP Intelligence service. With TMOS v11.6 this requires an iRule like **afm-no-evil-dst** which we define in the example. To keep APT's from sending spam it is customary to forbid connections to TCP port 25 (SMTP) except from the organization's mailserver.

More elaborate security policies may be desirable. You can add rules to keep intranet devices from accessing Internet services improperly. For example, you might forbid outbound connections to TCP port 179 (BGP) to prevent APT's harassing remote organizations.

Most application protocols use TCP and UDP in a straightforward manner but some like FTP require extra support (an ALG) to pass through the BIG-IP. You set up "more specific" virtual servers to process such protocols. For example, the basic configuration includes specific virtual servers to support outbound FTP connections as explained in *SOL8021: Configuring the BIG-IP LTM system to allow outbound FTP sessions*. Deploying other BIG-IP ALG's is basically similar.

An organizational "web proxy" is sort of "the ultimate ALG." A typical web proxy—such as *BIG-IP Secure Web Gateway (SWG)*—does a lot of security work. It enforces the organization's Acceptable Use Policy (AUP), guards against malicious web content, scans downloads for malware, and much more besides. Though specific details are beyond the scope of this document, to integrate your web proxy with your BIG-IP data center firewall you will create wildcard virtual servers on TCP ports 80 and 443 (at least) to collect traffic "transparently" from users and send it to the proxy. You will also create some source-specific virtual servers to pass the web-proxy's outbound traffic to the Internet.

You may also integrate Data Loss Prevention (DLP) or similar services with your BIG-IP data center firewall. The BIG-IP system supports ICAP for simple use cases, and much more comprehensive IDS/IPS/NGFW integrations when desired.

Securing an Email Gateway

A typical organization has a spam-filtering email gateway to handle SMTP inbound and outbound. The model network includes a mail server trusted to operate in the intranet rather than confined to a DMZ. It is very secure and uses DNS real-time blacklists to reject messages which come from undesirable senders or include dangerous links. Accordingly, the mail server needs to see the actual source IP address of every incoming SMTP connection. It is dual-stacked to support both IPv6 and IPv4 addresses conveniently.

Of course F5 customers use a variety of email solutions. Microsoft and F5 are technology partners and support a powerful integration of Microsoft Exchange Server with the BIG-IP system. See the *F5 Microsoft Exchange Deployment Guide* for information and details on the iApp template for Microsoft Exchange Client Access Servers.

Though the example in this guide is simple, the F5 document *Deploying the BIG-IP System with SMTP Servers* offers guidance for a variety of usage scenarios. For instance, it explains how to offload TLS processing for SMTP from mail servers to the BIG-IP system by attaching an SMTPS Profile and a Client SSL Profile to each SMTP virtual server.

Create the following objects to support inbound SMTP traffic (the basic AFM configuration handles outbound SMTP in the Rule List **filter-outbound**). This configuration requires two pools in order to pass both IPv6 and IPv4 remote addresses through to the mail server.

Network Firewall Policies (*Navigate to Security > Network Firewall > Policies*)	
Description of policy	This network firewall policy is for inbound email. The Destination address "any" covers IPv6 and IPv4, and lets you adjust virtual server addresses when, for example, you change ISP's and get a new public IP block.
Policy Name	**inbound-mail** Click **Finished** and then from the Policies table, click the Policy you just created.
Rules	Click the **Add** button and add the following rule.
	Name **ok-smtp**
	Protocol **TCP**
	Source > Address/Region **Any**
	Source > Port **Any**
	Source > VLAN/Tunnel **external**
	Destination > Address/Region **Any**
	Destination > Port **smtp-mta**
	Action **Accept** Click **Finished**.
Monitors (*Navigate to Local Traffic > Monitors*)	
Name	**mon-smtp**
Type	**SMTP**
Interval	**30**
Timeout	**91**
Domain	Type the appropriate domain. In our example, we use **example.net**.
Pools (*Navigate to Local Traffic > Pools*)	
IPv4 mail server pool	
Name	Type a unique name. In our example, we use **mailserver-4-pool**
Health Monitor	**mon-smtp**
New Members	Node Name **mailserver-4**
	Node Address **10.1.7.25**
	Service Port **25 (SMTP)**
IPv6 mail server pool	
Name	Type a unique name. In our example, we use **mailserver-6-pool**
Health Monitor	**mon-smtp**
New Members	Node Name **mailserver-6**
	Node Address **fdf5:f:5:c007::25**
	Service Port **25 (SMTP)**

Virtual Servers (*Navigate to Local Traffic > Virtual Servers*)	
IPv4 mail server virtual server	
Name	**inbox-4-vs**
Type	**Standard**
Destination Address	**192.0.2.25**
Service Port	**25 (SMTP)**
VLAN and Tunnel Traffic	Select **Enabled On**, and then select only the **external** VLAN you created.
Default Pool	Select the IPv4 mail server pool you created (**mailserver-4-pool** in our example).
	Click **Finished** and then from the Virtual Servers table, click the virtual server you just created. On the Menu Bar, click **Security > Policies**.
Network Firewall: Enforcement	Select **Enabled**. From the **Policy** list that appears, select the Accept All policy you created in *Creating the Network Firewall Policies on page 18* (**accept-all** in our example).
Network Firewall: Staging	Select **Enabled**. From the **Policy** list that appears, select the inbound mail policy you just created (**inbound-mail** in our example).
IP Intelligence	Leave IP Intelligence set to **Disabled**.
DoS Protection Profile	Leave DoS Protection Profile set to **Disabled**.
Log Profile	Select **Enabled**. From the **Available** list, select the Log profile you created in *Configuring AFM High-Speed Logging on page 23* (**afm-log-pfl** in our example).
IPv6 mail server virtual server	
Name	**inbox-6-vs**
Type	**Standard**
Destination Address	**2001:db8:16d:2::25**
Service Port	**25 (SMTP)**
VLAN and Tunnel Traffic	Select **Enabled On**, and then select only the **external** VLAN you created.
Default Pool	Select the IPv4 mail server pool you created (**mailserver-6-pool** in our example).
	Click **Finished** and then from the Virtual Servers table, click the virtual server you just created. On the Menu Bar, click **Security > Policies**.
Network Firewall: Enforcement	Select **Enabled**. From the **Policy** list that appears, select the Accept All policy you created in *Creating the Network Firewall Policies on page 18* (**accept-all** in our example).
Network Firewall: Staging	Select **Enabled**. From the **Policy** list that appears, select the inbound mail policy you just created (**inbound-mail** in our example).
IP Intelligence	Leave IP Intelligence set to **Disabled**.
DoS Protection Profile	Leave DoS Protection Profile set to **Disabled**.
Log Profile	Select **Enabled**. From the **Available** list, select the Log profile you created in *Configuring AFM High-Speed Logging on page 23* (**afm-log-pfl** in our example).

Securing a Multi-Tier Web Application with Presentation Servers in a DMZ

DMZ's hold servers which mediate between high- and low-trust environments. A DMZ typically has two trust boundaries (NS and EW), policed by firewall policy but also enforced by the LTM configuration. Among other things, the EW trust boundary should keep an adversary who compromises a server in the DMZ from extending his intrusion into the intranet.

This example shows a multi-tier web application (App1) with presentation servers in the DMZ (DMZ1) and additional (e.g., business-logic/database) processing in the data center. Only two kinds of traffic may enter DMZ1: web traffic and system-administration traffic. Most of the web traffic comes from the Internet but some comes from the intranet. Sys-admin traffic comes only from the management network in the intranet. Two kinds of traffic exit DMZ1: logging and limited admin (e.g., DHCP) traffic plus application traffic only to the application backend service (this might be a BIG-IP virtual server with a pool of application servers behind it). DMZ1 machines may initiate connections to approved data-center resources but nothing else.

In this section, the first task is to configure IP forwarding virtual servers with a network firewall policy. They carry and control DMZ1 EW traffic for administration, logging, and backend access. The next step is to deploy application-specific virtual servers, which admit NS and EW web traffic to DMZ1. The web virtual servers let you use an AFM HTTP Protocol Security Profile to protect the application and support server load balancing.

In this example App1 presentation servers talk SOAP or REST over HTTPS to their backend. To support another protocol, you could change the port in the relevant firewall rule (Rule "ok webback1", part of rule-list "permit dmz1-to dc") from https (443) to something like ms-sql (1433) for MS SQL Server 2012.

Use the following table for guidance on configuring the objects on the BIG-IP system. For instructions on configuring individual objects, see the Help tab or the product documentation.

VLANs (*Navigate to Network > VLANs*)	
Notes	Create this VLAN on each device. It carries the first DMZ subnet.
Name	Type a unique name. In our example, we use **dmz1.** Configure tagging as appropriate for your deployment.
Self IPs (*Navigate to Network > Self IPs*)	
On the big-s1 device, create the following Self IPs in the /Common partition	
Name	Type a unique name. In our example, we use **self-dmz1-s1-4**
IP Address	**172.30.1.2**
Netmask	**255.255.255.0**
VLAN	Select the VLAN you created in this section (**dmz1** in our example)
Port Lockdown	**Allow Default**
Traffic Group	**traffic-group-local-only (non-floating)**
Name	Type a unique name. In our example, we use **self-dmz1-s1-6**
IP Address	**fdf5:f:5:d001::2**
Netmask	**ffff:ffff:ffff:ffff::**
VLAN	Select the VLAN you created in this section (**dmz1** in our example)
Port Lockdown	**Allow Default**
Traffic Group	**traffic-group-local-only (non-floating)**
On the big-s2 device, create the following Self IPs in the /Common partition	
Name	Type a unique name. In our example, we use **self-dmz1-s2-4**
IP Address	**172.30.1.3**
Netmask	**255.255.255.0**
VLAN	Select the VLAN you created in this section (**dmz1** in our example)
Port Lockdown	**Allow Default**
Traffic Group	**traffic-group-local-only (non-floating)**
Name	Type a unique name. In our example, we use **self-dmz1-s2-6**
IP Address	**fdf5:f:5:d001::3**
Netmask	**ffff:ffff:ffff:ffff::**
VLAN	Select the VLAN you created in this section (**dmz1** in our example)
Port Lockdown	**Allow Default**
Traffic Group	**traffic-group-local-only (non-floating)**

On the both devices, create the following floating Self IPs in the /Common partition	
Name	Type a unique name. In our example, we use **self-dmz1-float-4**
IP Address	**172.30.1.1**
Netmask	**255.255.255.0**
VLAN	Select the VLAN you created in this section (**dmz1** in our example)
Port Lockdown	**Allow Default**
Traffic Group	**traffic-group-1 (floating)**
Name	Type a unique name. In our example, we use **self-dmz1-float-6**
IP Address	**fdf5:f:5:d001::1**
Netmask	**ffff:ffff:ffff:ffff::**
VLAN	Select the VLAN you created in this section (**dmz1** in our example)
Port Lockdown	**Allow Default**
Traffic Group	**traffic-group-1 (floating)**

Configuring DMZ1 Administrative and Backend Traffic

Note: You may download an iApp to perform much of the following configuration from: *https://devcentral.f5.com/codeshare/data-center-firewall-quick-start-iapp-template*

On each BIG-IP device, issue one of the following TMSH commands, depending on your configuration.

In the model network architecture, servers learn DNS settings from DHCPv6, so the command is:

```
create /net router-advertisement ra-dmz1 vlan dmz1 enabled prefixes
 add { /Common/pfx-dmz1 { prefix fdf5:f:5:d001:: prefix-length 64 }
 router } managed other-config
```

If you set up a network without DHCPv6, substitute this command:

```
create /net router-advertisement ra-dmz1 vlan dmz1 enabled prefixes
 add { /Common/pfx-dmz1 { prefix fdf5:f:5:d001:: prefix-length 64 }
 router }
```

Save your changes using the following command:

```
save /sys config
```

Configuring the BIG-IP objects

Use the following table for guidance on configuring the BIG-IP system.

DHCP virtual server (*Navigate to Local Traffic > Virtual Servers > your-DHCP-relay-virtual-server*)		
Navigation help	Navigate to **Local Traffic > Virtual Servers**. From the Virtual Server list, click the DHCP relay virtual server you created in *Configuring a DHCPv6 Relay on page 10* (or similar for IPv4 DHCP).	
VLAN and Tunnel Traffic	Add the DMZ VLAN you created (**dmz1** in our example) to the **Selected** list. There should be only two VLANs in the Selected list, the DMZ VLAN and the Internal VLAN. If applicable, repeat for the other DHCP relay virtual server.	
Address Lists (*Navigate to Security > Network Firewall > Address Lists*)		
Name	**all-dmz-nets**	**Notes (not a part of the configuration)**
Description	(example)	
Addresses/Regions	Type or copy and paste the following addresses into the Add new entry box: **172.30.0.0/16** | **fdf5:f:5:d000::/54**	All DMZ subnets (part of intranet-nets).

Name	**dmz1-nets**	**Notes (not a part of the configuration)**
Description	(example)	DMZ1
Addresses/Regions	Type or copy and paste the following addresses into the Add new entry box: **172.30.1.0/24** | **fdf5:f:5:d001::/64**	

Rule Lists (*Navigate to Security > Network Firewall > Rule Lists*)		
permit-mgmt-to-dmz Rule list		
Rule List Name	**permit-mgmt-to-dmz** Click **Finished** and then from the Rule Lists table, click the Rule List you just created.	
Rules	Click the **Add** button and add the following rules.	
	Name	**ok-tcp-mgmt**
	Protocol	**TCP**
	Source > Address/Region	Add **net-mgmt-nets**
	Source > Port	**any**
	Source > VLAN/Tunnel	**any**
	Destination > Address/Region	Add **all-dmz-nets**
	Destination > Port	**tcp-EW-generic**
	Action	**Accept** Click **Repeat** and then add the following rule.
	Name	**ok-udp-mgmt**
	Protocol	**UDP**
	Source > Address/Region	Add **net-mgmt-nets**
	Source > Port	**any**
	Source > VLAN/Tunnel	**any**
	Destination > Address/Region	Add **all-dmz-nets**
	Destination > Port	Add **snmp-mgr** and **udp-dhcp**
	Action	**Accept** Click **Finished**

Use the following table for guidance on configuring the following Rule lists.

Rule List Name	Rule Name	Protocol	Source address	Src. Port	Src. VLAN	Dst. Address	Dst. Port	Action
permit-dmz1-to-dc	ok-basic-udp	UDP	dmz1-nets	Any	dmz1	all-dc-nets	udp-EW-basic	Accept
	ok-basic-tcp	TCP	dmz1-nets	Any	dmz1	all-dc-nets	tcp-EW-basic	Accept
	ok-webback1	TCP	dmz1-nets	Any	dmz1	fdf5:f:5:c001:31	HTTPS	Accept
access-to-webapps	ok-intra	TCP	intranet-nets	Any	internal	Any	Any	Accept
	no-ITAR [1]	TCP	ITAR-countries	Any	Any	Any	Any	Drop
	ok-others	TCP	Any	Any	Any	Any	snmp-mgr	Accept

[1] For the no-ITAR rule list, from the Logging row, select Enabled

Network Firewall Policies (*Navigate to Security > Network Firewall > Policies*)		
Description of policy	This network firewall policy is for the Management network to any DMZ.	
Policy Name	**mgmt-to-dmz** Click **Finished** and then from the Policies table, click the Policy you just created.	
Rules	Click the **Add** button and add the following rule.	
	Name	**mgmt2dmz**
	Type	**Rule List**
	Rule List	**permit-mgmt-to-dmz**
Description of policy	This network firewall policy is for DMZ1 to the data center.	
Policy Name	**dmz1-to-dc** Click **Finished** and then from the Policies table, click the Policy you just created.	
Rules	Click the **Add** button and add the following rule.	
	Name	**dmz1-dc**
	Type	**Rule List**
	Rule List	**permit-dmz1-to-dc**

Virtual Servers (*Navigate to Local Traffic > Virtual Servers*)	
Note the first two objects support management traffic into all DMZ's. The latter two are DMZ1-specific. Essentially, we trust the management network a lot and various DMZ's very little.	
IPv4 management to DMZ virtual server	
Name	**mgmt-to-dmz-4-vs**
Type	**Forwarding (IP)**
Destination Address	**172.30.0.0/16**
Service Port	*** (*All ports)**
Protocol	*** All Protocols**
Protocol Profile (Client)	Select the TCP Loose Fast L4 profile you created (**tcp-loose-fastl4** in our example).
VLAN and Tunnel Traffic	Select **Enabled On**, and then select only the **internal** VLAN you created.
Source Address Translation	**None**
	Click **Finished** and then from the Virtual Servers table, click the virtual server you just created. Next, we stage the initial policies. After you generate some test traffic and make any adjustments suggested by log analysis you can move your policies from staging to enforcement. On the Menu Bar, click **Security > Policies**.
Network Firewall: Enforcement	Select **Enabled**. From the **Policy** list that appears, select the Accept All policy you created in *Creating the Network Firewall Policies on page 18* (**accept-all** in our example).
Network Firewall: Staging	Select **Enabled**. From the **Policy** list that appears, select the management to DMZ policy you just created (**mgmt-to-dmz** in our example).
IP Intelligence	Leave IP Intelligence set to **Disabled**.
DoS Protection Profile	Leave DoS Protection Profile set to **Disabled**.
Log Profile	Select **Enabled**. From the **Available** list, select the Log profile you created in *Configuring AFM High-Speed Logging on page 23* (**afm-log-pfl** in our example).
IPv6 mail server virtual server	
Name	**mgmt-to-dmz-6-vs**
Type	**Forwarding (IP)**
Destination Address	**fdf5:f:5:d000::/54**
Service Port	*** (*All ports)**
Protocol	*** All Protocols**
Protocol Profile (Client)	Select the TCP Loose Fast L4 profile you created (**tcp-loose-fastl4** in our example).
VLAN and Tunnel Traffic	Select **Enabled On**, and then select only the **internal** VLAN you created.
Source Address Translation	**None**
	Click **Finished** and then from the Virtual Servers table, click the virtual server you just created. On the Menu Bar, click **Security > Policies**.
Network Firewall: Enforcement	Select **Enabled**. From the **Policy** list that appears, select the Accept All policy you created in *Creating the Network Firewall Policies on page 18* (**accept-all** in our example).
Network Firewall: Staging	Select **Enabled**. From the **Policy** list that appears, select the management to DMZ policy you just created (**mgmt-to-dmz** in our example).
IP Intelligence	Leave IP Intelligence set to **Disabled**.
DoS Protection Profile	Leave DoS Protection Profile set to **Disabled**.
Log Profile	Select **Enabled**. From the **Available** list, select the Log profile you created in *Configuring AFM High-Speed Logging on page 23* (**afm-log-pfl** in our example).
IPv6 mail server virtual server	
Name	**dmz1-to-intraA-4-vs**
Type	**Forwarding (IP)**
Destination Address	**10.0.0.0/8**
Service Port	*** (*All ports)**
Protocol	*** All Protocols**
Protocol Profile (Client)	Select the TCP Loose Fast L4 profile you created (**tcp-loose-fastl4** in our example).
VLAN and Tunnel Traffic	Select **Enabled On**, and then select only the DMZ VLAN you created (**dmz1** in our example).
Source Address Translation	**None**
	Click **Finished** and then from the Virtual Servers table, click the virtual server you just created. On the Menu Bar, click **Security > Policies**.

Network Firewall: Enforcement	Select **Enabled**. From the **Policy** list that appears, select the Accept All policy you created in *Creating the Network Firewall Policies on page 18* (**accept-all** in our example).
Network Firewall: Staging	Select **Enabled**. From the **Policy** list that appears, select the management to DMZ policy you just created (**dmz1-to-dc** in our example).
IP Intelligence	Leave IP Intelligence set to **Disabled**.
DoS Protection Profile	Leave DoS Protection Profile set to **Disabled**.
Log Profile	Select **Enabled**. From the **Available** list, select the Log profile you created in *Configuring AFM High-Speed Logging on page 23* (**afm-log-pfl** in our example).
IPv6 mail server virtual server	
Name	**dmz1-to-intraC-4-vs**
Type	**Forwarding (IP)**
Destination Address	**192.168.0.0/16**
Service Port	*** (*All ports)**
Protocol	*** All Protocols**
Protocol Profile (Client)	Select the TCP Loose Fast L4 profile you created (**tcp-loose-fastl4** in our example).
VLAN and Tunnel Traffic	Select **Enabled On**, and then select only the DMZ VLAN you created (**dmz1** in our example).
Source Address Translation	**None**
	Click **Finished** and then from the Virtual Servers table, click the virtual server you just created. On the Menu Bar, click **Security > Policies**.
Network Firewall: Enforcement	Select **Enabled**. From the **Policy** list that appears, select the Accept All policy you created in *Creating the Network Firewall Policies on page 18* (**accept-all** in our example).
Network Firewall: Staging	Select **Enabled**. From the **Policy** list that appears, select the management to DMZ policy you just created (**dmz1-to-dc** in our example).
IP Intelligence	Leave IP Intelligence set to **Disabled**.
DoS Protection Profile	Leave DoS Protection Profile set to **Disabled**.
Log Profile	Select **Enabled**. From the **Available** list, select the Log profile you created in *Configuring AFM High-Speed Logging on page 23* (**afm-log-pfl** in our example).
IPv6 mail server virtual server	
Name	**dmz1-to-intra-6-vs**
Type	**Forwarding (IP)**
Destination Address	**fdf5:f:5::/48**
Service Port	*** (*All ports)**
Protocol	*** All Protocols**
Protocol Profile (Client)	Select the TCP Loose Fast L4 profile you created (**tcp-loose-fastl4** in our example).
VLAN and Tunnel Traffic	Select **Enabled On**, and then select only the DMZ VLAN you created (**dmz1** in our example).
Source Address Translation	**None**
	Click **Finished** and then from the Virtual Servers table, click the virtual server you just created. On the Menu Bar, click **Security > Policies**.
Network Firewall: Enforcement	Select **Enabled**. From the **Policy** list that appears, select the Accept All policy you created in *Creating the Network Firewall Policies on page 18* (**accept-all** in our example).
Network Firewall: Staging	Select **Enabled**. From the **Policy** list that appears, select the management to DMZ policy you just created (**dmz1-to-dc** in our example).
IP Intelligence	Leave IP Intelligence set to **Disabled**.
DoS Protection Profile	Leave DoS Protection Profile set to **Disabled**.
Log Profile	Select **Enabled**. From the **Available** list, select the Log profile you created in *Configuring AFM High-Speed Logging on page 23* (**afm-log-pfl** in our example).

Creating a Protocol Security Profile

Use the following table for guidance on configuring the Protocol Security profile.

Protocol Security Profiles (*Navigate to Security > Protocol Security > Security Profiles > HTTP*)		
Name	Type a unique name. In our example, we use **afm-http-secy-pfl**	
Parent Profile	**http_security**	
HTTP Protocol Checks Tab	**HTTP Protocol Checks**	Check the **High ASCII characters in headers** box. Leave the other settings at the default.
Blocking Page Tab	**Response Type**	Select **Custom Response** from the list (you may have to click the Custom box on the right).
	Response Headers	**HTTP/1.1 200 OK** **Content-Type: text/html; charset=ISO-8859-1** **Cache-Control: no-cache** **Pragma: no-cache** **Connection: close**
	Response Body	Leave the default Response Body.

Configuring the BIG-IP system to send web application traffic into the DMZ

To admit web traffic to the App1 presentation servers in DMZ1, the easiest way to configure the BIG-IP system is to use the iApp template for HTTP Applications (f5.http) as explained in the *F5 Deployment Guide Deploying the BIG-IP System with HTTP Applications*. The first configuration scenario in that Guide—using BIG-IP as a reverse (or inbound) proxy—describes the goal here.

The HTTP iApp template (and the manual configuration described in the deployment guide) supports BIG-IP AFM, including an option called **F5's recommended AFM configuration**. That is not actually the AFM configuration recommended in this Deployment Guide, so you should create an AFM network firewall policy manually using the following guidance. You can then select your policy in the iApp template.

This example assumes you have an RSA or ECC private key and corresponding public-key TLS certificate signed by a well-known CA (Certificate Authority) for App1. Before creating any other objects, import the TLS certificate and key as explained in *SOL14620: Managing SSL certificates for BIG-IP systems using the Configuration utility*. Name the certificate and key **20*YY*-app1.example.net** where 20YY is the year the certificate expires. Also import any intermediate-CA certificate(s) (in a bundle if more than one).

The web presentation servers (webfront1, webfront2 in our example) in DMZ1 can use TLS certificates signed by your organization's private CA. Import the private CA's root certificate to the BIG-IP system. Name it something like **2025-Private-CA-Root-example.net**.

Use the following table for guidance on configuring the BIG-IP system for web application traffic into the DMZ. After creating these objects, you will run the iApp template.

Network Firewall Policies (*Navigate to Security > Network Firewall > Policies*)		
Policy Name	**app1-access** Click **Finished** and then from the Policies table, click the Policy you just created.	
Rules	Click the **Add** button and add the following rule.	
	Name	**access-to-app1**
	Type	**Rule List**
	Rule List	**access-to-webapps**
Nodes (*Navigate to Local Traffic > Nodes*)		
Name	Type a unique name. We use **webfront1**.	
Address	Type the address. We use **fdf5:f:5:d001::41**.	
	Click **Repeat** and then add the following node	
Name	Type a unique name. We use **webfront2**.	
Address	Type the address. We use **fdf5:f:5:d001::42**.	
	Click **Finished**	

iRules (*Navigate to Local Traffic > iRules*)

Important: HSTS (HTTP Strict Transport Security; rfc6797) is an important security measure for TLS-protected (HTTPS) websites. With BIG-IP TMOS version 11.x an iRule is needed to implement HSTS. BIG-IP TMOS v12.x has support for HSTS built in (though the iRule will still work—use it when you want to defer to HSTS headers from the application web server). When using TMOS v12+ you may skip the creation of the HSTS iRule and create a custom HTTP Profile with HSTS enabled instead. Attach either the iRule or the custom HTTP Profile (but not both!) to your HTTPS virtual server.

Name — Give the new iRule unique name. We use **ir-add-HSTS-hdr**.

Definition — Copy and paste the following iRule, omitting the line numbers:

```
1  # If site is TLS-protected, tell browsers to lock in
2  # HTTPS to it for 31 days per rfc6797 (one warning per
3  # connection is enough).  A web server may supply its
4  # own HSTS header to override this iRule.
5
6  when CLIENT_ACCEPTED priority 605 {
7     set add_HSTS_hdr [PROFILE::exists clientssl]
8  } ; #CLIENT_ACCEPTED 605
9
10 when HTTP_RESPONSE priority 605 {
11    if {$add_HSTS_hdr &&
12       ![HTTP::header exists "Strict-Transport-Security"]} {
13       HTTP::header insert "Strict-Transport-Security" \
14          "max-age=2678400; includeSubDomains"
15    }
16    set add_HSTS_hdr false
17 } ; #HTTP_RESPONSE 605
```

DNS Records (*Your DNS implementation)*

Create suitable DNS records in your environment. In our example, the domain name for this service is app1.example.net, so we configure the following records. Add appropriate reverse entries too, like 15.245.1.10.in-addr.arpa PTR app1.example.net.

DNS View	Name	Type	Value
public	***app1.example.net***	AAAA	2001:db8:16d:2::15
public	***app1.example.net***	A	192.0.2.15
internal	***app1.example.net***	AAAA	fdf5:f:5:c245::15
internal	***app1.example.net***	A	10.1.245.15

Profiles (*Navigate to Local Traffic > Profiles)*

Client SSL *(Profiles > SSL)*	Name	Type a unique name. We use **clientssl-app1**
	Parent Profile	**clientssl**
	Certificate	Select the Certificate you imported onto the BIG-IP system for this deployment. We select **2025-app1.example.net.**
	Key	Select the associated key. We select **2025-app1.example.net.**
	Chain	If your implementation requires an intermediate cert from a CA, select it from the list
	Passphrase	Type the passphrase if the key is encrypted
	If the web servers support multiple applications with TLS SNI, you can add additional certificates here	
Server SSL *(Profiles-->SSL)*	Name	Type a unique name. We use **serverssl-app1**
	Parent Profile	**serverssl**
	Server Certificate	**require**
	Authenticate Name	**app1.example.net**
	Trusted Certificate Authorities	Select the Trusted Certificate Authority you want to use. We select **2025-Private-CA-Root-example.net**
HTTP *(Profiles > Services)* Important: *Only create this profile if you are using BIG-IP v12.0 or later as described in the iRules section*	Name	Type a unique name (we use http-hsts-app1)
	HTTP Strict Transport Security	**Mode** — Check the box to enable Mode. This enables the HSTS settings **Maximum Age** — **2678400** **Include Subdomains** — Ensure the Include Subdomains box is checked (Enabled)

Running the HTTP iApp template

The next task is to run the HTTP iApp template. In this example, you run the iApp template four separate times to create the virtual servers for web traffic into DMZ1. To run the iApp template, click **iApps > Application Services > Create**. From the Template list, select **f5.http**. Use the following table for guidance on configuring the iApp template for each scenario.

Question	1st: Answer IPv6 NS	2nd: Answer IPv4 NS	3rd: Answer IPv6 EW	4th: Answer IPv4 EW
Name	**app1-public-6**	**app1-public-4**	**app1-intra-6**	**app1-intra-4**
Which configuration mode do you want to use?	Advanced – Configure advanced options	Advanced – Configure advanced options	Advanced – Configure advanced options	Advanced – Configure advanced options
Do you want to restrict client traffic to specific VLANs?	Yes, enable traffic only on the VLANs I specify	Yes, enable traffic only on the VLANs I specify	Yes, enable traffic only on the VLANs I specify	Yes, enable traffic only on the VLANs I specify
On which VLANs should traffic be enabled or disabled?	Selected: **/Common/external** (only!) Options: (move all others here)	Selected: **/Common/external** (only!) Options: (move all others here)	Selected: **/Common/internal** (only!) Options: (move all others here)	Selected: **/Common/internal** (only!) Options: (move all others here)
How have you configured routing on your web servers?	Servers have a route to clients through the BIG-IP system	Servers **do not** have a route to clients through the BIG-IP	Servers have a route to clients through the BIG-IP system	Servers **do not** have a route to clients through the BIG-IP
How many connections do you expect to each web server?		More than 64,000 concurrent connections		More than 64,000 concurrent connections
Create a new SNAT pool or use an existing one?		Create a new SNAT pool		Create a new SNAT pool
What are the IP addresses you want to use for the SNAT pool?		2001:db8:16d:2::15		fdf5:f:5:c245::15
How should the BIG-IP system handle SSL traffic?	Terminate SSL from clients, re encrypt to servers (SSL bridging)	Terminate SSL from clients, re encrypt to servers (SSL bridging)	Terminate SSL from clients, re encrypt to servers (SSL bridging)	Terminate SSL from clients, re encrypt to servers (SSL bridging)
Which Client SSL profile do you want to use?	/Common/clientssl app1	/Common/clientssl app1	/Common/clientssl app1	/Common/clientssl app1
Which Server SSL profile do you want to use?	/Common/serverssl app1	/Common/serverssl app1	/Common/serverssl app1	/Common/serverssl app1
Do you want to use BIG-IP AFM to protect your application?	/Common/accept-all [1]	/Common/accept-all [1]	/Common/accept-all [1]	/Common/accept-all [1]
How do you want to control access to your application from sources with a low reputation score?	Allow all sources regardless of reputation (because in this configuration, we have a global IP Intelligence policy)	Allow all sources regardless of reputation (because in this configuration, we have a global IP Intelligence policy)	Allow all sources regardless of reputation (because in this configuration, we have a global IP Intelligence policy)	Allow all sources regardless of reputation (because in this configuration, we have a global IP Intelligence policy)
Would you like to stage a policy for testing purposes?	/Common/app1 access	/Common/app1 access	/Common/app1 access	/Common/app1-access
Which logging profile would you like to use?	afm-log-pfl	afm-log-pfl	afm-log-pfl	afm-log-pfl
What IP address do you want to use for the virtual server?	2001:db8:16d:2::15	192.0.2.15	fdf5:f:5:c245::15	10.1.245.15
What FQDNs will clients use to access the servers?	app1.example.net	app1.example.net	app1.example.net	app1.example.net
Do you want to redirect inbound HTTP traffic to HTTPS?	Redirect HTTP to HTTPS	Redirect HTTP to HTTPS	Redirect HTTP to HTTPS	Redirect HTTP to HTTPS
Which HTTP profile do you want to use?	**If using BIG-IP v12.0 or later (and not using the HSTS iRule) only:**			
	Select the HTTP profile with HSTS you created.	Select the HTTP profile with HSTS you created.	Select the HTTP profile with HSTS you created.	Select the HTTP profile with HSTS you created.
Should the BIG-IP system insert the X-Forwarded-For header?	Do not insert X Forwarded For HTTP header	Insert X Forwarded For HTTP header [2]	Do not insert X Forwarded For HTTP header	Insert X Forwarded For HTTP header [2]
Do you want to create a new pool or use an existing one?	Create a new pool	Select the pool created by the first iApp, e.g.: **app1-public-6_pool**	Select the pool created by the first iApp, e.g.: **app1-public-6_pool**	Select the pool created by the first iApp, e.g.: **app1-public-6_pool**
Which load balancing method do you want to use?	Observed (member)			
Which web servers should be included in this pool?	/Common/webfront1 [3] /Common/webfront2 [3]			
How many seconds should Slow Ramp time last?	60			

[1] This option chooses which policy to enforce. After you test and adjust policy 'app1-access' in staging mode you come back here to enforce it.
[2] Needed with SNAT
[3] Leave **Port** set to **443** and **Connection Limit** set to **0**

Question	1st: Answer IPv6 NS	2nd: Answer IPv4 NS	3rd: Answer IPv6 EW	4th: Answer IPv4 EW
How many seconds should pass between health checks?	5			
Which Web Acceleration profile do you want to use for caching?		Select the profile created by the first iApp, e.g.: **app1-public-6_optimized-caching**		Select the profile created by the 3rd iApp, e.g.: **app1-intra-6_optimized-caching**
Which compression profile do you want to use?		Select the profile created by the first iApp, e.g.: **app1-public-6_wan-optimized-compression**		Select the profile created by the 3rd iApp, e.g.: **app1-intra-6_wan-optimized-compression**
How do you want to optimize client-side connections?		Select the profile created by the first iApp, e.g.: **app1-public-6_tcp-wan-optimized**	tcp-lan-optimized	Select the profile created by the 3rd iApp, e.g.: **app1-intra-6_tcp-wan-optimized**
Which OneConnect profile do you want to use?		Select the profile created by the first iApp, e.g.: **app1-public-6_oneconnect**		Select the profile created by the 3rd iApp, e.g.: **app1-intra-6_oneconnect**
How do you want to optimize server-side connections?		Select the profile created by the first iApp, e.g.: **app1-public-6_tcp-lan-optimized**		Select the profile created by the 3rd iApp, e.g.: **app1-intra-6_tcp-lan-optimized**
Do you want to add any custom iRules to this configuration?	ir-add-HSTS-hdr [4]	ir-add-HSTS-hdr [4]	ir-add-HSTS-hdr [4]	ir-add-HSTS-hdr [4]

[4] If some portion of website is not served over HTTPS, or if the web server supplies its own HSTS header, you should omit this iRule.

Attaching the protocol security profile to the configuration produced by the iApp template

The next task is to attach the protocol security profile you created to the virtual servers created by the iApp template. This requires first disabling the Strict Updates feature on each iApp Application Service. For any other changes you need to make to the configuration, use the iApp template **Reconfigure** option.

To disable Strict Updates, click **iApps > Application Services > app1-public-6** (or the name you gave the iApp). On the Menu bar, click **Properties**. In the **Strict Updates** row, clear the checkbox to disable Strict Updates (you may have to select Advanced from the Configuration menu first). Click **Update**. Repeat this process for each of the iApp application services you just created.

Navigate to **Local Traffic > Virtual Servers > app1-public-6_vs** (or name you gave the iApp followed by **_vs**). On the Menu bar, click **Security**. From the Protocol Security list, select Enabled. From the Profile list, select the Protocol Security profile you created in *Creating a Protocol Security Profile on page 37* (**afm-http-secy-pfl** in our example), and then click **Update**. Repeat for each of the virtual servers created by the iApp.

Additional information: Admitting Application Traffic to a DMZ

At the NS boundary of some application's DMZ you typically apply both positive and negative security policy. The positive policy is simple: you entertain only connections to the specified application (defined, in most cases, by LTM virtual servers on particular addresses and ports). The negative policy is more nuanced: you prevent certain undesirable connections and exclude bad behavior, but otherwise remain open to clients. The negative policy will have several layers. You might first use IP geolocation to exclude connections from certain countries (in the example, countries on the ITAR list), then accept all other connections which don't violate some IP Intelligence and anti-DoS policies. (When you filter by IP geolocation consider layering on IP Intelligence policy to exclude connections from origin-concealing proxies as the example here shows.) You should layer on a Protocol Security Profile to detect—and stop—many attacks at the application protocol layer (e.g., HTTP). If you implement the BIG-IP ASM for comprehensive application protection it will supersede the AFM Protocol Security Profile.

Of course you will apply a complementary (mainly positive) security policy at the DMZ's EW trust boundary. The NS policy keeps adversaries from, say, testing RPC exploits against your application servers. The EW policy keeps an adversary who finds, for example, a command-injection vulnerability on a DMZ web server from using the compromised machine as a "jump server" to attack the rest of the intranet.

Note that for destination addresses and ports in application virtual-server firewall rules you should put any:any rather than exact values. That lets you update virtual-server addresses or ports later without having to modify firewall rules as well. This method does not reduce security. Only packets which match a specific listener can ever reach the virtual-server context, so destination address and port get checked automatically. (With a wildcard virtual server like mgmt-to-dmz-6-vs you will often use firewall rules to control both source and destination addresses and ports. However, with a single-address-single-port application virtual server you need only worry about source addresses & ports.)

The example shows some BIG-IP good practices. The App1 webservers accept HTTPS over IPv6 only, so you SNAT incoming IPv4 connections and add X-Forwarded-For headers to demystify the webserver logs. You use a per-service SNAT pool rather than SNAT AutoMap so even if you put more services in the DMZ you won't risk port-exhaustion on your self IP. (It is customary to put the application virtual-server's own address in its SNAT pool to help identify traffic in network traces and server logs.) Finally, you create separate virtual servers for traffic coming from the Internet and the intranet. That way you can use appropriate TCP and acceleration profiles with each, and you can attach distinct security policies when appropriate (you might, for instance, attach different DoS Protection Profiles to Internet-facing and intranet-facing virtual servers).

Securing a Branch-Office, Cloud-Infrastructure, or Business-Partner Link

Organizational networks often include VPN links to branch office, cloud infrastructure, or business partner networks which have independent Internet connectivity. Typically such remote networks have firewalls, web proxies, and so on, but they still constitute different trust environments. From a security policy point of view, such networks resemble DMZ's—malicious actors might penetrate their local NS defenses then exploit their EW connectivity to attack the intranet.

BIG-IP LTM includes support for a variety of VPN/tunnel schemes and protocols using IPSEC, TLS, etc. Whether you terminate a site-to-site VPN on your BIG-IP device or route it through another device doesn't really matter. Either way, the packets crossing it come and go through BIG-IP VLAN/tunnel objects.

Setting up a VPN with LTM is beyond the scope of this document, but we look at configuring AFM to secure the link once you establish it. If you do terminate a VPN on BIG-IP system you may have to add rules to the global-object network firewall policy and possibly route-domain or virtual-server policies to admit VPN envelope packets. For instance, you might need to add a rule to policy "control-from-Internet" to accept IPSEC traffic.

Because the trust boundary between a remote network and the intranet resembles that between a DMZ and the intranet, a similar AFM configuration (for positive security policy) is appropriate. Note that this is an EW boundary; your data center firewall doesn't control the NS trust boundary between the Internet and the remote network. Typically you will trust a branch office more than a DMZ and permit it to send more types of traffic to more of the intranet.

Use the following table for guidance on configuring the BIG-IP objects.

Address Lists (*Navigate to Security > Network Firewall > Address Lists*)		
Name	**all-BR-nets**	**Notes (not a part of the configuration)**
Description	(example)	All branch-office subnets (part of intranet-nets).
Addresses/Regions	Type or copy and paste the following addresses into the Add new entry box: **10.8.0.0/16** | **fdf5:f:5:b000::/54**	
Name	**br1-nets**	**Notes (not a part of the configuration)**
Description	(example)	br1 = branch office #1
Addresses/Regions	Type or copy and paste the following addresses into the Add new entry box: **10.8.1.0/24** | **fdf5:f:5:b001::/64**	

Rule Lists (*Navigate to Security > Network Firewall > Rule Lists*. See previous Rule List sections for specific guidance)

Rule List Name	Rule Name	Protocol	Source address	Src. Port	Src. VLAN	Dst. Address	Dst. Port	Action
permit-br1-to-intra	ok-br1-udp	UDP	br1-nets	Any	vpn-BR1	intranet-nets	udp-EW-basic udp-EW-generic	Accept
	ok-br1-tcp	TCP	br1-nets	Any	vpn-BR1	intranet-nets	tcp-EW-basic tcp-EW-generic tcp-rdp-smb	Accept
permit-intra-to-br1	ok-to-br1	TCP	intranet-nets	Any	Any	br1-nets	tcp-rdp-smb	Accept
permit-mgmt-to-BR	ok-tcp-mgmt	TCP	net-mgmt-nets	Any	Any	all-BR-nets	tcp-EW-generic tcp-rdp-smb	Accept
	ok-udp-mgmt	UDP	net-mgmt-nets	Any	Any	all-BR-nets	snmp-mgr udp-dhcp	Accept

Network Firewall Policies (*Navigate to Security > Network Firewall > Policies*)	
Policy notes	*BIG-IP AFM may detect an overlap between rules in policy "intra-to-br1" permitting tcp-rdp-smb traffic to br1 from both the management network and the intranet generally. This is tolerable because removing the redundancy would make it harder to crack down later on non-management use of tcp-rdp-smb—you would have to edit two rule lists instead of just one.*
Policy Name	**intra-to-br1** Click **Finished** and then from the Policies table, click the Policy you just created.
Rules	Click the **Add** button and add the following rule. ***Name*** **mgmt2BR** ***Type*** **Rule List** ***Rule List*** **permit-mgmt-to-BR** Click **Repeat** and then add the following rule. ***Name*** **intra2br1** ***Type*** **Rule List** ***Rule List*** **permit-intra-to-br1** Click **Finished**.

Policy Name	**br1-to-intra** Click **Finished** and then from the Policies table, click the Policy you just created.
Rules	Click the **Add** button and add the following rule.
	Name **br1-intra**
	Type **Rule List**
	Rule List **permit-br1-to-intra** Click **Repeat** and then add the following rule.
Virtual Servers (*Navigate to Local Traffic > Virtual Servers*)	
IPv4 intranet to branch office virtual server	
Name	**intra-to-br1-4-vs**
Type	**Forwarding (IP)**
Destination Address	**10.8.1.0/24**
Service Port	*** (*All ports)**
Protocol	*** All Protocols**
Protocol Profile (Client)	Select the TCP Loose Fast L4 profile you created (**tcp-loose-fastl4** in our example).
VLAN and Tunnel Traffic	Select **Enabled On**, and then select only the **internal** VLAN you created.
Source Address Translation	**None**
	Click **Finished** and then from the Virtual Servers table, click the virtual server you just created. Next, we stage the initial policies. After you generate some test traffic and make any adjustments suggested by log analysis you can move your policies from staging to enforcement. On the Menu Bar, click **Security > Policies**.
Network Firewall: Enforcement	Select **Enabled**. From the **Policy** list that appears, select the Accept All policy you created in *Creating the Network Firewall Policies on page 18* (**accept-all** in our example).
Network Firewall: Staging	Select **Enabled**. From the **Policy** list that appears, select the intranet to branch office policy you just created (**intra-to-br1** in our example).
IP Intelligence	Leave IP Intelligence set to **Disabled**.
DoS Protection Profile	Leave DoS Protection Profile set to **Disabled**.
Log Profile	Select **Enabled**. From the **Available** list, select the Log profile you created in *Configuring AFM High-Speed Logging on page 23* (**afm-log-pfl** in our example).
IPv6 intranet to branch office virtual server	
Name	**intra-to-br1-6-vs**
Type	**Forwarding (IP)**
Destination Address	**fdf5:f:5:b001::/64**
Service Port	*** (*All ports)**
Protocol	*** All Protocols**
Protocol Profile (Client)	Select the TCP Loose Fast L4 profile you created (**tcp-loose-fastl4** in our example).
VLAN and Tunnel Traffic	Select **Enabled On**, and then select only the **internal** VLAN you created.
Source Address Translation	**None**
	Click **Finished** and then from the Virtual Servers table, click the virtual server you just created. On the Menu Bar, click **Security > Policies**.
Network Firewall: Enforcement	Select **Enabled**. From the **Policy** list that appears, select the Accept All policy you created in *Creating the Network Firewall Policies on page 18* (**accept-all** in our example).
Network Firewall: Staging	Select **Enabled**. From the **Policy** list that appears, select the intranet to branch office policy you just created (**intra-to-br1** in our example).
IP Intelligence	Leave IP Intelligence set to **Disabled**.
DoS Protection Profile	Leave DoS Protection Profile set to **Disabled**.
Log Profile	Select **Enabled**. From the **Available** list, select the Log profile you created in *Configuring AFM High-Speed Logging on page 23* (**afm-log-pfl** in our example).
IPv4 branch office to intranet A virtual server	
Name	**br1-to-intraA-4-vs**
Type	**Forwarding (IP)**
Destination Address	**10.0.0.0/8**
Service Port	*** (*All ports)**
Protocol	*** All Protocols**
Protocol Profile (Client)	Select the TCP Loose Fast L4 profile you created (**tcp-loose-fastl4** in our example).
VLAN and Tunnel Traffic	Select **Enabled On**, and then select only the branch office VLAN you created (**vpn-BR1** in our example).
Source Address Translation	**None**

	Click **Finished** and then from the Virtual Servers table, click the virtual server you just created. On the Menu Bar, click **Security > Policies**.
Network Firewall: Enforcement	Select **Enabled**. From the **Policy** list that appears, select the Accept All policy you created in *Creating the Network Firewall Policies on page 18* (**accept-all** in our example).
Network Firewall: Staging	Select **Enabled**. From the **Policy** list that appears, select the branch office to intranet policy you just created (**br1-to-intra** in our example).
IP Intelligence	Leave IP Intelligence set to **Disabled**.
DoS Protection Profile	Leave DoS Protection Profile set to **Disabled**.
Log Profile	Select **Enabled**. From the **Available** list, select the Log profile you created in *Configuring AFM High-Speed Logging on page 23* (**afm-log-pfl** in our example).
IPv4 branch office to intranet C virtual server	
Name	**br1-to-intraC-4-vs**
Type	**Forwarding (IP)**
Destination Address	**192.168.0.0/16**
Service Port	*** (*All ports)**
Protocol	*** All Protocols**
Protocol Profile (Client)	Select the TCP Loose Fast L4 profile you created (**tcp-loose-fastl4** in our example).
VLAN and Tunnel Traffic	Select **Enabled On**, and then select only the branch office VLAN you created (**vpn-BR1** in our example).
Source Address Translation	**None**
	Click **Finished** and then from the Virtual Servers table, click the virtual server you just created. On the Menu Bar, click **Security > Policies**.
Network Firewall: Enforcement	Select **Enabled**. From the **Policy** list that appears, select the Accept All policy you created in *Creating the Network Firewall Policies on page 18* (**accept-all** in our example).
Network Firewall: Staging	Select **Enabled**. From the **Policy** list that appears, select the branch office to intranet policy you just created (**br1-to-intra** in our example).
IP Intelligence	Leave IP Intelligence set to **Disabled**.
DoS Protection Profile	Leave DoS Protection Profile set to **Disabled**.
Log Profile	Select **Enabled**. From the **Available** list, select the Log profile you created in *Configuring AFM High-Speed Logging on page 23* (**afm-log-pfl** in our example).
IPv6 branch office to intranet virtual server	
Name	**br1-to-intra-6-vs**
Type	**Forwarding (IP)**
Destination Address	**fdf5:f:5::/48**
Service Port	*** (*All ports)**
Protocol	*** All Protocols**
Protocol Profile (Client)	Select the TCP Loose Fast L4 profile you created (**tcp-loose-fastl4** in our example).
VLAN and Tunnel Traffic	Select **Enabled On**, and then select only the branch office VLAN you created (**vpn-BR1** in our example).
Source Address Translation	**None**
	Click **Finished** and then from the Virtual Servers table, click the virtual server you just created. On the Menu Bar, click **Security > Policies**.
Network Firewall: Enforcement	Select **Enabled**. From the **Policy** list that appears, select the Accept All policy you created in *Creating the Network Firewall Policies on page 18* (**accept-all** in our example).
Network Firewall: Staging	Select **Enabled**. From the **Policy** list that appears, select the branch office to intranet policy you just created (**br1-to-intra** in our example).
IP Intelligence	Leave IP Intelligence set to **Disabled**.
DoS Protection Profile	Leave DoS Protection Profile set to **Disabled**.
Log Profile	Select **Enabled**. From the **Available** list, select the Log profile you created in *Configuring AFM High-Speed Logging on page 23* (**afm-log-pfl** in our example).

Additional Information: Securing Links to Remote Networks

There are two additional concerns when dealing with a cloud infrastructure or business partner network instead of a branch office: IP addressing and narrow service design.

A remote network may use addresses you did not assign. If they conflict with yours (a common occurrence when both sides use IPv4 private addresses) then address translation, preferably BIG-IP SNAT, is required. If they don't (e.g., both sides use IPv6 ULA prefixes), either routing or (S)NAT will be needed. Routing is preferable to address translation but watch out for a business partner who runs public IPv6 addresses on his intranet—to avoid problems with his firewall policies you may have to route to just his internal subnets via the VPN and continue to reach the partner's external services via the Internet.

Many VPN's to cloud infrastructure or business partner networks support just one or a few tightly-constrained services like EDI or batch file transfer. That is a favorable situation: just create specific virtual servers to handle those connections. That is more secure (and easier to audit) than IP forwarding virtual servers gated by AFM network firewall policies.

You may have to create some application-specific virtual-servers in any case to support ALG's for protocols like FTP or SIP.

The final issue is how much to restrict connection attempts from the intranet to the remote network. If you don't expect too much configuration churn, it is good practice to filter outbound connections like inbound ones—to specific addresses and ports. If nothing else, that may hobble APT's on the intranet trying to spread to the remote network.

Note that you might choose to accelerate branch office traffic with F5 BIG-IP Application Acceleration Manager (AAM). Configuring AAM is beyond the scope of this document, but it does not obviate any network security principles.

The example describes a branch office link. Branch office workers are given fairly free connectivity to standard applications in the intranet. Among enhancements you should consider: limiting destination networks (for example, to the data center only, though that would prevent RDP from branch office workstations to PC's in headquarters offices); expanding destination ports to permit additional applications (for example, additional VDI services); and channeling web traffic to a filtering proxy in the intranet so it can scan for malware and/or data exfiltration. For instance, you could create virtual servers listening to TCP ports 80 and 443 on the VPN VLAN/tunnel to send all web traffic directed to the intranet from the branch office transparently through the proxy.

The example assumes there are no business servers in the branch office—just PCs someone might wish to access remotely—so access to the branch office network is quite limited for most intranet users. However, the management network is privileged as usual to facilitate remote administration of equipment in the branch office. The example firewall policy blocks web traffic from the intranet to the branch office to prevent intranet users routing through the branch office to avoid the home office web proxy (If you create a rule to allow only the main web proxy to send web traffic to branch offices you can put web servers there while ensuring that head office users send all traffic through the main proxy—even traffic to servers in branch offices). If necessary you could add rules to permit access to specific services on the remote network.

Defending the BIG-IP system Itself with AFM

The details of security management for BIG-IP devices are mostly beyond the scope of this document. That is not to minimize their importance. Network infrastructure devices, especially those enforcing security policy are prime targets for malicious users. BIG-IP platform security guidance is collected at *F5 SOL13092: Overview of securing access to the BIG-IP system*. F5 also offers a deployment guide and iApp to help you secure BIG-IP devices according to NIST Special Publication 800-53r4, Recommended Security Controls for Federal Information Systems, available at *http://f5.com/pdf/deployment-guides/nist-sp-800-53-r4-dg.pdf*. However, since not all BIG-IP devices have AFM provisioned, neither SOL13092 nor the F5 NIST SP800-53 deployment guide explains the role of AFM in BIG-IP platform security.

This deployment guide does show how to protect the BIG-IP management (network) port and self IPs using AFM network firewall policies.

Protecting the BIG-IP Management Port with AFM

It is wise (though not always feasible) to separate network management traffic from ordinary application traffic onto special "management network" subnets and VLANs. The BIG-IP system is optimized for such an environment (as the BIG IP management port *must* be homed on a different subnet than any TMM self IP). Some organizations, though, allow management traffic to originate from non-management network subnets. Such traffic, most commonly from administrators using workstations on corporate office subnets is merely routed through the management network.

The BIG-IP management port is pretty well hardened. To protect it further using a network firewall policy you must be able to recognize legitimate management traffic by source address. That is fairly easy when all such traffic comes from management network subnets. Otherwise you may have to maintain lists of other trusted subnets, use AFM identity-based security, or use an AAA gateway like BIG-IP APM to SNAT or VPN authorized traffic from less trusted networks onto the management network.

The model network architecture has a management network which owns all the subnets of 192.168.192.0/18. You can use AFM to ensure that only traffic from those subnets reaches the BIG-IP management port. You must update the nameless firewall policy permanently enforced on the management port to accept packets from management subnets and explicitly Drop packets from all other sources. (By default, packets which do not match any rule in the management-port context are Accepted. There is no fallback to the final (global-default) context from the management-port context.)

Warning *It is possible to lock yourself out of a BIG-IP management port by making an error in the management port firewall policy. Then, if you cannot access TMSH or the BIG-IP management GUI through some other port you may not be able to correct the error.*

Before you configure any AFM firewall policy on the management port, ensure you have access to:

- *The combination of (1) a BIG-IP self IP address with Port Lockdown set to "Allow Default" and (2) a permissive firewall policy (such as /Common/allow-all) enforced on that self IP object (allowing access from your PC to ports 22 and 443 at least) and (3) System / Platform / User Administration / SSH Access set to "Enabled"; or*
- *The serial console port on a hardware BIG-IP device; or*
- *The KVM virtual console of a BIG-IP VE device.*

Test your alternate access before changing the management port firewall policy. After you have configured and successfully tested your management-port firewall policy, you may adjust the firewall policy and/or Port Lockdown setting on any self IP you used as an alternate management access point.

Securing the management port

On each BIG-IP (big-s1 and big-s2 in our example—management-port settings are not synchronized across devices), to secure the management port add the following firewall rules in order.

Management Port Rules (*Navigate to System > Platform > Security > Add Rule*)	
Name	**ok-mgmt-nets**
Protocol	**Any**
Source > Address/Region	Add **net-mgmt-nets** and **fe80::/10**
Source > Port	**Any**
Destination > Address/Region	**Any**
Destination > Port	Add **udp-EW-basic** and **udp-EW-generic**
Action	**Accept**
Logging	**Disabled** Click **Repeat** and then add the following rule

Management Port Rules (*Navigate to System > Platform > Security > Add Rule*)	
Name	**no-others**
Protocol	**Any**
Source > Address/Region	**Any**
Source > Port	**Any**
Destination > Address/Region	**Any**
Destination > Port	**Any**
Action	**Drop**
Logging	**Enabled**

This completes the management port rules.

Deep Packet Inspection with AFM (Blocking Teredo)

BIG-IP AFM is not a dedicated deep-packet-inspection tool like an IDS. However, one job of a data center firewall is to detect and block attempts to evade network security policy. That is a different project than looking for malware signatures in application data. Some evasion techniques can only be recognized by fairly sophisticated deep-packet-inspection filters. The BIG-IP platform lets you implement complex filters using iRules. For example, you may use an iRule to block Teredo.

Teredo (*rfc4380*) is a protocol for conveying IPv6 packets across the IPv4 Internet. It is designed to "worm" through NATs and stateful firewalls. A Teredo tunnel is a sort of VPN; though unencrypted, it conceals the source, destination, and content of IP traffic it carries from most of the firewalls through which it passes. Teredo is a legitimate network protocol which may be misused to evade network security policy in many situations.

All versions of Microsoft Windows from XP-SP2 onward incorporate Teredo client drivers. Drivers also exist for other operating systems such as Linux and Mac. Special-purpose platforms may also use Teredo; Microsoft Xbox, for instance, routinely creates Teredo tunnels.

Because Teredo may be used to smuggle all sorts of network traffic past security policy (including APT traffic, botnet control channels, AUP-violating web traffic, etc.), it is generally wise to block it.

However, it is not easy to block Teredo tunnels without interfering with other, sanctioned network traffic. A couple of UDP ports are well-known to be used by Teredo but they are neither reserved for Teredo nor necessary for it to work. Closing the well-known ports won't prevent the use of Teredo but will disrupt other application traffic. The only sure way to block Teredo with a typical network firewall rule is to close all UDP ports—a cure commonly considered worse than the disease.

You may use a BIG-IP AFM iRule to block Teredo tunnels on all ports without disrupting other application traffic. The iRule analyzes the initial packets of UDP flows to detect Teredo connection attempts and reject them. This is particularly efficient on the BIG-IP system because only one packet per flow need be examined. Attached to the virtual server for outbound UDP traffic the iRule prevents use of Teredo tunnels from the intranet.

Create the following iRule then attach it to virtual server for outbound UDP traffic (**udp-icmp-outbound-4-vs** in our example).

iRules (*Navigate to Local Traffic > iRules*)

Name Give the unique name. We use **ir-block-teredo**

Definition Copy and paste the following iRule, omitting the line numbers:

```
1  when FLOW_INIT priority 400 {
2   # to prevent Teredo tunnels we recognize and block Router
3   # Solicitation messages to Teredo servers.  This closes all
4   # ports to Teredo without interfering with other services
5   # on those ports.  (Teredo is not limited to port 3544; for
6   # example, Microsoft Xbox uses 3074 for Teredo.)
7   #
8   # (Attach this iRule to your forwarding-virtual-server for
9   # outbound UDP traffic.)
10
11  if {([IP::protocol] != 17) ||
12      ([set len [eval {UDP::payload length}] < 48)} { return }
13
14  # Teredo authentication header may precede RS
15  set ofs 0
16  binary scan [eval "UDP::payload 4"] Scc aID CL AL
17  if {$aID == 1} {
18   set ofs [expr {13 + ($CL & 0xff) + ($AL & 0xff)}]
19   if {$len < (48 + $ofs)} { return }
20  }
21
22  #extract possible Teredo RS
23  set buf [string range [eval "UDP::payload [expr {$ofs + 48}]"] $ofs end]
24
25  #Teredo RS will have a bunch of fields set to particular values,
26  #we look for IPv6 version, length, next header, destination
27  #address upper and lower portions, source address prefix and
28  #flag bits, and ICMPv6 RS type-code.  Those seem sufficient but
29  #if we get false positives we can also check the hop-count,
30  #reserved-word contents (~0), and/or validate ICMPv6 checksum.
31
32  binary scan $buf H1@4SccwS@24wWsSI \
33              Iver Ilen Inxt Ihop Ipfx Iflg Id1 Id2 \
34              RStypc RSchek RSrsvd
35
36  if { ($Iver eq "6") && ($RStypc == 133) &&
37       ($Id1 == 0x2ff) && ($Id2 == 2) &&
38       ($Ilen == 8) && ($Inxt == 58) &&
39       ($Ipfx == 0x80fe) && (($Iflg & 0x7fff) == 0) } {
40   #this packet is almost certainly a Teredo RS
41   reject ; #no tunnel for you!
42  }
43
44  #otherwise this packet is not a Teredo RS, let it pass
45 } ; #FLOW_INIT 400
```

Appendix A: Supplemental Information

In this appendix, you'll find additional and complementary information to the guidance provided in this document.

Dual-Stack IPv6/IPv4 Networking

The world ran out of IPv4 addresses in 2014. Many firms and most individuals haven't quite noticed but now IPv6 is the only platform for growth. Internet service providers and their largest customers have already made the change. Every organization must follow suit.

A decade of industry experience with IPv6 has taught everyone that a dual-stack network with intelligent proxy gateways is key to a successful transition from IPv4 to IPv6. Though dual-stack clients are handy, it is not critical for servers to be dual-stacked so long as the network is and gateways are provided to help application traffic get from one addressing realm to the other.

By definition, a dual-stack network carries both IPv4 and IPv6 traffic—so it can potentially carry harmful traffic in either realm. Even if your network is not yet officially dual-stack, you have IPv6 traversing the network already because every version of Microsoft Windows since XP has been dual-stack. At this time, the real problem with dual-stack networking is maintaining your security posture. You have many adversaries eager to use IPv6 to evade IPv4-based controls and vice-versa. Your team may be less experienced with IPv6 vulnerabilities. The question is whether your dual-stack network will be efficient and secure. The only way to ensure either is to configure your infrastructure so both IPv4 and IPv6 are fully managed. You must make security planning, policies, enforcement, and monitoring coextensive for IPv6 and IPv4—ideally using tools like the BIG-IP system which unify configuration and processing of both.

The era of tunneling IPv6 around is over for businesses and other organizations (for end-users that is—carriers and ISP's have special uses for tunneling and cross-realm NAT schemes). Business-class ISP's are dual-stack; they give simultaneous IPv4 and IPv6 service. You want both, so you can grow using IPv6 addresses while interoperating with the IPv4 installed base. Getting rid of tunnels gets rid of many operations and security challenges as well.

Of course, IPv6 clients on a dual-stack network can talk to IPv6 servers and IPv4 clients can talk to IPv4 servers. In the past it was not obvious how clients in one realm should talk to servers in the other. Given a large installed base of IPv4-only servers and a growing supply of IPv6-only servers, the industry experimented with various NAT schemes to facilitate cross-realm communication. None was really satisfactory. However, dual-stacked proxy gateways worked beautifully, and the F5 BIG-IP is the archetypical proxy gateway.

BIG-IP virtual servers can accept traffic on IPv4 and/or IPv6 addresses, and then proxy (and load balance) application traffic to destinations in the other realm—or the same realm; the client neither knows nor cares. You may add a server using IPv6 alongside one using IPv4 and then use the BIG-IP system to serve even single-stack clients from both servers at the same time.

In the optimal dual-stack network design—experimentally refined since the late-2000's—you home your real servers logically behind the BIG-IP device (where it doesn't matter which stack they run) and publish both IPv6 and IPv4 virtual servers to the intranet and to the world. Clients connect via the address realm they find most convenient and the BIG-IP system proxies the connections. When your single-stack clients need to connect to some remote service which is only offered in the other realm, you may proxy the outbound connection—the distant server never knows it is (logically) behind your BIG-IP device. The BIG-IP system even supports dynamic gateway configuration with automatic DNS translation for IPv6 clients.

The model network architecture for this Deployment Guide is dual stack and so are the practical examples. For instance, the network firewall policy examples filter IPv4 and IPv6 traffic simultaneously to demonstrate unified security policy in the dual-stack network.

BIG-IP in a Dual-Stack Network

This Deployment Guide shows how to configure your BIG-IP for dual-stack networking. However, there are a few points which the examples don't illuminate:

- You should not add any server to a particular LTM pool (as a member) more than once, even if that server has both IPv6 and IPv4 addresses (i.e., is multihomed).
- Don't attach a Fast HTTP Profile to a virtual server with an IPv6 address (see *SOL11449*).
- You may configure *dynamic IPv4 to IPv6 DNS translation* and set up dynamic UDP and TCP gateways to allow IPv6-only clients to connect to IPv4-only services without identifying those services in advance. The details are beyond the scope of this document, but not very difficult.
- On very busy IPv6 networks, AFM DoS Protection Device Configuration may classify benign ICMPv6 messages with type numbers above 132 as DoS attacks. You may adjust the detection threshold if necessary.

BIG-IP AFM Overview

As a reminder, this document is based on BIG-IP/TMOS version 11.6 and 12.0. Subsequent releases of AFM may include features beyond those discussed here.

The major components of AFM are Network Firewall, IP Intelligence, Denial-of-Service (DoS) Protection, and Protocol Security. All are integrated into the BIG-IP full-proxy architecture and with LTM especially.

Only the Protocol Security component normally inspects packets very deeply, but the Network Firewall may perform deep packet inspection with iRules (for a practical example, see *Deep Packet Inspection with AFM (Blocking Teredo) on page 48)*, and the DoS Protection DNS and SIP Profiles deal with complex protocols. Of course AFM is just one part of the BIG-IP Application Delivery Firewall. The BIG-IP system enforces protocol compliance at multiple layers using LTM configuration objects and ALG's.

You typically use the Network Firewall to apply some negative and much positive security policy. The limitation of the Network Firewall is that it gets only one chance to make a decision about each connection, chiefly by looking at the L2–4 characteristics of its initial packet. That really is sufficient control for 99+% of traffic across the network, but in nearly all customer environments a few superficially-legitimate connections will actually come from malicious actors who abuse connections to transmit attacks.

IP Intelligence, DoS protection, and Protocol Security support mostly negative security policy. IP Intelligence lets you recognize and exclude connection requests from known-bad origins. DoS Protection and Protocol Security recognize and limit misuse of active connections which seemed legitimate when initiated.

AFM Protocol Security defeats many attacks at the HTTP or DNS protocol level. However, to defend against attacks on application logic (for example, web SQL-injection attacks, which are transmitted in syntactically-valid HTTP requests) you will want an application-level firewall like F5 *BIG-IP Application Security Manager (ASM)*.

AFM's tight integration with other BIG-IP features is essential to overall system capability and performance. For example, BIG-IP LTM TLS (SSL) processing enables Protocol Security to protect secure websites.

The examples in this document illustrate at least basic use of each major AFM component.

From Zone-Based Firewall to Application Delivery Firewall

With the BIG-IP system, the constructs *interface* and *zone* commonly used to configure less advanced firewalls are superseded by listener and application. Application-based security policy is both more granular and better-informed than zone-based policy. Tying policy to applications frees administrators to move application-related resources around without the traditional problem of detaching them from their carefully-crafted security controls. This helps avert the gradual lessening of security due to "rule rot" and the accumulation of abandoned zone-policy rules which degrade firewall performance and auditability.

See the F5 *Tech Brief Replacing Abstract Zones with Real Application Security Policy* for a high-level discussion.

Network Firewall

This section contains additional information on the AFM Network Firewall.

AFM Network Firewall Structure

An AFM network firewall *policy* is an ordered collection of firewall *rules* attached to a BIG-IP traffic-handling object (such as a *listener* like a virtual server) which decides whether that object will process an incoming *packet*. The packet (more precisely, a set of attributes describing the packet—such as its IP protocol, source and destination addresses and ports, the VLAN from which it arrived, etc) is tested against the policy's rules, in order, until a match is found or no rules remain. The *action* of the first matching rule determines the result of the policy. (If no rule matches the packet another policy will judge it. A last-resort policy discards never-matched packets.)

Each object belongs to a firewall *context*. All similar objects (e.g., virtual servers) belong to the same context. Each context belongs to a hierarchy that determines the order in which policies are applied to each packet entering the BIG-IP system.

The contexts are: *global; route-domain; virtual-server; self IP; global-default* (which is more easily understood as *final*); and *management-port*. The primary context hierarchy includes (in order of policy application) the global, route-domain, virtual-server *–or–* self IP, and final (global-default) contexts (see Figure 2). The second hierarchy includes only the management-port context.

The context hierarchy helps keep policies simple. The global-context policy, for example, may contain general rules (like one to discard packets improperly sourced from multicast addresses) which you would otherwise have to tediously place into every virtual-server's policy. There is no "VLAN context," but VLAN-specific rules in the global context give almost the same effect.

You may attach up to two policies to most objects (the management port is an exception): one *enforced* policy and one *staged*. The *action* determined by an *enforced* policy actually controls whether a packet proceeds through the BIG-IP system. The *action* determined by a *staged* policy is logged but not effectuated. The point of staging a policy is to find out how it would affect traffic *if* you were to enforce it—so you may debug policies before applying them. Staged policies have an auxiliary use: counting and logging selected packets without changing the mainline configuration of the BIG-IP system. For example, you could count connections initiated from subnet 203.0.113.0/24 by staging a policy with a rule of the form "Accept src 203.0.113.0/24:any dst any:any." Staged rules don't interfere with traffic, but they do count matches for your review.

The global and management-port contexts each contain only one object. The management- port object has one nameless but editable policy permanently enforced on it.

When you have not enforced a named policy on an object, BIG-IP AFM enforces a default *implicit* policy which matches packets from any source addressed to the object according to its normal properties. For example, the implicit policy for a virtual server listening on VLAN "external" for web traffic to 203.0.113.7 would amount to "*take action X on packets from source= any:any, via VLAN= external, protocol= TCP, to destination= 203.0.113.7:80.*" All *implicit policies* on virtual servers and self IPs take the *same* action X, which you set in Security > Options > Network Firewall > Virtual Server & Self IP Contexts to one of Accept, Reject, or Drop.

When you set the action for implicit policies on virtual servers and self IP's to *Accept*, AFM runs in *ADC mode*. That means BIG-IP traffic-handling objects process the packets you would expect them to and all other packets get discarded. Implicit policies are dynamic. They are neither visible in per-object configuration nor recorded in BIG-IP configuration files. However, you may see the implicit policies active at run time by navigating to Security > Network Firewall > Active Rules and selecting Context "All (Show implied rules)". Since global and route-domain objects are promiscuous, the implicit policies for them match all packets (those packets are subsequently filtered by policies further down the context hierarchy).

Some organizations wish all firewall policy to be explicit and visible in configuration files—to facilitating auditing, or to take a belt-and-suspenders approach to configuration change control. You may run AFM in *Firewall mode* by setting the action for implicit policies to Reject or Drop. In Firewall mode, no virtual server or self IP object will process any traffic until you enforce on it a named firewall policy which has at least one rule with action "Accept". To activate a new service you must, e.g., both create a virtual server and enforce a named policy on it. Firewall mode is not inherently more secure than ADC mode (though it is more cumbersome). In both modes network firewall policy controls all packets which arrive to the BIG-IP. (Per the model network architecture, the examples in this document show AFM in Firewall mode.)

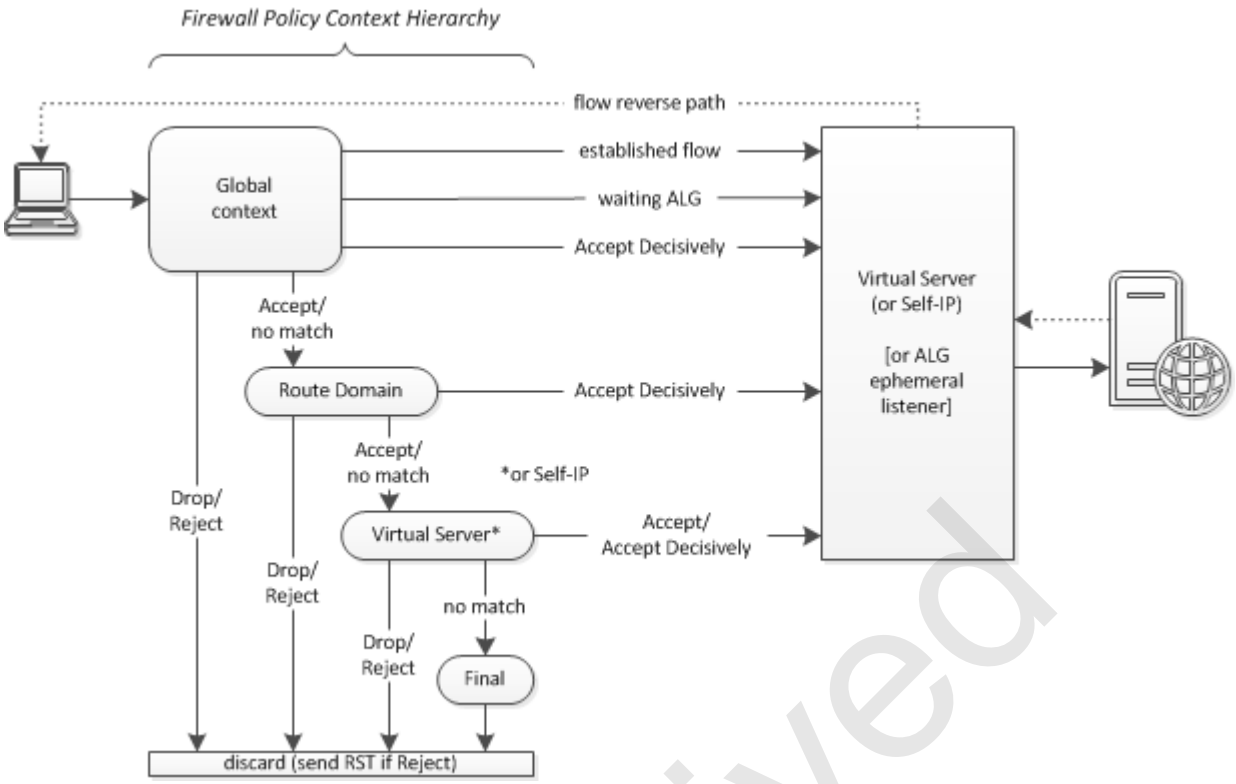

Figure 3: The Firewall Policy Context Hierarchy (omits management-port context)

The firewall policy on the global object—the sole global-context policy—applies to all packets reaching the BIG-IP except those arriving to the management port (which has an independent context and policy). If a packet traverses the global context it will reach the route-domain context where AFM will apply the policy enforced on the relevant route-domain object. If the packet traverses the route-domain context it must find a listener (either a virtual server or a self IP object) or it will be discarded. Assuming the packet finds a listener, then in most cases AFM will apply the policy enforced on that object (in the virtual-server or self IP context as appropriate). For performance reasons, a firewall rule in the global or route-domain context may authorize a packet to bypass the virtual-server- or self IP-context policy (for instance, when a packet arrives from a highly-trusted subnet).

A typical incoming packet traverses the global and route-domain contexts, then the virtual-server or self IP context. In every context, the packet is tested against the applicable policy's rules in order. The first rule to match a packet determines whether it will be forwarded to the next context (or otherwise processed). When the first matching rule has action Reject or Drop, AFM discards the packet (silently for Drop, or with notice to the sender for Reject). However, in the global and route-domain contexts, if the first matching rule has action Accept, or if no rule matches the packet, then the packet will merely be forwarded to the next context. In other words, actions Drop and Reject are always final in any context, but action Accept is provisional until a packet reaches either the virtual-server or self IP context, where Accept leads to processing by the listener.

It may happen that no rule in a policy matches a packet. In that case the packet is forwarded to the next context. However, the next context after the virtual-server and self IP contexts is the final (global-default) context where a fixed single-rule policy matches and discards all packets that reach it. In other words, to get processed by a listener, a packet must match an Accept rule in the virtual-server or self IP context, or an Accept-Decisively rule in the global or route-domain context. Accordingly, in the virtual-server and self IP contexts, actions "Accept" and "Accept Decisively" are equivalent.

Three invisible rules are always logically prepended to the global-context policy, where they may match a packet before any normal rule gets a chance to match it. The first invisible rule is the most important: it Accepts Decisively any packet which belongs to an established flow to or through the BIG-IP.

A flow represents a network connection, i.e., a bidirectional stream of closely-related packets. For example, when a virtual server replies to a TCP SYN packet from a client, it creates a flow to recognize additional packets in the new TCP connection by the classic five-tuple of {protocol, source address and port, destination address and port} plus some additional criteria such as route domain and VLAN. Outbound connections create flows too. Flows are logical entities represented by entries in a flow table (you can use the

TMSH command **`show /sys connection`** to inspect the flow table). Since the BIG-IP system operates most of the time as a full proxy, what seems to be a single (logical) connection at the application layer commonly has two flows, one each on the "client" and "server" sides of the BIG-IP device.

In combination with the first invisible rule, flow-table entries act as ephemeral firewall rules to permit packets for established connections to pass the firewall very efficiently. AFM's support for flows also removes any need for "reverse" firewall rules. Nearly every packet that leaves the BIG-IP is part of a flow (including packets sent by BIG-IP processes like service heath monitors), so AFM will recognize related packets coming the other way and Accept them Decisively in the global context. (A packet accepted by AFM may still be discarded for some other reason.)

The first invisible rule means that once the initial packet of a new connection has been accepted for processing by a virtual-server or self IP object, other packets (flows) associated with that connection are accepted by association. (Of course when a traffic-handling object like a virtual-server refuses or terminates a connection, that removes all associated flows.)

The second invisible rule discards every TCP packet which does not belong to a flow except for SYN packets requesting new connections.

The third invisible rule Accepts Decisively any packet for which an ALG (Application Layer Gateway) is specifically listening. Like other listeners, ALG's may accept connection requests and establish new flows. This rule has the same security implications as the first invisible rule because ALG listeners and connections are always auxiliary to ordinary listeners' connections.

The three invisible rules make AFM a *stateful* firewall so far as packet-matching rules are concerned. Of course, the BIG-IP is really an Application Delivery Firewall. BIG-IP is much more "stateful" than most network firewalls because BIG-IP maintains connection state at Layers 2–7 and enforces protocol compliance at every layer.

Finding a Listener

To choose which policy will judge a packet in the route-domain, virtual-server, or self IP context, AFM interacts with a fundamental and critical feature of BIG-IP: each (accepted) connection will be handled by the listener (virtual server, usually) that offers the most specific match for it based on VLAN, IP address, port, and protocol.

When defining firewall rules you must order them carefully to make "more specific" ones match before "less specific." For example, to let traffic reach one host on a subnet while blocking access to all other hosts on that subnet you would put rules like "no-net1: deny src any dst 203.0.113.0/24" and "ok-host7: permit src any dst 203.0.113.7/32" in the order (ok-host7, no-net1). If you were to reverse that order, the "block whole subnet" rule (no-net1) would match before the "allow particular host" rule (ok-host7) so no packet would ever reach 203.0.113.7.

Matching connections to BIG-IP listeners *doesn't* work that way.

The BIG-IP automatically chooses the *best-matching* listener for each connection. Suppose you configure listeners *P* on 203.0.113.0/24:any and *Q* on 203.0.113.7/32:any. When a packet addressed to 203.0.113.7:993 arrives, BIG-IP will deliver it to the more-specific listener *Q*. A packet addressed to 203.0.113.5 would go to the less-specific listener *P*. Suppose you replace *Q* with two listeners *R* on 203.0.113.7/32:80 and *S* on 203.0.113.7/32:22. In that case a packet addressed to 203.0.113.7:993 would find listener *P* on 203.0.113.0:any. Because it offers the best match—the others are too picky.

This approach lets BIG-IP listeners "peel off" portions of the traffic stream for various kinds of treatment. For example, each ALG you configure has a listener which attracts precisely the traffic the ALG is meant to handle (based on IP protocol, port, etc.) while other traffic goes to a more generic listener (or to the bit-bucket).

Virtual servers are the most important type of listener. Virtual servers provide the nexus between application traffic and security policy (really, all kinds of application-delivery policy) at all layers. You bind network firewall policies to applications as well as mere routing paths (e.g., IP forwarding virtual servers) by applying them in the virtual-server context.

AFM with Virtual Servers for Positive Security

BIG-IP's listener-matching model embodies a positive security policy. Indeed, to create a pretty effective firewall you can simply define listeners for desirable traffic and let BIG-IP discard everything else. However, the generality of BIG-IP with AFM lets you choose from two approaches to enforcing strict security policy at trust boundaries. You may configure specific virtual servers to collect and proxy each class of traffic from one trust environment to selected resources in another, or you may configure IP forwarding virtual servers and use AFM network firewall policy to filter packets going through them. The second approach resembles a traditional interface-and-port (zone-based, router-type) firewall. It is simple to set up but tricky to maintain. The first (specific-virtual-server) approach is a bit more trouble to set up but easier to "keep tight."

In practice you will use both approaches, as shown in the examples here. When you want to enforce security above Layer 4 (for instance, using an AFM Protocol Security Profile) and when you wish to be very careful (for instance, when exposing a service to the Internet) you will configure application-specific virtual servers backed by specific resource pools. When packet-based security will cover most of your needs (for example, controlling EW traffic to well-known application ports) and when you anticipate that network users (say, sys-admins in the data center) may move services and clients without telling you, you will set up generic virtual servers and use network-firewall policy to keep traffic through them fairly clean.

(Perhaps the toughest reorientation for experienced zone-firewall admins mastering BIG-IP is to stop trying to guess which interface an incoming packet is likely to leave through. Let the BIG-IP listener worry about that. BIG-IP is more than a router, and in many cases an incoming packet will never leave because it is part of a flow to one side of a BIG-IP virtual server (proxy) that will forward only payload data (possibly transformed) to the remote destination. The critical security question is: should an incoming connection request be accepted (and corresponding flows established)? The combination of an appropriate listener and the AFM policies or profiles attached to it will provide the answer. Once a connection request is vetted, LTM will handle communications to the ultimate resources.)

AFM Network Firewall Rules

Network firewall *rule lists* collect rules for use in one or more network firewall policies. Each rule in a *policy* may match packets directly (by protocol, addresses, etc.) or refer to a rule list. Rules in rule lists must match packets directly; rule lists cannot be nested. When a policy rule refers to a rule list, packets are tested against the rules in that list in order, as if those rules appeared in the policy at the point of reference.

For example, here are two rules that make up a policy to control access to an email gateway:

Policy Name	Rule Name	Protocol	Source address	Src. Port	Src. VLAN	Dst. Address	Dst. Port	Action
email-gw	no-419s	TCP	Nigeria (NG)	Any	external	Any	25	Drop
	ok-smtp	TCP	Any	Any	external	Any	25	Accept

If you modify a rule list—or a rule inside a rule list—all policies which refer to that rule list will be changed. However, altering a direct-matching rule in one policy will not affect other policies even if they include similar rules. Since one rule list cannot reference another, active rules are never more than one step removed from a policy.

Even though you give each rule inside a policy or rule list a name, you cannot refer to that rule elsewhere. To use a particular rule in more than one policy you must add it to a rule list (which you can reference in multiple policies).

Address lists collect addresses for use in one or more rules. Address lists may be nested. The order of addresses in a list is not significant.

Each *address* in a rule or an address list is static, dynamic, or symbolic. Static addresses consist of an IPv6 or IPv4 prefix and (subnet-) *mask length* (CIDR notation). If the mask length is 128 (IPv6) or 32 (IPv4) the address specifies a host (when you don't supply a mask length, a length of 128 or 32 is presumed). Otherwise the address specifies a subnet or a block of adjacent subnets. A static address in a rule matches a packet address when the leftmost mask-length bits of both addresses are the same. For instance, a subnet address matches all host addresses in that subnet. Dynamic addresses include geolocation and user-identity indicators. At run time, rules which refer to dynamic addresses test whether some actual (source or destination) address in a packet matches a given geolocation indicator or user identifier according to system logic. Address "any" signifies ::/0 and 0.0.0.0/0 *simultaneously*. (To use geolocation effectively you should download and install updates to the geolocation database at least monthly—refer to *SOL11176*. IP addresses are linked to user identifiers in the BIG-IP's IF-MAP server.)

A symbolic address is a fully-qualified domain name (FQDN) that a DNS resolver translates to one or more IP addresses which are then used to match packet addresses. Symbolic addresses let you tie security rules to named services or logical hosts rather than to ephemeral IP addresses (as commonly seen in cloud computing), thereby minimizing rule churn. The BIG-IP system refreshes FQDN-to-IP-address translations at intervals controlled by an adjustable timer. Note that the IP addresses discovered for each FQDN will be used during the whole refresh interval even if their DNS time-to-live (TTL) values are shorter. To avoid having rules with symbolic (FQDN) addresses match packets against stale IP addresses, coordinate your network firewall FQDN Resolver Refresh Interval with the TTL values in the DNS address resource records (A and AAAA RR's) for your FQDN's. Symbolic addresses are supported by TMOS v12.0 and later. To view the current IP-address translations for an FQDN, issue a command of the following form (replace xxxx with an FQDN like "example.net" or with "all"):

```
tmsh show /security firewall fqdn-info fqdn xxxx
```

Symbolic addresses are convenient but using them expands the attack surface of your firewall to include your DNS service. Information about firewall rules will leak to adversaries who snoop on DNS traffic. An adversary who can alter or spoof DNS entries can effectively modify (some of) your firewall rules. When using symbolic addresses you should: (1) secure your own DNS nameservers against unauthorized access (including via dynamic DNS updates); (2) configure BIG-IP AFM for more secure DNS resolution; (3) provide a secure network path (such as a management-access-only subnet and VLAN) between your BIG-IP devices and any local nameservers they access; and (4) implement DNSSEC to secure the RR's for your symbolic addresses. By default, AFM uses an insecure BIG-IP Network DNS Resolver which sends queries directly to the Internet. Consult *Appendix C: Securing AFM Domain Name Resolution on page 66* for instructions on configuring more secure DNS resolution for BIG-IP AFM.

An address with a route-domain identifier (%ID) appended will match packet addresses in the indicated route domain; addresses without route-domain ID's will match packets from any Route Domain. You can use rules with addresses lacking route-domain identifiers to filter just packets received from a particular route domain by putting those rules in a policy applied to the route-domain object. A policy on a route-domain object only ever sees packets from that route domain.

When writing rules be mindful that address "any" matches both IPv6 and IPv4 addresses even if all the other rules in some policy or rule-list refer to, say, IPv4 only.

Port lists collect port numbers for use in one or more rules. Port lists may be nested, which is handy for putting names to port numbers.

A rule which specifies a VLAN matches only packets that enter the BIG-IP from that VLAN. All AFM firewall rules are "ingress" rules.

If you put a rule in *disabled* state it will not match any traffic. You may enable/disable rules on a time schedule.

Use action Reject only in rules which discard packets arriving from your intranet (including business-partner networks via secure VPN links) with intranet source addresses. Always "Drop" unwanted packets from the Internet (or from bogus source addresses!). Rejecting unwanted packets helps you find and correct configuration problems in your intranet. However, you cannot fix configuration problems outside your network. If you Reject packets from the Internet you cause two problems. First, you let bad actors launder attacks through your facilities just by forging packet source addresses—when you Reject such packets, you actually send useless RST or similar packets to the real owners of spoofed source addresses (annoying them and damaging your own reputation). Second, you enable DoS attacks to waste outgoing bandwidth as well as incoming!

When a firewall rule matches a packet it may invoke a specified *iRule*. That iRule may override the firewall-rule's action (thus determining the policy result) and/or do something like log a message or tinker with subsequent processing. An iRule may inspect a packet deeply. For example, an iRule could check some higher-level protocol data inside a UDP packet to decide whether to send it to an alternate listener (virtual server) after accepting it. Since iRules are handy for logging packet characteristics (e.g., length) other than normal matching criteria, you may configure a rule to invoke an iRule only for a sample (say, 1%) of packets matched.

You may wish to disable logging on rules which match a lot of benign packets. You probably should log (and react to) indications of configuration errors or APT activity in your intranet—including packets with bogus source addresses reaching the firewall from the intranet and attempted connections to Internet addresses with bad reputations.

Defining and Attaching Network Firewall Policies

You may create and edit a firewall policy before (or while) enforcing or staging it on an object. You may attach a given policy to more than one object. (The management port is exceptional.) Changing the policy on an object will not interrupt any flows established under a previous (named or implicit) policy, even if those flows could not be initiated under the new policy.

To create a new firewall policy using the TMOS 11.6 management GUI, navigate to Security > Network Firewall > Policies and click the Create button. To edit an existing policy, simply click its name in the list.

After you create a policy you may enforce or stage it on a route domain, virtual server, or self IP object from the Security policies tab for that object (e.g., Network / Route Domains / Route Domain 0 / Security).

The BIG-IP management GUI support for global-context firewall policy is a bit confusing in TMOS 11.6. To enforce or stage a policy on the global object, navigate to Security / Network Firewall / Active Rules, then click the word "Global" at the left end of the very first line in the list (ignore everything else on that line). The Security / Network Firewall / Active Rules / Global Firewall Rules page will open. To enforce a policy go to the Network Firewall block, set "Enforcement" to "Enabled", then set "Policy" on the same line to the policy's name (do not choose "Create"). To stage a policy, set "Staging" to "Enabled" and "Policy" on the same line to the policy name (do not choose "Create"). Then click the Update button. Beware of the rule list at the bottom of the page. Use it to review rules (including

their status and hit counts) but not to change rules. Do not click the “Add” button on this page. Note: the “Policy Type” setting at the top of the list changes which policy (enforced/staged) you see.

The single permanent network-firewall policy for the BIG-IP management port is found at System / Platform / Security. For important advice about changing it, review Protecting BIG-IP Management Ports with AFM elsewhere in this document.

Optimizing Network Firewall Policies

For best system performance you may optimize your network firewall configuration in several ways: (1) when possible replace duplicated data in rules and policies with references to address lists, port lists, and rule lists; (2) specify port ranges and address ranges instead of listing adjacent ports or addresses (that is, use I–N rather than I,J,K,L,M,N); (3) when possible summarize adjacent subnets or addresses as CIDR blocks (e.g., 10.0.0.0/24 and 10.0.1.0/24 become 10.0.0.0/23); (4) when you have two (or more) rules which specify the same source and different destinations (or vice-versa), and the different destinations (or sources) could be summarized using address and/or port blocks or ranges, consolidate those rules; (5) check (in the Active Rules List) for and remove or correct overlapping and conflicting rules; and (6) remove unused address/port/rule lists and firewall policies from the BIG-IP configuration.

IP Intelligence

AFM IP Intelligence offers enhanced capabilities for dynamic address matching. Like network firewall rules that match packet addresses against geolocation or user identifiers, IP Intelligence policies match packet source addresses against *blacklist categories*. (In TMOS 11.6 matching packet destination addresses against IP Intelligence blacklist categories requires an iRule like the example afm-no-evil-dst in this document.) Categories represent dynamic lists of (subnet and host) addresses. IP Intelligence has eleven blacklist categories built in (identifying botnets, DoS sources, address-concealing proxies, and so forth). Not every class of untrustworthy IP addresses is represented by a built in category. You may add more local blacklist categories. For example, you might add a local category to blacklist unallocated public addresses (“bogons”).

To build an IP Intelligence policy you select blacklist categories of concern, specifying for each the action—Drop, Accept, or “Use Policy Default”—to take when a packet matches that category. It is best to choose “Use Policy Default”, which takes whatever the *policy’s* “Default Action” (Drop or Accept) happens to be *at the time the packet matches the category*. The order of blacklist categories in a policy does not matter. When a packet matches several categories it will be discarded if the action for any one of them is Drop (i.e., Drop trumps Accept). (The main use of action Accept is to log the receipt of packets which match some category.)

You may simulate "staging" an IP Intelligence policy as follows: Set the policy’s Default Action to Accept and enable the two Default Log Actions. Set the Action for each blacklist category to "Use Policy Default. Finally, attach the policy to a traffic-handling object to log which packets the policy matches without actually blocking any traffic. When you are satisfied with the policy’s criteria change its Default Action to Drop.

Setting the action for a blacklist category to Accept is not the same as defining a *whitelist*. There is only one whitelist per policy and it does not use categories. An IP Intelligence policy will Accept any packet that matches a whitelist address, even if it also matches a blacklist category (and regardless of any blacklist category action). You use blacklist categories to recognize and exclude connection attempts from malicious actors or compromised networks (and sometimes just nuisances, like competitors trying to probe your online catalog). You use whitelists to ensure that connections from trusted correspondents are welcomed even when their addresses inadvertently match a blacklist category (probably because some other customer of their ISP is misbehaving).

There are two ways to add local blacklist categories but only one way to delete obsolete categories. You may add and delete categories by editing the BIG-IP configuration (using TMSH or the management GUI at Security / Network Firewall / IP Intelligence / Blacklist Categories). AFM automatically adds new local blacklist categories specified in address-list data (details below) but never deletes any categories.

F5 supplies address data for the built-in blacklist categories on a near-realtime basis by subscription. The addresses are updated frequently in response to activity on the Internet. For instance, when someone on a regional ISP network starts firing DoS attacks, that will quickly be noticed and the afflicted network will be added to the relevant IP Intelligence category. To populate local blacklist categories or policy whitelists with addresses you must configure one or more *Feed Lists*. Each Feed List periodically fetches address data from HTTP(S) or FTP servers you specify.

IP Intelligence policies attached to global, route-domain, and virtual-server objects are enforced in that order. There is no default IP Intelligence policy and self IP’s can’t have separate policies (though the global policy protects them). You cannot override a blacklist Drop action in one policy with an Accept action or a whitelist match in a subsequent policy. Neither category nor whitelist addresses

may include Route Domain identifiers. (To match addresses in some route domain, attach an IP Intelligence policy to the route-domain object.)

Generally speaking you should apply a global "master" IP Intelligence policy to discard traffic matching most blacklist categories. Virtual-server policies are mainly useful when you wish to exclude public proxy users or local nuisances from certain services only. In that case, omit the "proxy" or "nuisance" blacklist category from the master policy but add it to a secondary policy, then attach the secondary policy to specific virtual servers.

If you subscribe to the F5 IP Intelligence service you should Drop incoming packets that match any malice-associated blacklist category. You might ignore the category "Cloud Provider Networks" since many benign connections originate from such addresses. The "Proxy" category also deserves special consideration. Some bad actors use proxies to evade well-deserved shunning of their real addresses, but legitimate actors may also use proxies (for instance, to dodge political controls on Internet usage in some countries).

It is wise to whitelist the public addresses of business partners with whom you maintain machine-to-machine links (EDI partners, for example). Don't forget to include the Feed List(s) for your whitelist(s) in any IP Intelligence policy which guards a resource any business partner needs to access. The way IP Intelligence works, a whitelist match in a global policy will not avert or bypass a blacklist-category match in a virtual-server policy (i.e., there is no IP Intelligence action equivalent to the network firewall action "Accept Decisively"). Always whitelist remote VPN endpoint addresses. Also whitelist any public IP addresses you control (your IPv6 PI prefix, for example).

Once you attach a global IP Intelligence policy you may test how it judges some address by issuing a command of the form:

```
tmsh show /security ip-intelligence info address xxxx
```

where xxxx is something like 203.0.113.9 or 2001:db8:cc0f:2152::9. To test a route-domain or virtual-server IP Intelligence policy append "virtual server VVV" or "route domain /Common/DDD" to the command.

How To Create An IP Intelligence Policy

In TMOS 11.6 the procedure to create an IP Intelligence policy with a whitelist is not perfectly intuitive. Here is the recommended approach:

First identify the web or FTP server from which a Feed List will retrieve the policy's whitelist addresses. Place a whitelist address-list on the server and note the URL needed to fetch it. (The address-list format is explained below. If you lack a suitable server, *Appendix B: Storing IP Intelligence Address Lists in BIG-IP Data Groups on page 62* shows how to supply addresses to Feed Lists from BIG-IP data groups.) Next, create an IP Intelligence Feed List at Security / Network Firewall / IP Intelligence / Feed Lists and add a Feed URL with "List Type" set to "White List" using the URL of your whitelist address list. Set "Poll Interval" to 600 seconds. (To merge whitelist addresses from several sources, add more Feed URL's with List Type set to White List.)

If the policy will include local blacklist categories (or if you plan to add addresses from non-F5 sources to built in categories), add one or more Feed URL's for blacklist addresses to the Feed List. Set each blacklist Feed URL's "List Type" to "Black List" and select the "Blacklist Category" which addresses from that Feed URL should populate by default (read down for details). You may wish to create extra Feed Lists to segregate whitelist and blacklist Feed URL's. Set each Feed URL's Poll Interval to the largest value consonant with the source data's update schedule.

Next, create the IP Intelligence policy at Security / Network Firewall / IP Intelligence / Policies. Select the Feed List(s) which will retrieve whitelist addresses and populate or augment the blacklist categories you will use in the policy. If you want to use a whitelist in an IP Intelligence policy you must select at least one Feed List. If you would like to "stage" the policy, set Default Action to Accept (otherwise leave it set to Drop). For each blacklist category you wish to filter in this policy: select the category name; leave the per-category Action and Log choices as "Use Policy Default"; click Add. (Note that to use a local category in a policy you must also select Feed List(s) to populate it. Otherwise the category will be empty and no packets will match it!) Finally, click Finished.

(Every address whitelisted by some Feed URL Z will be added to the whitelist of any policy which selects a Feed List that includes Z.)

IP Intelligence Address Data Format

AFM expects Feed URL's to retrieve address lists in Comma-Separated Value (CSV) format. Each record (line) must have an IP address in the first field, then optionally: a prefix/subnet mask-length in the second field; a whitelist indicator in the third field; and a blacklist category identifier in the fourth field. Any or all fields after the first may be empty. An empty mask-length means the address identifies a single host.

The third and fourth fields (whitelist indicator and blacklist category name) are interpreted as follows: first, if the third field is empty a value of “wl” or “bl” is presumed: “wl” if the Feed URL’s “List Type” is “White List” and “bl” otherwise.

Then, if the third field’s value is “wl” or “whitelist” the address is whitelisted and the fourth field is ignored. If the third field’s value is “bl” or “blacklist” the address is added to the blacklist category named in the fourth field, or if that field is empty, to the Feed URL’s Blacklist Category (any other value in the third field makes AFM ignore the whole line). When the third field’s value is “bl” and the blacklist category named in the fourth field does not exist, AFM creates that blacklist category automatically up to a limit of 62 categories.

Feed URL’s can fetch mixed lists of whitelist and blacklist-category addresses but it is better to separate the two types of addresses. When you expect a Feed URL to retrieve mainly whitelist (or blacklist) addresses, set its List Type to White List (Black List). That will help people understand the BIG-IP configuration even if the addresses actually come with whitelist indicators. However, when you expect a Feed URL to fetch mixed addresses set its List Type to Black List. That won’t affect the way address records with whitelist indicators are processed but will ensure that addresses lacking whitelist indicators are not whitelisted by default. It is much more dangerous to whitelist an address by mistake than to blacklist it.

Denial-of-Service (and DNS) Protection

This section provides additional information about DoS and DNS protection.

The Denial-of-Service Landscape

AFM DoS Protection mitigates two main types of DoS (DDoS) attacks: processing-logic attacks and resource-exhaustion attacks (the types do overlap).

Processing-logic attacks involve sending packets or messages (attacks are possible at every protocol layer) which are malformed in some way (sometimes in a way which only matters to a few possible recipients). Ideally any machine receiving a malformed message would simply reject it but in practice many recipients react badly—some even crash. In the worst case recipients react by sending replies which the attacker can direct to other victims by address spoofing.

Resource-exhaustion attacks involve sending well-formed packets or messages which ask the recipient to expend some resources. The more packets the attacker sends the more resources are wasted, leaving less for legitimate correspondents. These attacks may be subtle or crude. The well-known SYN-flood attack is fairly subtle. Only a few thousand packets are required to disrupt a vulnerable device because each provokes a big waste of memory. So-called volumetric attacks are cruder: the attacker just sends (or uses an amplification attack to make lesser victims send) very large numbers of packets. Indeed, the biggest DoS attacks commonly aim to exhaust the victim's ISP link capacity. Under such a DoS attack it doesn't matter how efficiently the victim's firewall recognizes and discards malicious packets—they crowd-out legitimate packets upstream.

AFM DoS Protection inspects packets and messages. It discards those which are malformed then recognizes and rate-limits those which are likely subtle resource-exhaustion attacks. DoS Protection also rate-limits volumetric attacks. LTM and AFM also implement specialized defenses to some attacks of the latter kind—for example, using SYN cookies to mitigate SYN floods.

The only defense to a really big DoS attack is *cloud scrubbing*. Cloud scrubbing means diverting traffic addressed to the victim to a facility with enormous ISP bandwidth and firewall capacity which "scrubs" malicious packets from the datastream then sends the residuum of legitimate packets on to the victim. F5 offers a cloud-scrubbing service called *F5 Silverline DDoS Protection (Silverline)*. If your organization might be subjected to a volumetric DoS attack you should consider a Silverline on-demand subscription. You may configure your BIG-IP data center firewall to recognize DoS attacks and activate Silverline cloud scrubbing when needed. That way you minimize network latency in the normal case but enjoy massive defensive capacity when required.

The majority of DoS attacks are not so big. In fact, almost all organizations log a constant stream of petty DoS attacks as malicious actors probe target networks more or less randomly. Bad actors are always looking for minor victims they can exploit to mount amplification attacks against other targets. Note that AFM IP Intelligence and network firewall provide critical defenses against DoS attacks: IP Intelligence empowers you to discard packets from malicious senders without wasting system resources trying to analyze them. The network firewall discards packets with bogus addresses—nearly all of them are attacks.

Understanding AFM DoS Protection

The first element of AFM DoS Protection is called (somewhat obscurely) *Device Configuration*. Device Configuration applies a global DoS-protection policy to all traffic through the BIG-IP system except management traffic. It recognizes malformed (e.g., Christmas-tree) and otherwise suspicious packets that match patterns called "DoS vectors." It also analyzes the arrival rates of various packet types to distinguish likely volumetric DoS attacks from normal traffic. Log messages from DoS Device Configuration alert you to DoS attacks and help you tune policy criteria to mitigate them.

DoS Device Configuration simply discards most malformed packets (at wire speed with F5 BIG-IP hardware acceleration). It accepts merely suspicious packets for further processing unless it sees too many of them in a short time, in which case it rate-limits similar packets until the presumed attack subsides.

How many is too many? The answer depends on the characteristics of your network and applications. You should tune the policy (Security > DoS Protection > Device Configuration) of quantity and rate-of-change thresholds for detection of various DoS vectors. The default criteria represent F5's best judgment for a "starter policy," which is essential because the DoS Device Configuration policy is always enforced. (Note that the starter policy imposes no rate limit at all on some packet types (UDP, for example). You are encouraged to make adjustments.)

Some applications are more sensitive than others to DoS attacks of various sorts. You may wish to protect a sensitive application with a "tight" policy, but if you tighten the global DoS Device Configuration policy too much it may interfere with traffic to other, less sensitive or simply busier applications. To avoid that problem you can utilize the second element of AFM DoS Protection, the *DoS Protection Profile*.

Each DoS Protection Profile defines a DoS-protection policy for some application or service. DoS Protection Profiles include general criteria and special criteria for DNS and SIP traffic. To enforce a DoS Protection Profile you must attach it to a virtual server. The quantity and rate-of-change thresholds for various vectors are then applied to detect and mitigate DoS attacks in traffic through that virtual server specifically. (DoS Protection Profiles don't see obviously-malformed packets because those are discarded early by DoS Device Configuration or by LTM, which implements many network protocols very exactly.)

When you protect specific applications and services with Dos Protection Profiles you may tune DoS Device Configuration policy to the overall contours of traffic through your network.

This Deployment Guide cannot offer a better example of DoS Protection policy than the default DoS Device Configuration supplied with BIG-IP AFM. Review the *AFM DoS Protection documentation* for additional insight into this important BIG-IP AFM feature.

Protecting DNS

Most attacks on DNS can be classified as DoS attacks (though intrusive attacks are possible; see *CVE-2001-0010* for an example). You may use AFM to mitigate many DoS threats by discarding packets with bogus source addresses (network firewall, IP Intelligence), rate-limiting malformed packets, bad queries, and query floods (AFM DoS Protection), and filtering out query types your servers don't support (DNS Protocol Security). However, many adversaries will try to overwhelm your DNS with a huge volume of superficially-valid queries. To distinguish malicious queries from legitimate ones you must process them, incurring the exact cost you wish to avoid. You cannot safely rate-limit queries because that facilitates cache-poisoning attacks against legitimate clients. The most effective mitigation measure is simply to deploy an ultra-high-performance DNS intermediate server: *F5 BIG-IP DNS Express*.

DNS Express is part of the *F5 BIG-IP Global Traffic Management and DNS Services module (GTM)*. GTM is a remarkably useful product which you may configure with AFM and LTM to achieve comprehensive DNS security. However, the details are beyond the scope of this Deployment Guide.

Due to the variety of DNS architectures implemented by F5 customers this document cannot provide definitive guidance on deploying AFM with DNS. Basically, if you expose nameservers to the public Internet, you should admit queries via LTM virtual servers with DNS Profiles, let your AFM global network-firewall and IP Intelligence policies scrub off packets with bad source addresses, *tune an AFM DoS Protection Profile to your DNS traffic and attach it to your DNS virtual servers*, and if appropriate, *configure and attach a DNS Protocol Security Profile* as well.

Troubleshooting AFM

The chief symptom of a firewall configuration error is lack of connectivity to some resource. That is also the chief symptom of many another infrastructure or resource (server) problem. General network troubleshooting is beyond the scope of this document. The advice which follows assumes you have good reason to suspect a firewall problem.

First check if BIG-IP service monitors can talk to the resource in question. Since they initiate flows from the BIG-IP itself they are never blocked by firewall policy. You may also try to reach the resource from the BIG-IP command line. If a monitor cannot talk to a resource, fix that before digging into firewall configuration.

If you suspect a non-firewall problem but your colleagues are skeptical, you might hazard a pinhole rule for service proof. Note the address of a test client plus the address and port of the target resource. Find the network firewall policy enforced on the global object. Prepend a rule of the form "src clientIP:any dst tgtIP:tgtPort vlan any action Accept_Decisively log Enabled" to that policy (temporarily, of course). Also whitelist the test client's address in any IP Intelligence policy along the path to the resource. Test connectivity. If the test client still cannot talk to the resource, that points toward a non-firewall issue. (If your pinhole rule alleviates the problem, brag that you have narrowed things down a lot, then extend your analysis.)

Examine the logs. Look for logs matching both the client and resource addresses. In many cases log messages will indicate which firewall policies and rules (if any) are interfering with access.

If AFM is running in Firewall mode, check that the traffic-handling objects in the path to the resource have suitably-permissive policies enforced on them. In some situations it pays to move a policy you suspect is causing trouble to *staging* and temporarily *enforce* a policy like /Common/accept-all while you analyze the logs.

If the complaint involves intermittent connectivity, slowness, or excessive retransmissions when accessing a popular resource, check whether a DoS Protection policy is rate-limiting traffic to the resource. If not, consider the possibilities of route-flapping or some-such elsewhere in the network. Also consider load-balancing problems, like a pool member which responds to the service monitor but not to actual application clients.

You may test the network-firewall policy hierarchy from the TMSH command line. Note the client's source address (and port, if possible—many connections are initiated from random ports above 1023 so you may have to choose one yourself; 9999 is easy to type). Note the address and port of the target resource (as seen by the client; i.e., a virtual-server address and port in many cases). Note as well the protocol (typically TCP or UDP) and the VLAN (like /Common/internal) from which client packets reach the BIG-IP. Then invoke a command of this form (line breaks inserted here for readability—don't type them into the BIG-IP):

```
tmsh show /security firewall matching rule
  source-addr Q.R.S.T source-port NNN
  dest-addr W.X.Y.Z dest-port MMM
  protocol PPP vlan LLL
```

You will recall the existence of a similar command for checking IP Intelligence policy results:

```
tmsh show /security ip-intelligence info address W.X.Y.Z
```

where only the client's address matters. To test a route-domain or virtual-server IP Intelligence policy append "virtual server VVV" or "route domain /Common/DDD" to that command.

To review the current IP address translations for a symbolic (FQDN) address in a firewall rule enter a command of this form (replace xxxx with the FQDN like "example.net" or with "all"):

```
tmsh show /security firewall fqdn-info fqdn xxxx
```

If all the symbolic address translations are missing, check and ensure that you configured a management default route or specific route(s) to your DNS server(s).

Appendix B: Storing IP Intelligence Address Lists in BIG-IP Data Groups

The only way to populate addresses into IP Intelligence whitelists and blacklist categories in TMOS 11.6 is through Feed Lists. However, it is not always convenient to set up a web or FTP server for IP Intelligence Feed URLs to poll. A workaround is possible. The combination of virtual server and iRule shown below lets you make address lists available to Feed URLs from BIG-IP data groups. You may use the management GUI or TMSH to edit records in "internal" data groups (Local Traffic > iRules > Data Group List), or you can edit address lists elsewhere and upload them to "external" datagroups (System > File Management > Data Group File List).

BIG-IP data groups of type **Address** hold records with two fields, an IP address in CIDR notation and a string value. As detailed above in IP Intelligence Data Format, Feed URLs consume records with two to four fields. The data group address field fills the first two Feed URL fields exactly. The data group string value stores data for the remaining two fields plus tag information for the convenience of administrators.

Compose each address record like this:

```
address/mask-length := [+][tag][=category]
```

where an optional leading '+' indicates the address should be whitelisted, an optional tag names the address to help administrators maintain the list, and an optional category (preceded by '=') says to which blacklist category the address belongs. For example:

```
10.0.0.0/8          :=  +intranet
2001:db8:cc0f::/48  :=  Mythical Medical Ctr=shun
2001:db8:ffff::/48  :=  =bogons
```

Since all the data in the string value is optional, string values may be empty. If the first character of the string value is '+' the third field of the Feed URL record will be "wl". If present, the category will fill the fourth field of the Feed URL record. The tag will not appear in the Feed URL record.

To serve address lists from data groups to multiple BIG-IP's (and/or other clients) put your datagroup-to-address-list (D2AL) virtual server in traffic-group-1 and give it an intranet (locally-routeable) address. If all of your BIG-IP's belong to one Device Group you may create a D2AL virtual server on each BIG-IP in traffic-group none and rely on config-sync to replicate your address-list data groups.

This example shows the second approach (one D2AL virtual server on each BIG-IP at a non-routeable address).

Create the iRule on the following page to format Data Group records for Feed List Feed URLs.

iRules (*Navigate to Local Traffic > iRules*)

Name Give the unique name. We use **afm datagroup feedurl**

Definition Copy and paste the following iRule, omitting the line numbers:

```
# In TMOS 11.6, IP Intelligence blacklist and whitelist addrs
# may be updated by IPI Feed Lists but not from TMSH or TMGUI.
# It is easy, though, to edit an internal Address datagroup...
#
# Create at least one "Address" datagroup (internal or external;
# edit internals from Local Traffic/iRules/Data Group List) to
# list IP's and (optional) values.  E.g., /Common/graylist.
# When given, values are "[+][TAG][=CAT]" where optional TAG is
# address owner and optional CAT is IPI category.  Optional '+'
# indicates address should be whitelisted.  Examples:
#   10.0.0.0/8      :=  "+intranet"
#   203.0.113.0/24  :=  "example.com=phishing"
# Create a virtual server with HTTP Profile on an IP reachable
# by intended list users.  Attach this iRule.  No pool.  Set a
# Feed List source URL like "http://198.18.0.5/Common/graylist"
# with datagroup's full /Partition/name as URL path. This iRule
# reports datagroup records as CSV lines: "addr,masklen,W,CAT"
# where W or CAT may be empty.  For example:
#   10.0.0.0,8,wl,  |  203.0.113.0,24,,phishing
# URL queries can select records by TAG and/or CAT:
#   http://198.18.0.5/Common/graylist?t=example.com
#   http://198.18.0.5/Common/graylist?c=phishing

when HTTP_REQUEST {
 set dg [HTTP::path]
 if {([HTTP::method] ne "GET") || ![class exists $dg] ||
     ([class type $dg] ne "IP") || ![class size $dg]} {
  log local0.err "[IP::client_addr] bad query to ${dg}"
  HTTP::respond 404 ; return
 }
 set buf [set c [set t ""]]
 foreach {q} [split [HTTP::query] "&"] {
  foreach {n v} [split $q "="] {
   if {($n eq "t") || ($n eq "c")} { set $n $v }
  }
 }
 foreach {item} [class get $dg] {
  foreach {ip val} $item {
   set w ""
   if {[string range $val 0 0] eq "+"} {
    set w "wl" ; set val [string range $val 1 end]
   }
   foreach {addr len} [split $ip "/"] {
    foreach {tag cat} [split "${val}=" "="] { break }
    if {(($t ne "") && ($tag ne $t)) ||
        (($c ne "") && ($cat ne $c))} { break }
    append buf "${addr},${len},${w},${cat}\n"
   }
  }
 }
 set ct "text/plain; charset=UTF-8"
 HTTP::respond 200 content $buf Content-Type $ct
}
```

The next task is to add a virtual server that Feed URLs can poll.

Unless you choose to assign a routeable intranet address to this virtual server, create it separately on each BIG-IP device (big-s1 and big-s2 in our example) and move its virtual address to traffic-group none.

Virtual Servers (*Navigate to Local Traffic > Virtual Servers*)	
Name	**datagroup-feedurl-vs**
Destination Address	**198.18.0.5** (you could use an intranet address; see the note preceding this table)
Service Port	**80 (HTTP)**
HTTP Profile	**http**
VLAN and Tunnel Traffic	Select **Enabled On**, and then select only the **Internal** VLAN you created.
Source Address Translation	**None**
iRules	Enable the iRule you just created (**afm-datagroup-feedurl** in our example).
Default Pool	**None**

Modifying the Traffic Group

If the datagroup-feedurl-vs you just created has a non-routeable address, you must move its virtual address to traffic-group none. If using a routeable intranet address leave it in traffic-group-1.
If necessary, click **Local Traffic > Virtual Servers**. On the menu bar, click **Virtual Address List**. Click the IP address of the virtual server you just created (198.18.0.5 in our example). From the **Traffic Group** list, select **None**, and then click **Update**.

Creating blacklist categories and feed lists

The next task is to create blacklist categories and then Feed Lists to populate them. Use the following table for guidance.

Black List Categories (*Navigate to Security > Network Firewall > IP Intelligence > Black List Categories*)	
Black List Category Name	**shun**
Description	Public IP addresses of nuisances or competitors we want to keep out of our websites.

You will keep local blacklist and whitelist addresses in data groups. Internal data groups (as shown here) can hold a few thousand records, but are not convenient for more than fifty or so. Consider using external data groups instead. Just maintain your address lists with a text editor and upload them to the BIG IP system for distribution.

iRule Data Group (*Navigate to Local Traffic > iRules > Data Group List*)		
AFM Blacklist Data Group		
Name	**afm-blacklist**	
Type	**Address**	
Records > Address	***Address***	***Value***
	2001:db8:cc0f::/48	**Mythical Medical Ctr=shun**
AFM Whitelist Data Group		
Name	**afm-whitelist**	
Type	**Address**	
Records	***Address***	***Value***
	192.0.2.0/24	**Example.net (whitelist our own public addresses)**
	2001:db8:16d::/48	**Example.net (whitelist our own public addresses)**
	170.167.0.0/16	**Costco Wholesale**

Feed Lists (*Navigate to Security > Network Firewall > IP Intelligence > Feed Lists*)		
Local blacklist Feed List		
Name	**local-blacklist-feed**	
Description	local blacklist feed	
Feed List Properties: Feed URLs	***Name***	**local-blacklist**
	URL	**http://198.18.0.5/Common/afm-blacklist**
	List Type	**Black List**
	Blacklist Category	**shun**
	Poll Interval	**600**
Local whitelist Feed List		
Important Note	This is the white list feed list referenced in *Creating the Local whitelist feed on page 22. If you have already created that Feed List, simply edit the existing list to add the URL referenced below.*	
Name	**local-whitelist-feed**	
Description	Trusted addresses for our company and our business partners.	
Feed List Properties: Feed URLs	***Name***	**local-whitelist**
	URL	**http://198.18.0.5/Common/afm-whitelist**
	List Type	**White List**
	Blacklist Category	**spam_sources** (What you choose here does not matter; it can be anything)
	Poll Interval	**600**

Updating the IP Intelligence Policy

The final task is to update the IP Intelligence policy you created in *Creating an IP Intelligence policy on page 22* (in this example, the one on the global object) to use the objects you just created.

IP Intelligence Policy (*Navigate to Security > Network Firewall > IP Intelligence > Policies*)		
Important Note	This is the IP Intelligence policy referenced in *Creating an IP Intelligence policy on page 22. If you have already created that policy, simply edit the existing policy to add the local blacklist feed and shun category you just created.*	
Name	**global-IPI-policy**	
Description	Prevents connections from bogus and malicious source addresses	
IP Intelligence Properties	***Feed Lists***	Use the Add arrows (<<) to move the four Feed Lists you created (bogons feed, DROP_EDROP feed, local-whitelist-feed, and **local-blacklist-feed**) to the **Selected** box. Note that the local-blacklist-feed is the one you just created.
	Default Action	**Drop**
	Default Log Actions	Check the **Log Blacklist Category Matches** and **Log Whitelist Overrides** boxes.
	Blacklist Matching Policy	From the **Blacklist Category** list, select each of the following categories and then click **Add** (leave the Action and both log lists at the default (**Use Policy Default**)
		bogons
		botnets
		denial_of_service
		DROP_EDROP
		illegal_websites
		infected_sources
		phishing
		proxy
		shun *(Note this is the category you just created)*
		scanners
		spam_sources
		web_attacks
		windows_exploits

Appendix C: Securing AFM Domain Name Resolution

In TMOS version 12.0 the BIG-IP AFM network firewall resolves symbolic (FQDN) address domain names to IP addresses using a BIG-IP Network DNS Resolver. A TMOS 12.0 Network DNS Resolver sends DNS queries directly to Internet nameservers by default, but you can make it query local nameservers by configuring a suitable Forward Zone. TMOS 12.0 Network DNS Resolvers do not implement DNSSEC so they are unable to validate responses to their queries cryptographically.

To reduce the danger that an adversary will subvert your AFM network firewall rules by interfering with Internet DNS resolution, you should use a Network DNS Resolver Forward Zone to direct AFM's DNS queries to a BIG-IP DNS Cache Validating Resolver or to local nameservers which use DNSSEC to cryptographically-validate DNS responses.

If your BIG-IP device has a suitable license, you may configure a BIG-IP DNS Cache Validating Resolver to supply cryptographically-validated DNS responses to AFM's Network DNS Resolver. This approach is the most secure since it excludes the local network from your firewall attack surface (to the extent that symbolic addresses in AFM network firewall rules refer to cryptographically-secured (DNSSEC) DNS resources).

Otherwise, if you have DNSSEC-enabled local nameservers and can provide secure network paths between them and your BIG-IP device(s) you may obtain a pretty good level of security by directing Network DNS Resolver queries to those nameservers. Since responses to those queries will not be cryptographically-secured in transit, your firewall attack surface will still include network paths to your local nameservers.

We do not recommend allowing AFM's Network DNS Resolver to send queries directly to the Internet.

Configuring a BIG-IP DNS Cache Validating Resolver

If your BIG-IP device has a suitable software license, follow these instructions to configure a DNS Cache Validating Resolver. Otherwise, skip to *Using Secure Local Nameservers on page 69*.

On the day you configure the DNS Cache Validating Resolver you must obtain some fresh DNSSEC public keys. You will need the long-term DNS root key-signing-key(s) (KSK's), the long-term dlv.isc.org KSK(s), and optionally the long-term DLV KSK(s) for your local secure DNS zone(s). Use the dig(1) command from the BIG-IP Advanced Shell command line (or from another workstation) to obtain key-signing-key RR's as follows (use the virtual console or an SSH client such as putty to access the BIG-IP Advanced Shell):

First obtain the long-term root KSK RR(s):

```
dig -t dnskey . | grep 'IN.DNSKEY.257'
```

You will see something like this example:

```
.                       107052  IN      DNSKEY  257 3 8
AwEAAagAIKlVZrpC6Ia7gEzahOR+9W29euxhJhVVLOyQbSEW0O8gcCjF FVQUTf6v58fLjwBd0YI0EzrAcQqBGCzh/RStIoO8g0NfnfL2MTJRkxoX
bfDaUeVPQuYEhg37NZWAJQ9VnMVDxP/VHL496M/QZxkjf5/Efucp2gaD X6RS6CXpoY68LsvPVjR0ZSwzz1apAzvN9dlzEheX7ICJBBtuA6G3LQpz
W5hOA2hzCTMjJPJ8LbqF6dsV6DoBQzgul0sGIcGOYl7OyQdXfZ57relS Qageu+ipAdTTJ25AsRTAoub8ONGcLmqrAmRLKBP1dfwhYB4N7knNnulq
QxA+Uk1ihz0=
```

That is the long-term root key-signing-key RR. If you see two RR's that means the root KSK will be replaced soon. Copy and save the key RR(s) to use momentarily.

Note: the value shown in this document is just an example. Use the dig(1) command to obtain fresh key RR(s).

Next obtain the long-term dlv.isc.org key RR:

```
dig -t dnskey dlv.isc.org. | grep 'IN.DNSKEY.257'
```

You will see something like this example:

```
dlv.isc.org.            7200    IN      DNSKEY  257 3 5
BEAAAAPHMu/5onzrEE7z1egmhg/WPO0+juoZrW3euWEn4MxDCE1+lLy2 brhQv5rN32RKtMzX6Mj70jdzeND4XknW58dnJNPCxn8+jAGl2FZLK8t+
1uq4W+nnA3qO2+DL+k6BD4mewMLbIYFwe0PG73Te9fZ2kJb56dhgMde5 ymX4BI/oQ+cAK50/xvJv00Frf8kw6ucMTwFlgPe+jnGxPPEmHAte/URk
Y62ZfkLoBAADLHQ9IrS2tryAe7mbBZVcOwIeU/Rw/mRx/vwwMCTgNboM QKtUdvNXDrYJDSHZws3xiRXF1Rf+al9UmZfSav/4NWLKjHzpT59k/VSt
TDN0YUuWrBNh
```

That is the long-term **dlv.isc.org** KSK RR. If you see two RR's that means the KSK will be replaced soon. Copy and save the key RR(s) to use momentarily.

Note: the value shown in this document is just an example. Use the dig(1) command to obtain fresh key RR(s).

If your organization has any local DNSSEC-secured zones which use DLV KSK's not registered with dlv.isc.org, obtain RR's for those keys from your DNS administrator.

Configuring the BIG-IP objects

Use the following guidance to configure the required objects on the BIG-IP system. Each table contains only the required settings for each configuration object. Unless otherwise stated, settings not listed in the table can be configured as applicable for your implementation. For specific instructions on configuring the objects, see the inline help or product documentation.

Creating a new Cache

Use the following table to create a new Cache with a type of Validating Resolver.

Cache (*Navigate to DNS > Caches > Cache List > Create*)	
Name	Type a unique name, such as **fw-secure-resolver**.
Resolver Type	**Validating Resolver**

Modifying the Cache to add a Trust Anchor

Next, you open the Validating Resolver object you just created and use the tabs to edit key properties. The first task is to add each long-term root key-signing-key RR to Trust Anchors:

Cache: Trust Anchor (*Click the Cache you just created > Trust Anchors (on the Menu bar) > Add*)	
Trust Anchor	Paste one entire long-term root key-signing-key RR which you obtained earlier using dig(1). If you have more than one long-term root key-signing-key RR, click the Add button and insert the additional key.

Modifying the Cache to add a DLV Anchor

Next, add each long-term dlv.isc.org KSK RR to DLV Anchors:

Cache: DLV Anchor (*Click DLV Anchor (on the Menu bar) > Add*)	
DLV Anchor	Paste one entire long-term dlv.isc.org key-signing-key RR which you obtained earlier using dig(1). If you have more than one long-term dlv.isc.org key-signing-key RR, click the Add button and insert the additional key.
Repeat to add any long-term DLV KSK RR's for your local secure zones to DLV Anchors.	

Important *With TMOS 12.0 you must maintain Trust and DLV Anchors for DNS Cache Validating Resolvers manually. When the key-signing-key for the DNS root, the dlv.isc.org zone, or any local secure zone expires you must remove it and insert the replacement KSK RR.*

If your BIG-IP system does not have a route to send DNS queries to the Internet

If your BIG-IP device does not have a route to send DNS queries directly to the Internet, you may configure your DNS Cache Validating Resolver to send all queries to local forwarding nameservers. You must add a special Forward Zone as described in this section.

Note: *Do not use a Forward Zone to make your DNS Cache Validating Resolver send all DNS queries to local nameservers unless necessary. Replace the example nameserver addresses here with values appropriate in your environment.*

In the following table, we show an example with two forwarding nameservers: 192.168.193.5 and 192.168.193.7.

Cache: Forward Zones (*Click Forward Zones (on the Menu bar) > Add*)	
Name	**.** [that is a single dot '.' by itself]
Nameservers	**Address: 192.168.193.5** **Service Port: 53** Click Add to include more Nameservers. In our example, we add **192.168.193.7**

Additionally, if you have any local zones which should be resolved by queries to specific local servers, you may add corresponding Forward Zones. For example, you might have a zone **corp.example.net** which is only resolved by a local nameservers 192.168.193.5 and 192.168.193.7.

Note: *The following is just an example. Do not add any local Forward Zones unless you need them. Replace the example data here with values appropriate in your environment.*

Cache: Forward Zones (*Click Forward Zones (on the Menu bar) > Add*)	
Name	**corp.example.net** (our example)
Nameservers	**Address:** **192.168.193.5** **Service Port:** **53** Click Add to include more Nameservers. In our example, we add **192.168.193.7**

Creating a DNS profile

The next task is to create a DNS profile. Use the following table for guidance on configuring the profile.

Cache: Forward Zones (*Click Forward Zones (on the Menu bar) > Add*)	
Name	Type a unique name, such as **fw-secure-dns-pfl**
Parent Profile	**dns**
DNS Cache	**Enabled**
DNS Cache Name	Name of the cache you created **(fw-secure-resolver** in our example)
Unhandled Query Actions	**Reject**
Use BIND Server on BIG-IP	**Disabled**
Process Recursion Desired	**Enabled**

Creating the virtual servers

The next task is to create the virtual servers on the BIG-IP system. To avoid IP address conflicts you may use a special "class E" IP address valid only inside the BIG-IP system for these virtual servers. We suggest the one shown in the example.

Virtual Servers (*Main tab > Local Traffic > Virtual Servers*)	
UDP virtual server	
Name	Type a unique name, such as **fw-secure-dns-udp-vs**.
Destination Address	Type the IP address for this virtual server, such as **245.0.0.53/0**.
Service Port	**53**
Protocol	**UDP**
DNS Profile	Name of the DNS profile you created **(fw-secure-dns-pfl** in our example)
VLANs and Tunnels	**All VLANs and Tunnels**
Source Address Translation	None
TCP virtual server	
Name	Type a unique name, such as **fw-secure-dns-tcp-vs**.
Destination Address	Type the IP address for this virtual server, such as **245.0.0.53/0**.
Service Port	**53**
Protocol	**TCP**
DNS Profile	Name of the DNS profile you created **(fw-secure-dns-pfl** in our example)
VLANs and Tunnels	**All VLANs and Tunnels**
Source Address Translation	None

Adding a network firewall policy to each virtual server

The next task is to attach an AFM network firewall policy to the virtual servers you just created.

Virtual Server (*Navigate to Local Traffic > Virtual Servers > your-udp-virtual-server > Security (Menu bar) > Policies*)	
Navigation help	Click **Local Traffic > Virtual Server**. Click the UDP virtual server you created. On the Menu bar, click **Security > Policies**.
Network Firewall: Enforcement	Select **Enabled**. From the **Policy** list that appears, select the Accept All policy you created in *Creating the Network Firewall Policies on page 18* (**accept-all** in our example).
Network Firewall: Staging	Select **Disabled**. .
IP Intelligence	Leave IP Intelligence set to **Disabled**.
DoS Protection Profile	Leave DoS Protection Profile set to **Disabled**.
Log Profile	**None**
Repeat this table for the TCP virtual server you created	

When you configure the Network DNS Resolver Forward Zone in the following section, the IP address for your Forward Zone nameserver will be the address you used for the DNS Cache Validating Resolver virtual servers, such as 245.0.0.53 in our example.

Using Secure Local Nameservers

If your BIG-IP device's license does not include DNS Cache Validating Resolver capability, you may use secure local nameservers to improve the security of AFM symbolic-address resolution. Ensure that your local nameservers are configured to handle recursive queries from clients and to validate results using DNSSEC. Ensure the network paths between your BIG-IP AFM devices and your nameservers are secure.

Configuring AFM to Resolve Symbolic (FQDN) Addresses in Network Firewall Rules

The next task is to create a Network DNS Resolver. In the following table, the small Cache Size value is appropriate because this object will use external caches.

DNS Resolver (*Navigate to Network > DNS Resolvers > Create*)	
Name	Type a unique name, such as **fw-net-resolver**.
Route Domain Name	**0**
Cache Size	**131072**
Use IPv6	Uncheck this option unless both of the following are true: you are using local secure nameservers (not a DNS Cache Validating Resolver), and at least one of your local secure nameservers has an IPv6 address.

Adding a Forward Zone to the DNS resolver

The next task is to open the resolver you just created and add a single Forward Zone. The nameserver IP address(es) for that Forward Zone will either be the one you assigned to the virtual servers for any DNS Cache Validating Resolver you created (e.g., 245.0.0.53) or the IP addresses of your secure local nameservers (replace the example addresses 192.168.193.5/7 with the IP addresses of your nameservers).

Forward Zone (*Navigate to Network > DNS Resolvers > name of the DNS resolver you just created > Forward Zone (on the Menu bar)* > Add	
Name	**.** [that is a single dot '.' by itself]
Nameservers	*When using a DNS Cache validating Resolver (actual IP)* **Address:** **245.0.0.53** **Service Port:** **53** *When using secure local nameservers (example IP's)* **Address:** **192.168.193.5** **Service Port:** **53** **Address:** **192.168.193.7** **Service Port:** **53**

Next, you configure AFM to use your Network DNS Resolver (remember to click Update after making changes):

Network Firewall (*Navigate to Security > Options > Network Firewall*)	
Global Context	Select the DNS Resolver you just created (**fw-net-resolver** in our example).
Refresh Interval	**20** minutes.

If you use more than 200 different symbolic addresses in network firewall rules you may wish to lengthen the FQDN Resolver Refresh Interval. Longer refresh intervals reduce BIG-IP CPU utilization but increase the chance that IP addresses in network firewall rules will be stale. Consider the TTL values of your symbolic addresses when you adjust the refresh interval.

DNS Cache Validating Resolver Statistics

When you use a DNS Cache Validating Resolver you can check its performance. Navigate to **DNS / Caches / Cache List / [name of your Validating Resolver, such as fw-secure-resolver]** then select the Statistics tab. Click the **View** link in the *Details* column of the relevant row to review detailed information about queries, cache performance, resolver traffic, and DNSSEC coverage.

Glossary of Terms

AAA Gateway	A device which enforces security policy very high in the OSI stack by authenticating users, authorizing their connections, and managing their access by proxying or filtering traffic, possibly through dynamic configuration of other infrastructure devices (from Authentication, Authorization, and Access Management)
ADC Mode	A BIG-IP AFM operating mode in which the implicit policy for every listener (virtual server or self IP) accepts whatever traffic reaches it (from "Application Delivery Controller," BIG-IP's fundamental job)
ALG	Application Layer Gateway—a traffic-handling object which supports a complex network protocol. A BIG-IP ALG for some protocol may create ephemeral listeners for secondary flows used in that protocol and may discard packets which don't conform to that protocol.
APT	Malware resident on a computer such as a virus, rootkit, or bot (from "Advanced Persistent Threat")
AUP	Acceptable Use Policy—an organization's rules for use of computing and Internet resources
Deep Packet Inspection	Examining packet payload, not just header and L2 context. Very sophisticated devices (e.g., BIG-IP) can examine payloads split across multiple packets in a flow
Denial-of-Service Protection	Component of BIG-IP AFM which recognizes and excludes network DoS attacks
DDoS	Distributed DoS
DLP	Data Loss Prevention—tool which scans network traffic looking for confidential data in payloads. May log or interrupt data transfers which lack administrative approval
DMZ	A small trust environment where devices that mediate between high- and low-trust environments are segregated (from "Demilitarized Zone")
DNSSEC	DNS Security — cryptographically ensures the integrity of DNS information using digital signatures.
DoS	Denial-of-Service (attack)
Dual-stack network	Carries IPv6 and IPv4
Edge firewall	North-South firewall directly connected to Internet
East-West (EW)	(Traffic) between portions of the intranet
FIPS	(US) Federal Information Processing Standard (published by NIST)
Firewall Mode	A BIG-IP AFM operating mode in which the implicit policy for every listener (virtual server or self IP) discards all traffic (so to accept traffic to a listener you must enforce a named policy on it)
Five-Tuple	Basic data from the IP packet header which firewall rules may match: IP protocol, source address and port, destination address and port. BIG-IP AFM firewall rules can match additional packet attributes
FQDN	A Fully-Qualified Domain Name (FQDN) is a DNS domain name (for example, www.f5.com.) with a Top Level Domain (TLD) name such as com or org as the rightmost component (rfc1034). Technically an FQDN (also called an "absolute" domain name) must end with a dot (.) but the trailing dot is almost always omitted.
HIPAA	US regulations for security of health data (from "Health Insurance Portability and Accountability Act")
HSTS	HTTP Strict Transport Security—a protocol extension to avert crypto downgrade attacks against secure websites (rfc6797)

ICAP	A protocol for integrating malware-detecting and DLP tools with network infrastructure devices (from "Internet Content Adaptation Protocol" rfc3507)
ICSA Labs	Firm which certifies computer security products
IDS/IPS	Intrusion Detection/Prevention System—tools that scan network traffic looking for signatures of malware and symptoms of malicious activity. Both log suspicious traffic, IPS tries to interrupt it as well
IF-MAP	A protocol for interrogating network configuration data, which BIG-IP uses to dynamically associate user identities with IP addresses (from "Interface for Metadata Access Points")
Implicit Policy	A dynamic BIG-IP AFM network firewall policy which applies when no named policy has been enforced on a traffic-handling object
IP Geolocation	Associating a public IP address with a geographic region or country
IP Intelligence	Component of BIG-IP AFM which categorizes remote IP addresses by types of traffic they source or sink
iRule™	Event-driven script used to extend the capabilities of BIG-IP
ISP	Internet Service Provider
Management Network	A portion of the intranet dedicated to system administration work. Generally a high-trust environment maintained and guarded with special care
Listener	A logical BIG-IP object which processes network traffic. The most common type of listener is a virtual server
Negative Security	A model for security policy which defines what is forbidden and implicitly permits everything else
Network Firewall	Component of BIG-IP AFM which filters network connections mainly (though not only) at L2–4
NGFW	Next-Generation Firewall—tool combines a zone-based firewall with an IPS and is intended mainly to control traffic originating from organizational intranet. Like web proxies NGFW's are used to enforce AUP
NIST	US National Institutes for Standards and Technology (a government agency)
North-South (NS)	(Traffic) between Internet and intranet
PCIDSS	Payment Card Industry Data Security Standards. Standards for firms handling card payment data
Pinhole Rule	A positive-model firewall rule to match and admit some narrowly-defined traffic
Positive Security	A model for security policy which defines what is permitted and implicitly forbids everything else
Private (IP) Address	An address meaningful only within an intranet or non-public inter-network. Packets bearing private addresses cannot be routed over the public Internet
Protocol Security	Component of BIG-IP AFM which enforces compliance with higher-level protocols like HTTP
Proxy Gateway	Device which mediates communications between other network correspondents while maintaining separate connections to each. By re-encapsulating (and possibly altering) data received on one connection for transmission on the other, proxy gateways can enable communication at one OSI layer between correspondents using different protocols at another layer
Public (IP) Address	Packets bearing public addresses may be routed over the public Internet
SIEM	Security Incident/Event Management system (typically accepts and analyzes log messages)
Symbolic Address	A long-lived name (such as an FQDN) for a network resource which may be translated into one or more short-lived numeric network addresses.

TMOS	F5 BIG-IP Traffic Management Operating System
TMSH	BIG-IP Traffic Management Shell; command-line interface to TMOS
Trust Environment	A logical portion of the network within which entities communicate freely
VPN	Virtual Private Network—a private network link over shared infrastructure. One commonly assures privacy by data encryption
Web Proxy	A type of proxy gateway used to control traffic (chiefly web but often some other types as well) originating from the intranet. Typically enforces AUP and detects malware in data from Internet. Often integrated with cache (for performance) and DLP

Document Revision History

Version	Description	Date
1.0	New deployment guide	07-06-2015
1.1	Added support for BIG-IP v12.0. The main change is that in version 11.x, an iRule is needed to implement HSTS. BIG-IP v12.0 and later includes support for HSTS (although the iRule will still work). HSTS support in v12 is enabled through an HTTP profile (detailed in *Configuring the BIG-IP system to send web application traffic into the DMZ on page 37).*	09-29-2015
1.2	- Added *Appendix C: Securing AFM Domain Name Resolution on page 66.* - Added a link to the F5 NIST SP800-53r4 deployment guide. - Added information about symbolic addresses to *AFM Network Firewall Rules on page 54*	02-05-2016

F5 Networks, Inc. 401 Elliott Avenue West, Seattle, WA 98119 888-882-4447 www.f5.com

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com