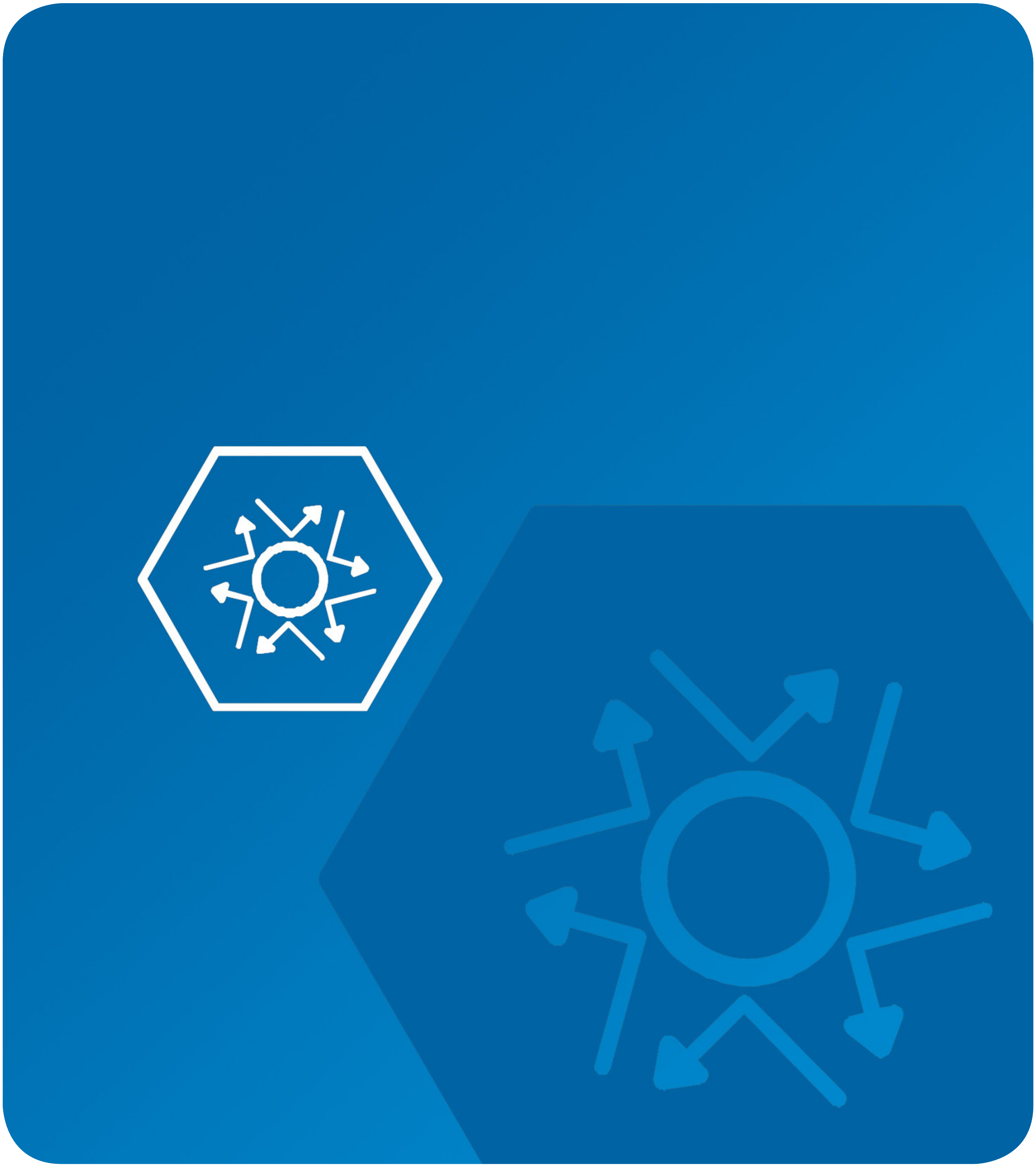
RECOMMENDED PRACTICES

F5 DDoS Protection
Volume 2



Contents

1 Concept

Distributed Denial-of-Service (DDoS) is a top concern for many organizations today, from high-profile financial industry brands to service providers. Experienced administrators know that F5 equipment is not only well-suited to mitigating DDoS attacks, but sometimes it is the only equipment that can mitigate certain types of DDoS. What many administrators do not know is that a complete DDoS mitigation solution can be achieved with a complement of F5 products.

A DDoS attack can be a stressful engagement where parts of the network will be unresponsive and equipment may be failing all around. That is not the time to be planning a defense—preparing your network applications during "peacetime" will go a long way to helping you mitigate a future attack.

This guide assumes that you have an F5 networking solution and an optional F5 security solution.

All configurations, commands, and platforms are assumed to be TMOS 11.6.0 unless otherwise stated.

Even though much of the technical information is specific to F5 equipment, some of the strategies (such as the use of SNAT pools to avoid port exhaustion) may apply to other vendors' devices as well.

2 DDoS-Resistant Architecture

It is possible to build an application delivery network that is DDoS-resistant by designing an enterprise architecture that addresses attacks both on- and off-premises. This complementary hybrid approach is the most cost-effective and provides the greatest amount of resistance to modern DDoS attacks.

2.1 F5's Recommended Architecture

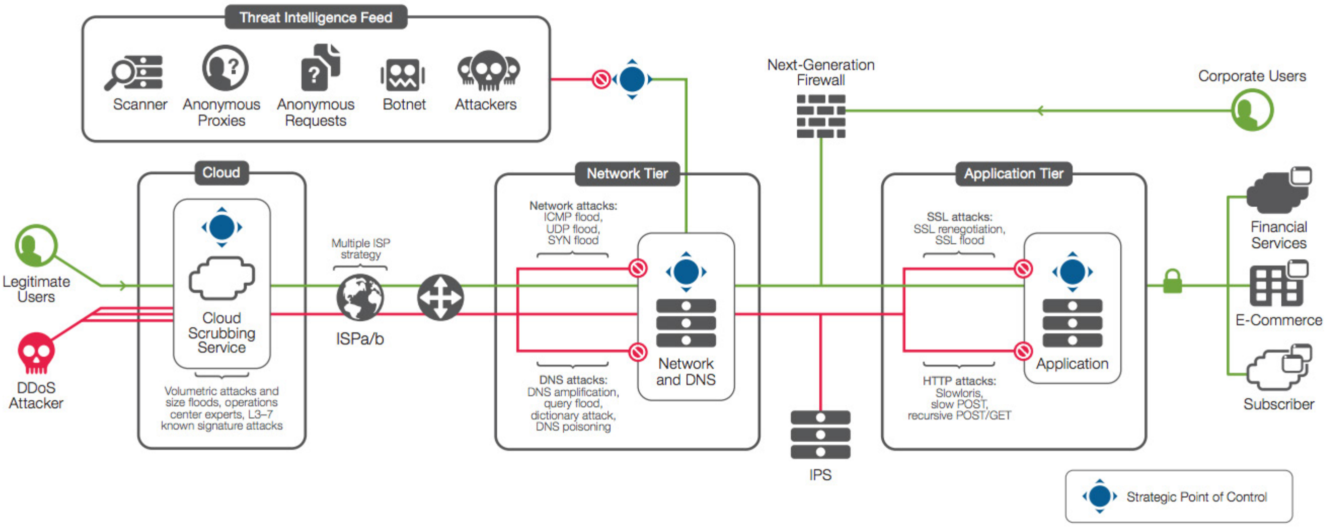

Threat Intelligence Feed
Scanner
Anonymous Proxies
Anonymous Requests
Botnet
Attackers
Next-Generation Firewall
Corporate Users
Cloud
Network Tier
Application Tier
Legitimate Users
DDoS Attacker
Cloud Scrubbing Service
Volumetric attacks and size floods, operations center experts, L3–7 known signature attacks
Multiple ISP strategy
ISPa/b
Network attacks: ICMP flood, UDP flood, SYN flood
DNS attacks: DNS amplification, query flood, dictionary attack, DNS poisoning
Network and DNS
IPS
SSL attacks: SSL renegotiation, SSL flood
HTTP attacks: Slowloris, slow POST, recursive POST/GET
Application
Financial Services
E-Commerce
Subscriber
Strategic Point of Control

Figure 1: Architectural diagram for DDoS protection

The many high-profile DDoS attacks of the previous three years have provided the motivation for organizations to integrate DDoS resilience into their network architecture. F5 has been working with organizations during these restructuring efforts. The collaboration has yielded a common architecture that resists the three main types of DDoS attacks: volumetric (layer 3), network (layer 4), and application (layer 7).

The architecture contains DDoS-resistant components in the cloud, at the network tier, and at the application tier. Most enterprise customers have these tiers already, and the question becomes tuning the configuration of the F5 devices in those tiers to be resilient to DDoS attacks.

Cloud Tier

A cloud-based DDoS scrubbing service is a critical component of any DDoS mitigation architecture. When an attacker is sending 50 Gbps of data at an organization's 1 Gbps ingress point, no amount of on-premises equipment is going to solve that problem. The cloud service, hosted either from a true public cloud or within the organization's bandwidth service provider, solves the problem by sorting out the obvious bad from the likely good. There are several cloud-based DDoS scrubbers to choose from, but F5 recommends its own Silverline.

Network Tier

The network firewall has been the keystone of perimeter security for a long time. However, many network firewalls are not resistant to DDoS attacks at all. In fact, many of the best-selling firewalls can be disabled with the simplest layer 4 attacks. Sheer throughput is not the answer if the firewall does not recognize and mitigate the attack. For a layer 3- and 4-based security control device, F5 recommends that architects choose a high-capacity, DDoS-aware network firewall. Specifically, architects should be looking to support millions (not thousands) of simultaneous connections and be able to repel SYN floods without affecting legitimate traffic.

Application Tier

The application tier is where incoming traffic is typically decrypted, inspected, and load-balanced to the back-end servers. The application tier can be strengthened against DDoS by tuning the configurations of the F5 LTM load-balancing module and the F5 ASM web application firewall. The application tier will be the home of some of the most sophisticated layer 7 defenses in the architecture, including tactical defenses like login-walls, URL rate-limits, and brute force CAPTCHA mitigations.

	Cloud	Network Defense	Application Defense	DNS
F5 Components	SilverLine DDoS Protection	BIG-IP AFM BIG-IP LTM	BIG-IP LTM BIG-IP ASM	BIG-IP GTM with DNS Express™
OSI Model	Layers 3 and 4	Layers 3 and 4	Layer 7	DNS
Capabilities	Volumetric scrubbing Traffic dashboarding	Network firewall Layer 4 load balancing IP blacklists	SSL termination Web application firewall Secondary load balancing	DNS resolution DNSSEC
Attacks Mitigated	Volumetric floods Amplification Protocol whitelisting	SYN floods ICMP floods Malformed packets TCP floods Known bad actors	Slowloris Slow POST Apache Killer RUDY/Keep Dead SSL attacks	UDP floods DNS floods NXDOMAIN floods DNSSEC attacks

2.2 Cloud Defense Recommended Practices

The most common component of a DDoS attack is the volumetric attack. A volumetric attack is an overwhelming amount of network traffic that clogs the “pipe” between the Internet and your datacenter or services. Volumetric attacks do not have to be particularly sophisticated to be effective; throwing multiple gigabits of junk UDP packets at a website that has only 1 gigabit of ingress capacity will be sufficient to deny service to the website. While volumetric attacks may not be sophisticated, sometimes they are used as a smokescreen to hide the real intent of the attack, which may be to get at sensitive corporate data.

F5’s Silverline DDoS Protection service provides a first line of defense against volumetric attacks. Silverline works by intercepting and absorbing the volumetric attack and “scrubbing” the malicious network traffic from it. It then sends only clean traffic to the target website. Before, during, and after the DDoS attack, an organization’s security team can work with the F5 Silverline Security Operation Center (SOC) staff to identify the critical traffic through the noise, enabling faster, more accurate mitigation in the future.

Once F5 Silverline is properly integrated as a solution, it will, of course, provide the heavy lifting against volumetric attacks, so the recommended practices for volumetric defense are for integrating the Silverline DDoS Protection service.

2.2.1 Engage with Volumetric Defense Service before an attack

The most important recommendation to any organization that is concerned about DDoS is to make an arrangement with a volumetric defense provider (like F5 Silverline DDoS Protection) before an attack takes place. This is for two reasons:

1. Nearly every volumetric defense service charges a significant premium if the subscriber is already under attack when the service is first engaged.

2. Integrating a volumetric defense service can be a non-trivial procedure even during peacetime. When networks are down and stress is high during an attack, it becomes much more challenging.

There are multiple volumetric defense services. Of course, F5 recommends engaging with the Silverline DDoS Protection service. You can work with your account manager or security partner to establish the engagement. Call 866-329-4253 (or, internationally, +1 (206) 272-7969) to get started.

2.2.2 Select defense routing option

As with other volumetric services, there are multiple ways to configure network interfaces to the Silverline DDoS service. You will very likely have to do a bit of homework to choose which is the right way for your organization (which is another reason to do this before an attack hits). Two of the most common methods are proxy mode and routed mode. With proxy mode, F5 Silverline terminates network traffic and proxies separate connections to your network. This allows F5 Silverline to perform advanced services such as Web Application Firewall protection. Routed mode transits network traffic through F5 Silverline to your network.

Work with the Silverline team to determine which configuration is the right one for your organization.

2.2.2.1 Proxy Mode: Simple Application Protection

For enterprises requiring protection for a subset of applications with minimal network changes, F5 Silverline DDoS protection can be deployed in a proxy configuration. This is useful to provide relatively quick mitigation by simple DNS changes. Once the proxy architecture is implemented, the client requests are directed to the Silverline service for mitigation. Scrubbed traffic is then proxied back to the enterprise application. The application response is given to the Silverline proxy to, in turn, service the client request. As requests are made on behalf of the client but sourced from F5's infrastructure, technologies such as X-Forwarded-For headers are required for decisions or logging based on the client's original IP address.

For maximum protection, network access control lists (ACLs) should be configured to limit requests to the application origin IP addresses to only those ranges originating from the F5 Silverline networks. In this configuration, DDoS and WAF configurations can ensure the trust in the X-Forwarded-For header information and reduce the likelihood of spoofed attacks.

2.2.2.1.1 Investigate necessity for SSL dual-key strategy

An increasing number of application layer attacks are hidden within SSL. The Silverline DDoS service can help defend against these attacks, but only if it can decrypt the necessary SSL traffic, and this means having a key and certificate in the name of the subscriber company.

Subscribers are (rightly) hesitant to share their SSL keys with any third party, and, for many, such sharing would violate several compliance directives. But there are ways around this problem. The dual-key strategy is one of them.

F5 Silverline can generate a new certificate request (CSR) that chains up to the subscriber's certificate management chain. This allows the F5 Silverline service to decrypt and scrub the subscriber's SSL traffic, and also allows the subscriber to revoke the F5 certificate at its discretion. And at no time is the subscriber's encryption key outside of their control or infrastructure.

2.2.2.2 Routed Mode: Network Protection

Silverline's "Routed mode" is designed for those organizations looking to mitigate attacks against their entire network infrastructure. Routed mode uses Border Gateway Protocol (BGP) for routing changes, and client traffic is directed through the nearest Silverline scrubbing center for attack mitigation. Sanitized traffic is then sent to the customer premises (typically) over Generic Router Encapsulation (GRE) tunnels. The return traffic exits the customer's infrastructure and goes directly to the client. This methodology is called n-path or direct server return (DSR).

2.2.2.2.1 Set MTU with GRE Tunnels

When traversing network connections over encapsulated tunnels, it is important to understand that the header reduces the size of the data allowed to traverse the link. In the case of the GRE tunnel, the IP maximum transmission unit (MTU) is 24 bytes less than the IP MTU of the real outgoing interface. For a typical network interface that means the IP MTU on the tunnel interface would be 1500 minus 24, or 1476 bytes.

Since most clients and servers are typically connected at the 1500 byte MTU, when the packets traverse the GRE tunnel they must be "broken up" into smaller fragments to be transmitted across the link. This fragmentation can add additional overhead latency to the connection. Additionally, some packets specifically request not to be fragmented (DF-bit is set). To reduce the overhead of fragmentation, enterprises can modify the MTU within their infrastructure or modify the maximum segment size (MSS) negotiated during the TCP handshake.

This is especially important when subscribers are applying Silverline DDoS Protection for existing VPN services such as SSL VPN or IPSec where the compounding reduction in MTU due to the various tunneling mechanisms can cause adverse effects if not properly tuned.

2.2.2.2.2 IP Reflection: Network Protection

IP Reflection is an alternative asymmetric technique to provide network infrastructure protection without the need for GRE tunnels. Enterprises with F5 BIG-IP Local Traffic Manager (LTM), can use IP Reflection. With IP Reflection there is no need to change any IP address, and the traffic is not impacted as it is with GRE. Subscribers who are interested in IP Reflection must have at least one spare/available publicly advertiseable network prefix equal to or greater than the network to be protected.

2.2.2.2.3 L2VPN

F5 Silverline DDoS Protection supports providing clean traffic back to subscribers via a partnership with an Ethernet-handoff / MPLS provider. This integration avoids the MTU and fragmentation issues as well, but requires the customer have provisioned circuits with the L2VPN provider, which results in increased costs. While a more expensive solution, the L2VPN option may be advantageous in some environments where the subscriber may not be able to control or influence the MTU settings for a GRE-based solution or have spare routable networks for IP Reflection.

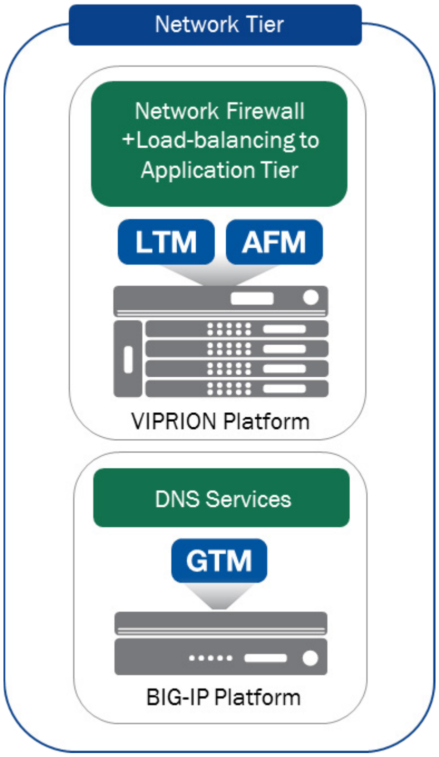
2.2.2.2.4 Equinix Cloud Exchange

F5 Silverline DDoS Protection has partnered with Equinix and is a published provider on Equnix's Cloud Exchange marketplace. Equinix Cloud Exchange is a service offering that allows subscribers who may already be hosting their infrastructure within Equinix metro locations to connect directly to F5 Silverline without traffic returning to the Internet as with GRE or IP Reflection. Additionally, service costs with Equinix Cloud Exchange are typically much lower than an L2VPN service offering.

2.3 On-Premises Network Defense Recommended Practices

At the organization's datacenter, the first defense is built around the network firewall. Most likely you already have a network firewall (it may or may not be F5) and a network firewall team (or at least an administrator). At this network firewall tier you will prepare defenses around layers 3 and 4 (IP and TCP). This is where you will mitigate smaller SYN and TCP floods, and block source addresses during a DDoS attack.

The following sections apply to the equipment at the network tier, whether that is the F5 AFM firewall module or an F5 LTM load balancer in front of another vendor's network firewall.



2.3.1 Choosing Virtual Server Types

Organizations using either the F5 firewall (AFM) or the F5 load balancer (LTM) at the network tier have a choice about how to structure their configuration. There are four options for defining a "listening" object. While all of these are valid ways to arrange the configuration, some have different strengths when dealing with DDoS.

- Full-Proxy Virtual Servers are the standard virtual servers in an F5 configuration. These listeners establish a real connection with each incoming client before they initiate a secondary connection to the server. The very act of terminating and validating the client connection provides a broad range of protection before the second tier is even invoked.
- Forwarding Virtual Servers perform faster and still protect against SYN floods, but do not provide the broader-level protection that full proxy virtual servers do.
- Wildcard Virtual Servers allow the decoupling of firewall rules from the application virtual server. This enables the creation of a rule that says "for any address supplying FTP services, apply this rule set, this mirroring policy, and this source NAT policy."
- Route Domains, which isolate duplicate IP subnets into logical, separate routing tables, are common in service provider environments. While route domains provide little or no benefit regarding DDoS in and of themselves, they can be used as pegs on which to hang layer 4 security policies.

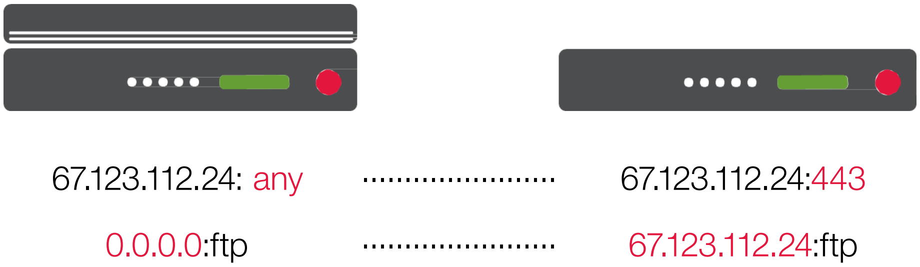


Figure 3: Wildcard servers are a common option at the network tier

An example configuration for the wildcard virtual server depicted in Figure 3 would look like the output below.

```
ltm virtual ws_ftp {
      destination 0.0.0.0:ftp
      ip-protocol tcp
      profiles {  ftp {  }  tcp {  }  }
      translate-address disabled
      }
```

In general, **F5 recommends the use of Full Proxy or Forwarding Virtual Servers** at the network tier when DDoS is a concern.

2.3.2 Mitigate SYN floods at the Network Tier

TCP SYN floods are always mitigated by F5, even SYN floods against Direct Server Return (DSR) virtual servers. To verify that your BIG-IP is managing SYN flood protection, you can view SYN flood statistics for each individual virtual server with the simple **show** command.

```
% tmsh show ltm virtual vip1

SYN Cookies
  Status                         full-hardware
  Hardware SYN Cookie Instances              1
  Software SYN Cookie Instances              0
  Current SYN Cache                          0
  SYN Cache Overflow                         5
  Total Software                             6
  Total Software Accepted                    0
  Total Software Rejected                   16
  Total Hardware                         19.1K
  Total Hardware Accepted                    1
```

Many F5 platforms can mitigate SYN floods in hardware. This allows the main traffic-steering CPUs to perform other tasks. The SYN Check™ Activation Threshold is the configurable number of connections at which point SYN Cookies will be activated.

The BIG-IP will send SYN cookie challenges on configured virtual servers. The FastL4 and TCP profiles have hardware SYN cookies enabled by default. To implement SYN cookies in a DSR deployment, create an additional profile and enable software SYN cookies similar to the following command example:

```
% tmsh create ltm profile fastl4 dsr_fastl4 loose-close enabled software-syn-cookie enabled
```

vCMP guests are able to leverage the hardware SYN cookie feature of the host. This allows for lightweight virtual BIG-IP guests to take advantage of the hardware processing while resources such as CPU and memory are isolated from other guests.

F5's vCMP virtualization technology is an ideal candidate for network tier services where resource reservation and resource isolation benefit your perimeter design. Virtual vCMP guests can leverage hardware resources within the host BIG-IP appliance including DDoS Hardware Signatures, SSL Offloading, and SYN cookie responses.

Table 1: SYN flood hardware support platform list

Platform	Hardware SYNs per second
B4300/B4340 Blade	80M
B2250 Blade	60M
B2100/B2150 Blade	40M
10200V	80M
10000S	40M
7200V	40M
7000S	20M
5200V	40M
5000S	20M

Older platforms may support SYN Cookies but may not be hardware-accelerated.

The following example creates a tcp profile with a tighter security posture by setting two DDoS-related variables. It enables hardware **SYN cookies**. It also sets the **deferred-accept** variable that reduces the impact that “zero-window” TCP attacks can have on the virtual server.

```
% create ltm profile tcp tcp_ddos { hardware-syn-cookie enabled deferred-accept enabled zero-window-timeout
10000 }
```

Then associate the new tcp profile with the virtual server by replacing the existing “tcp” profile:

```
% tmsh list ltm virtual vip1 profiles
% tmsh modify ltm virtual vip1 profiles replace-all-with { tcp_ddos my_ddos1 http }
```

2.3.3 Mitigate Stateless Protocol Processing

There are several stateless protocols that are valid for inbound requests to your datacenter. DNS is one in particular that uses the UDP protocol. Within the BIG-IP platform, the hardware-based disaggregator (DAG) is responsible for directing traffic to different CPU cores. For stateful traffic it is important that the response gets returned to the originating CPU. However, stateless protocols are forgiving in this regard. For optimal processing of stateless traffic, you should enable DAG round-robin load-balancing to ensure a reasonable distribution of traffic across the cores. This is enabled on the VLAN configuration. It should be set on your ingress and egress VLANs to ensure optimal distribution of traffic.

```
% modify net vlan VLAN120 dag-round-robin enabled
```

To validate which VLANs DAG round-robin is applied to you must add ‘all-properties’ to your list command.

```
% list net vlan VLAN120 all-properties

net vlan VLAN120_PUBLIC_SUBNET {
    dag-round-robin enabled
    if-index 96
    tag 120
}
```

If you have a virtual server protecting your DNS servers, you can direct the DAG to process UDP traffic on port 53:

```
% tmsh modify sys db dag.roundrobin.udp.portlist value “53”
```

2.3.4 Deny UDP and UDP Floods

UDP floods are a common DDoS vector because they are easy to generate and can be hard to defend. In general, do not allow UDP traffic to a virtual server unless the application behind it is actively accepting it.

Even for applications that accept UDP, a UDP flood can overwhelm the system, and you may find it necessary to temporarily deny UDP traffic to the application's virtual server.

```
% tmsh create security firewall rule-list drop_udp { rules add { drop_udp_rule
{  action drop ip-protocol udp   place-after first }  }  }
% tmsh modify ltm virtual vip1 fw-rules { drop_udp_vip1 { rule-list drop_udp }
}  }
```

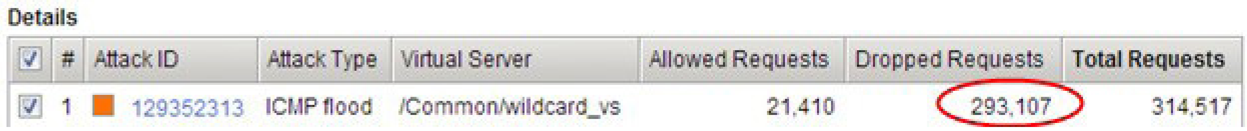
When the attack has ceased, you can remove the rule from the virtual server.

BIG-IP can monitor and mitigate UDP floods with granular exceptions. This enables a baseline of UDP traffic to pass through a virtual server at network tier. If the UDP traffic exceeds the thresholds, it is dropped unless it matches one of eight user-defined port exceptions (for example, RTSP or DNS).

2.3.5 Deny ICMP Floods

ICMP is another common DDoS vector. ICMP fragments are easy to generate and easy to spoof, and can tie up resources on many different types of networking devices.

AFM can differentiate between a normal amount of ICMP and an ICMP flood based on traffic pattern analysis. When AFM's network firewall is enabled on a virtual server, it will monitor for an increase in several types of traffic. A normal amount will be allowed, with the rest of the flood prohibited.



Details

☑	#	Attack ID	Attack Type	Virtual Server	Allowed Requests	Dropped Requests	Total Requests
☑	1	🟧 129352313	ICMP flood	/Common/wildcard_vs	21,410	293,107	314,517

Figure 4: Attack reporting

2.3.5.1 ICMP Checksum Hardware Offloading

BIG-IP devices take advantage of ICMP checksum offloading in hardware. This feature can save up to 7% of the CPU resources when handling large amounts of ICMP traffic.

2.3.6 Use the DDoS Device Profile of AFM

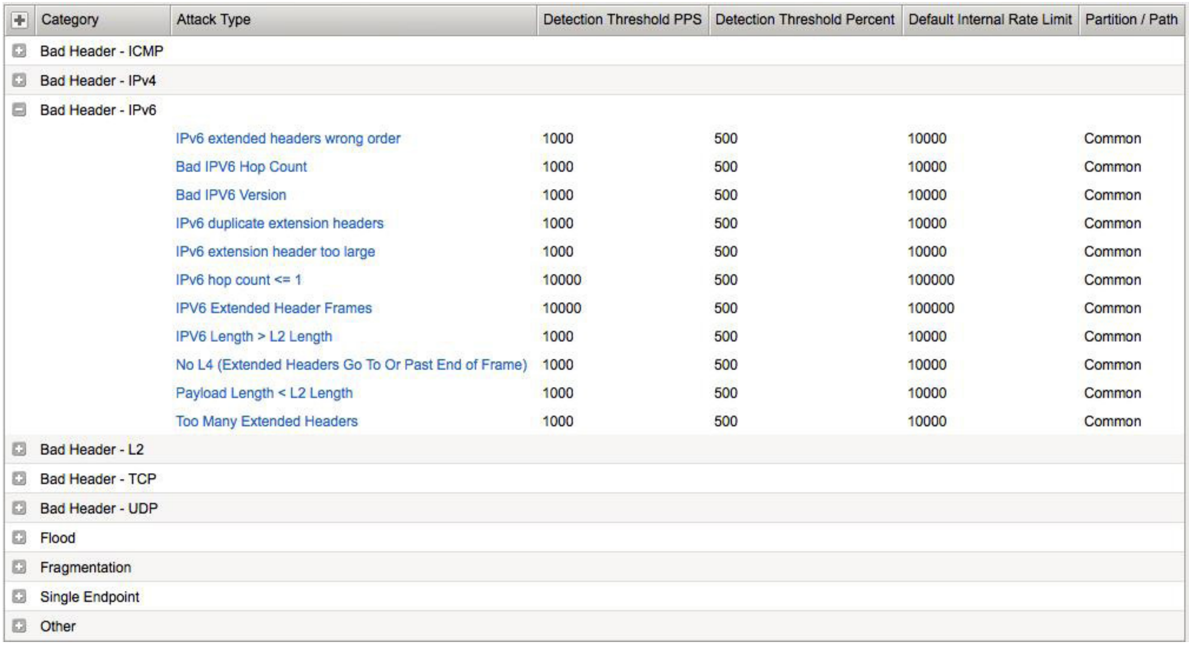
One way that attackers can consume firewall resources is by throwing floods of specially crafted invalid packets. The firewall will need to look at (and log) each packet. F5 has found that suspect combinations of flags (such as PSH+ACK with empty payloads) can be popular one month and then be abandoned in favor of a different combination later.

This shifting landscape makes it difficult to be predictive about what layers 3 and 4 attacks are likely to happen. The security administrator (for other vendors' firewalls) should be aware of these attacks and be ready to insert rules to block them, taking care to avoid using more CPU than necessary.

F5's approach to this problem has been to move much of the L3/L4 protocol validation into the custom hardware logic on the TMOS platforms that support it. By default, the AFM module is monitoring for dozens of layer 3 and layer 4 DDoS attack vectors such as floods of Christmas tree packets or LAND attack packets. Nearly all of these packets are discarded regardless of any BIG-IP setting. AFM can send a special log message when a flood of these packets is detected.

Table 1 (in section 2.2.2) shows which TMOS platforms have support for hardware-assisted L3/L4 protocol validation. These are the same platforms that have SYN flood hardware support.

All platforms (including the virtual edition) allow management of the parameters that track these L3/L4 suspect packet floods. The management screen is available from the Security tab of the user interface. Then select **DoS Protection** and **Device Configuration**.



Category	Attack Type	Detection Threshold PPS	Detection Threshold Percent	Default Internal Rate Limit	Partition / Path
Bad Header - ICMP					
Bad Header - IPv4					
Bad Header - IPv6					
	IPv6 extended headers wrong order	1000	500	10000	Common
	Bad IPV6 Hop Count	1000	500	10000	Common
	Bad IPV6 Version	1000	500	10000	Common
	IPv6 duplicate extension headers	1000	500	10000	Common
	IPv6 extension header too large	1000	500	10000	Common
	IPv6 hop count <= 1	10000	500	100000	Common
	IPV6 Extended Header Frames	10000	500	100000	Common
	IPV6 Length > L2 Length	1000	500	10000	Common
	No L4 (Extended Headers Go To Or Past End of Frame)	1000	500	10000	Common
	Payload Length < L2 Length	1000	500	10000	Common
	Too Many Extended Headers	1000	500	10000	Common
Bad Header - L2					
Bad Header - TCP					
Bad Header - UDP					
Flood					
Fragmentation					
Single Endpoint					
Other					

Figure 5: Network DDoS configuration

These settings are also available via the command line with the security dos device-config command. Also note that these settings are per traffic management microkernel (tmm), not per platform. In the table, the columns map to these values:

- Detection Threshold PPS. This is the number of packets per second (of this attack type) that the BIG-IP system uses to determine if an attack is occurring. When the number of packets per second surpasses the threshold amount, the BIG-IP system logs and reports the attack, and then continues to check every second and marks the threshold as an attack as long as the threshold is exceeded.
- Detection Threshold Percent. This is the percentage increase value that specifies an attack is occurring. The BIG-IP system compares the current rate to an average rate from the last hour. For example, if the average rate for the last hour is 1000 packets per second, and you set the percentage increase threshold to 100, an attack is detected at 100 percent above the average, or 2000 packets per second. When the threshold is passed, an attack is logged and reported. The BIG-IP system then automatically institutes a rate limit equal to the average for the last hour, and all packets above that limit are dropped. The BIG-IP system continues to check every second until the incoming packet rate drops below the percentage increase threshold. Rate limiting continues until the rate drops below the specified limit again.
- Default Internal Rate Limit. This is the value, in packets per second, that cannot be exceeded by packets of this type. All packets of this type beyond the threshold are dropped; rate limiting continues until the rate drops below the specified limit again.

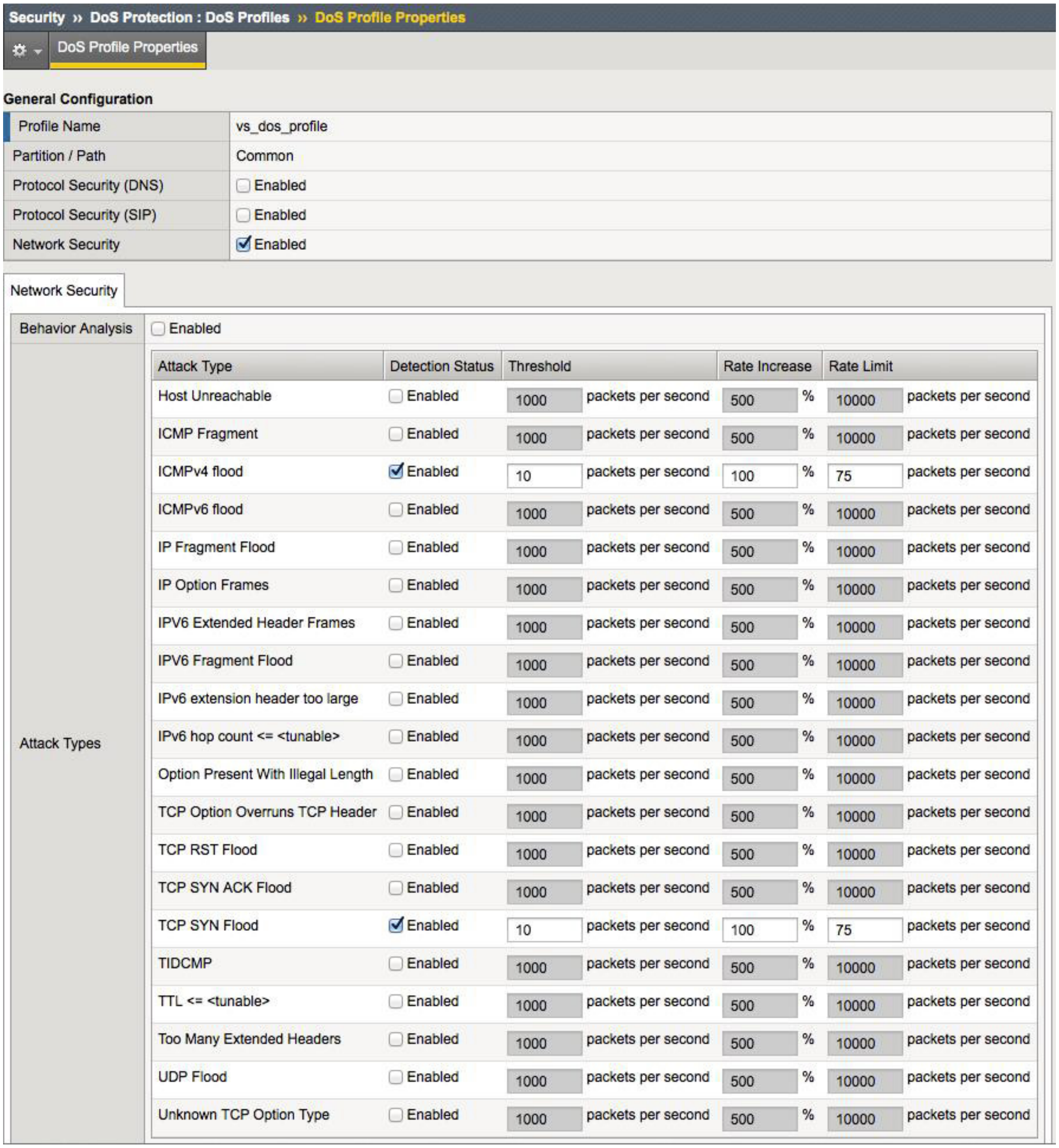
2.3.6.1 Single Endpoint Sweep and Flood Protection

BIG-IP can monitor and mitigate IP/TCP/UDP/ICMP floods and sweeps from a single endpoint. This can be used to identify and mitigate an attacker while allowing legitimate traffic to pass through a virtual server at the network tier. If a single IP address is responsible for traffic that exceeds the thresholds, it can be rate-limited or dropped. The following command configures the sweep-and-flood reaper:

```
% tmsh modify security dos device-config flood dos-device-vector { flood { detection-threshold-percent 100
detection-threshold-pps 1000 default-internal-rate-limit 20000 packet-types add|delete|none|replace-all-with {
tcp-syn-only }
```

2.3.7 Virtual Server DoS Profiles

Virtual-server processing takes place in software prior to the hardware processing at the global level. This is designed to prevent a single virtual server that is under attack from having the rate limit configuration applied to other virtual servers not under attack.



Security » DoS Protection : DoS Profiles » DoS Profile Properties

DoS Profile Properties

General Configuration

Profile Name	vs_dos_profile
Partition / Path	Common
Protocol Security (DNS)	□ Enabled
Protocol Security (SIP)	□ Enabled
Network Security	☑ Enabled

Network Security

Behavior Analysis □ Enabled

Attack Types

Attack Type	Detection Status	Threshold	Rate Increase	Rate Limit
Host Unreachable	□ Enabled	1000 packets per second	500 %	10000 packets per second
ICMP Fragment	□ Enabled	1000 packets per second	500 %	10000 packets per second
ICMPv4 flood	☑ Enabled	10 packets per second	100 %	75 packets per second
ICMPv6 flood	□ Enabled	1000 packets per second	500 %	10000 packets per second
IP Fragment Flood	□ Enabled	1000 packets per second	500 %	10000 packets per second
IP Option Frames	□ Enabled	1000 packets per second	500 %	10000 packets per second
IPV6 Extended Header Frames	□ Enabled	1000 packets per second	500 %	10000 packets per second
IPV6 Fragment Flood	□ Enabled	1000 packets per second	500 %	10000 packets per second
IPv6 extension header too large	□ Enabled	1000 packets per second	500 %	10000 packets per second
IPv6 hop count <= <tunable>	□ Enabled	1000 packets per second	500 %	10000 packets per second
Option Present With Illegal Length	□ Enabled	1000 packets per second	500 %	10000 packets per second
TCP Option Overruns TCP Header	□ Enabled	1000 packets per second	500 %	10000 packets per second
TCP RST Flood	□ Enabled	1000 packets per second	500 %	10000 packets per second
TCP SYN ACK Flood	□ Enabled	1000 packets per second	500 %	10000 packets per second
TCP SYN Flood	☑ Enabled	10 packets per second	100 %	75 packets per second
TIDCMP	□ Enabled	1000 packets per second	500 %	10000 packets per second
TTL <= <tunable>	□ Enabled	1000 packets per second	500 %	10000 packets per second
Too Many Extended Headers	□ Enabled	1000 packets per second	500 %	10000 packets per second
UDP Flood	□ Enabled	1000 packets per second	500 %	10000 packets per second
Unknown TCP Option Type	□ Enabled	1000 packets per second	500 %	10000 packets per second

Figure 5: Per Virtual Server DoS Profile Configuration

2.3.8 Mitigate TCP Connection Floods

TCP Connection floods are a layer 4 anomaly and can affect any stateful device on the network, especially firewalls. Often these floods are empty of actual content. LTM or AFM at the network tier can mitigate these by absorbing the connections into high-capacity connection tables.

Table 2: Connection Table Sizes

Platform	TCP-Connection Table Size	SSL-Connection Table Size
VIPRION 4800 (8 X B4340)	576 million	124 million
VIPRION 4800 (1 X B4340)	72 million	15.5 million
VIPRION 4480 (4 X B4340)	288 million	62 million
VIPRION 4480 (1 X B4340)	72 million	15.5 million
VIPRION 4480 (4 X B4300)	144 million	32 million
VIPRION 4480 (1 X B4300)	36 million	8 million
VIPRION 4400 (4 X B4200)	48 million	5 million
VIPRION 4400 (1 x B4200)	12 million	1 million
VIPRION 4400 (4 X B4200)	48 million	5 million
VIPRION 4400 (1 x B4200)	12 million	1 million
VIPRION 2400 (4 x B2250)	192 million	40 million
VIPRION 2400 (1 x B2250)	48 million	10 million
VIPRION 2400 (4 x B2150)	96 million	20 million
VIPRION 2400 (1 x B2150)	24 million	5 million
VIPRION 2400 (4 x B2100)	48 million	10 million
VIPRION 2400 (1 x B2100)	12 million	2.5 million
11000 series	24–30 million	2.64–3.9 million
10200 series	36 million	7 million
8900 series	12 million	2.64 million
7000 series	24 million	4 million
6900 series	6 million	660 thousand
5000 series	24 million	4 million
4200V series	10 million	850 thousand
3900 series	6 million	660 thousand
Virtual Edition	3 million	660 thousand

2.3.9 Configure Adaptive Reaping

Even with high-capacity connection tables, there are still settings that can be adjusted to deepen the protection profile against flood attacks.

If the BIG-IP connection table does begin to fill, connections will be “reaped” according to the adaptive reaping low-water and high-water settings. These can be adjusted downward from the default values of 85 and 95 in order to begin mitigating a “spiky” DDoS faster, and thus reducing the window during which the initial attack will load the servers.

```
%   tmsh modify ltm global-settings connection adaptive-reaper-lowater  75
```

The reaping function is configured through an eviction policy. An eviction policy is much more flexible in two respects: the variety of methods of reaping connections and the ability to be applied in separate contexts. F5 allows you to now specify a connection threshold under route domains and virtual servers and apply an eviction policy to those objects. The eviction policy can contain one or more eviction strategies to dictate how connections are removed from the connection table. A global eviction policy can be applied similarly to previous versions with the additional option of configuring a more granular eviction strategy.

2.3.9.1 Connection Reaper Eviction Strategies

There are four eviction strategies that can influence the reaper’s selection of connections to remove from the connection table:

- **Bias Idle.** Specifies that the system biases flow removal toward the existing flows that have been idle, with no payload bytes, for the longest period.
- **Bias Oldest.** Specifies that the system biases flow removal toward the oldest existing flows.
- **Bias Bytes.** Will evict traffic flows that show less data being transferred. Configures the grace period to allow new connections to be spun up.
- **Low Priority.** Specifies objects that are identified as lower priority by the flow removal strategy. Those object are as follows:
    - Route Domains
    - Virtual Servers
    - Specified Ports and Protocols
    - Countries

The default eviction policy mimics the behavior of previous versions with bias oldest and bias idle as the selected algorithms. Below is an eviction policy created at the global level:

```
% tmsh create ltm eviction-policy my-dos-eviction-policy slow-flow { enabled true } strategies { bias-bytes
{ enabled true delay 10 } low-priority-geographies {  countries add { US } enabled true } }
```

To complete the configuration apply the eviction policy at any of the three available contexts:

- **Virtual Server** - based on Connection Counts

  ```
  modify ltm virtual vs_web flow-eviction-policy my-evict-policy
  ```

  ```
  modify net route-domain 0 connection-limit 100000
  ```

- **Route Domain** - based on Connection Counts

  ```
  modify net route-domain 1 flow-eviction-policy  my-evict-policy
  ```

  ```
  modify ltm virtual vs_web connection-limit 100000
  ```

- **Global** - based on memory consumption

  ```
  modify ltm global-settings connection global-flow-eviction-policy  my-evict-policy
  ```

Below is an example of the eviction strategy's performance. The Linux watch command updates the table every second.

```
% watch tmctl flow_eviction_policy_stat

policy_name                           swept_context          context_name           evicted
----------------------------------    ---------------------  ---------------------  -------
/Common/default-eviction-policy       route domain           /Common/0              0
/Common/default-eviction-policy       virtual server         /Common/vs_web         0
/Common/my-dos-eviction-policy        route domain           /Common/0              701
/Common/my-dos-eviction-policy        virtual server         /Common/vs_web         501
/Common/my-dos-eviction-policy        virtual server         /Common/vs_web2        200
/Common/sweeper                       route domain           /Common/0              5460
/Common/sweeper                       virtual server         /Common/vs_web         5460
/Common/sweeper                       virtual server         /Common/vs_web2        0
```

2.3.9.2 Slow-Flow Monitoring

Slow flows are determined by the number of bytes per second that are being sent on a given connection. This is a configurable sub-component of an eviction policy. To mitigate slow flows, modify the eviction policy to throttle slow connections. Also, configure a grace period to allow for new TCP connections to be completed as setting this too low can prevent new connections from being established.

- **Absolute.** All slow flows above this number of connections are dropped.
- **Percentage.** This percentage of detected slow flows are dropped. By default with this setting, all slow flows (100%) are dropped.

The last step in slow-flow mitigation is to define how slow flows should be evicted according to the bullets and example below.

```
% tmsh modify ltm eviction-policy slow_conn_only slow-flow { enabled true throttling enabled maximum 10 }
```

2.3.10 Modify Idle Timeouts to Combat Empty Connection Floods

While layer 4 connection floods do not typically pose a high risk to F5 devices, they definitely impact other stateful devices such as other firewalls. Those devices will nearly always collapse long before the F5 state tables fill up (see Table 2 in section 2.3.8). If the connection flood consists primarily of empty connections, you can instruct BIG-IP to be more aggressive about closing these empty connections.

There are three primary profiles associated with layer 4 on BIG-IP:

- **fastL4**—the hardware-assisted, high-performance TCP profile
- **tcp**—the standard TCP profile used by the majority of virtual servers
- **udp**—the standard UDP profile

Note: You may see other profiles, such as those associated with WAN optimization, which are based on the tcp or udp profiles.

Use the following attributes of these profiles to control how long a connection is idle before it is closed by BIG-IP. During a heavy attack, use smaller and smaller values.

For the fastL4 profile, override the **reset-on-timeout** and **idle-timeout** values. The default timeout is 300 seconds, which should be trimmed significantly during an attack. During normal operations the idle-timeout should be tuned to the application timeout the BIG-IP is protecting to ensure consistency and application performance.

```
%   tmsh create ltm profile fastl4 fastl4_ddos { reset-on-timeout disabled idle timeout 15  }
```

For each fastL4 virtual server under attack, replace the fastL4 profile with your new one.

For the tcp profile, override the same two values for the same reasons. While you are there, you may also want to adjust the **hardware-syn-cookie** and **zero-window-timeout** values. See section 2.3.2.

For the udp profile, reduce only the **idle-timeout** value (the default is 60 seconds).

2.3.11 Control Rate-Shaping

Another defensive technique that can be deployed quickly is rate-shaping. Rate-shaping can limit the rate of ingress traffic at the BIG-IP and may be the easiest way to push back against a volumetric attack. Though powerful, rate-shaping is a less-than-ideal technique for defending against DDoS because it does not differentiate between good and bad requests: rate-shaping can discard your good traffic as well. Granular rate-shaping based on attack types within the DoS Profiles allows you to better restrict bad traffic while allowing good traffic to pass without restriction.

You configure rate-shaping profiles manually and then assign them to a virtual server.

In this example, the rate-shaping class named “protect_apache” guarantees that at least 1Mbs of traffic will reach the target, but that no more than 10Mbs will be allowed:

```
net rate-shaping class protect_apache  {
rate 1mbps
      ceiling 10m bps
}
```

Then apply this rate-shaping class to each of your targeted virtual servers.

2.3.12 Set the Max ICMP Reject Rate

The **TM.MaxRejectRate** system variable can reduce the effects of a denial-of-service attack by allowing you to limit the number of TCP RSTs or ICMP unreachable packets that the BIG-IP system sends in response to incoming connections that cannot be matched with virtual server connections. The default value for the **TM.MaxRejectRate** system variable is 250 TCP RSTs or 250 ICMP unreachable packets per second.

Dropping the value to 100 can contribute to a reduction in outbound congestion without otherwise affecting network performance:

```
%   tmsh modify sys d b  tm.maxrejectrate value 100
```

2.3.13 Configure DoS Whitelist

DoS whitelists allow you to configure trusted networks, protocols, and VLANs. There are eight entries available, which when configured are excluded from standard DoS checks. A whitelist can contain:

- IP addresses or networks
- VLANs
- Ports
- Protocols

In a clustered scenario it is recommended to, at a minimum, create a DoS whitelist for your HA VLAN where configuration sync and mirroring traffic are enabled.

```
% modify security dos network-whitelist dos-network-whitelist entries add { ha-whitelist { source { vlans
2000 } } }
```

2.3.14 IP Intelligence Blocking

IP Intelligence is a subscription service that allows F5 customers to configure blocking of known sources of bad traffic. Many of the categories such as “Botnets“ and “Denial of Service” are sources identified for previously having been participants in attacks or having the future potential to participate in a DDoS attack. Configuring the BIG-IP to drop this traffic if you can identify categories that should never communicate with your application can reduce the threat from these sources. These categories when configured can be used to supplement your security policy to drop traffic from known bad actors. F5 also includes the ability to create your own feed sources, which can be populated from other tools or intelligence systems.

IP Intelligence policies can be configured and applied to the Global context, to route-domains, or virtual servers. This selection of categories identified for mitigation is Botnets, Cloud Service Providers, Denial of Service, Illegal Websites, Infected Sources, Phishing, Anonymous Proxies, Scanners, Spam Sources, Web Attacks, and Windows Exploits.

To create an IP Intelligence policy, navigate to Security > Network Firewall : IP Intelligence : Policies and select the category.

```
% tmsh create security ip-intelligence policy tier1-blacklist blacklist-categories add { botnets }
```

Custom feeds for blacklists or whitelists can be created based on your organization's investigations or other sources and downloaded from a URL regularly. The below examples configure IP intelligence and associate it to the available contexts:

```
% tmsh create security ip-intelligence feed-list fraud_detected { feeds add { bad_pcs { default-blacklist-
category windows_exploits } http://10.20.0.80/feed_list.txt } }
```

```
% tmsh modify ltm virtual vs_web ip-intelligence-policy ip-intelligence
```

```
% tmsh modify net route-domain 0 ip-intelligence-policy ip-intelligence
```

```
% tmsh modify security ip-intelligence global-policy ip-intelligence-policy ip-intelligence
```

2.4 On-Premises Application Defense Recommended Practices

The application tier is where you deploy application-aware, CPU-intensive defense mechanisms like login-walls, web application firewall policy, and LTM iRules. This tier is also where SSL termination typically takes place. While some organizations terminate SSL at the network tier, it is less common there due to the sensitivity of SSL keys and policies against keeping them at the security perimeter.

2.4.1 Understand GET Floods

Recursive GETs and POSTs are among today's most pernicious attacks. They can be very hard to distinguish from legitimate traffic.

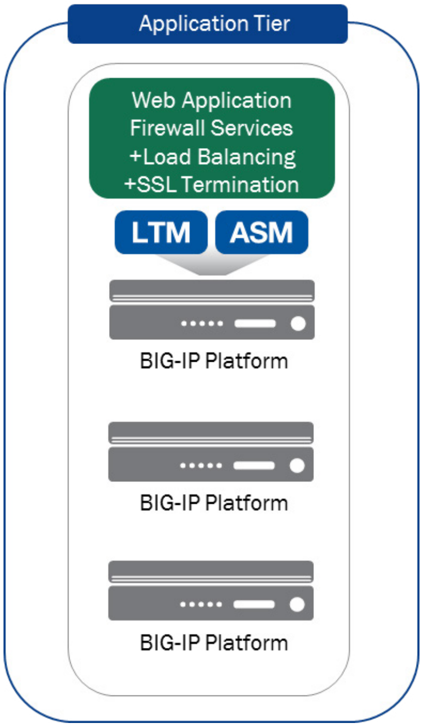


Figure 7: Application tier

GET floods can overwhelm databases and servers and can also cause a "reverse full pipe." F5 recorded one attacker that was sending 100Mbs of GET queries into a victim and bringing out 20Gbs of data.

If you have a signature-based anti-DDoS solution (from F5 or another vendor), leverage it to protect your application. With LTM and ASM, F5 provides many different ways to mitigate difficult application-layer attacks.

Mitigations strategies for GET floods include:

- The Login-Wall Defense
- DDoS Protection Profiles
- Real Browser Enforcement
- CAPTCHA
- Request-Throttling iRules
- Custom iRule

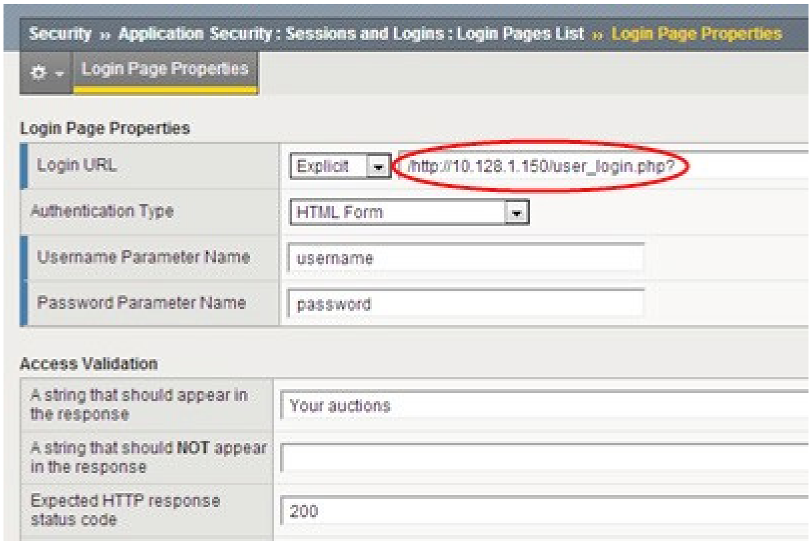
2.4.2 Reduce Threat Surface by Configuring a Login-Wall

The most powerful technique to foil application-level attacks is to allow only authenticated users to access the database portions of your application. Creating a login-wall can be delicate work that is much better done during peacetime and not during a hectic DDoS attack. Note that not all applications can rely on registered users and have to process anonymous traffic, but for those that can, login-walls are the defense.

2.4.2.1 Designate a Login-Wall with ASM

ASM offers facilities to do this within the ASM policy through the use of login pages and login enforcement. This feature will ensure that users cannot interact with a set of URLs until they have successfully authenticated themselves at one of the login pages.

First define the login pages from the Security -> Application Security -> Sessions and Logins screen.



Security » Application Security : Sessions and Logins : Login Pages List » Login Page Properties
Login Page Properties

Login Page Properties

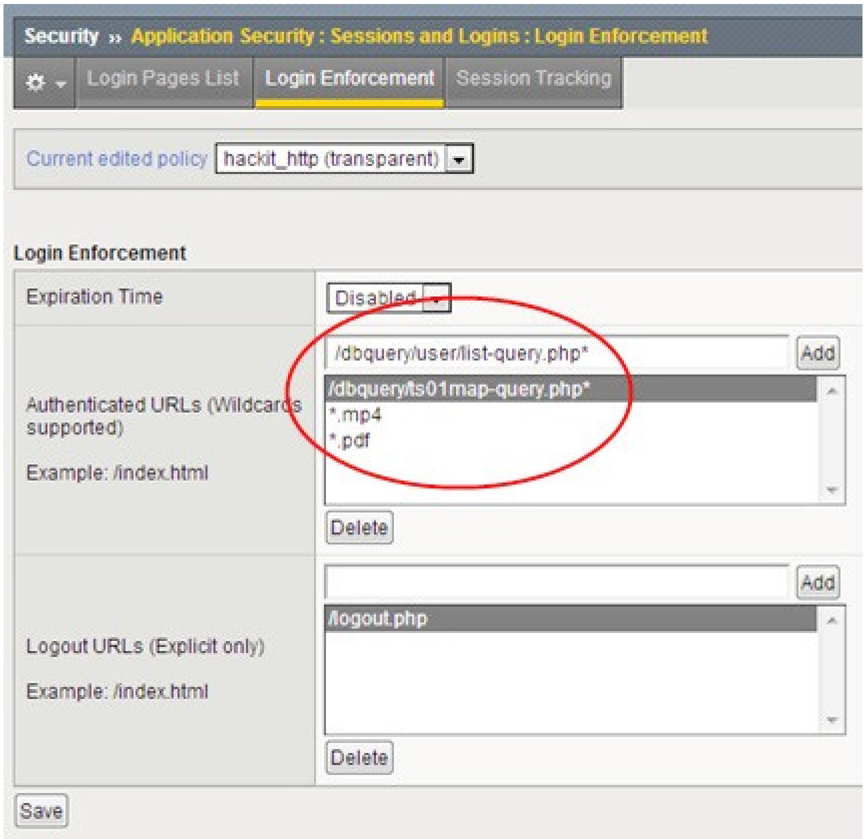
Login URL	Explicit /http://10.128.1.150/user_login.php?
Authentication Type	HTML Form
Username Parameter Name	username
Password Parameter Name	password

Access Validation

A string that should appear in the response	Your auctions
A string that should **NOT** appear in the response	
Expected HTTP response status code	200

Figure 8a: Defining a Login-Wall with ASM

Then use the Login Enforcement tab to specify which pages need to be protected. Ideally these will be large objects such as .MP4s and .PDFs, and any database queries that could be used against you in an asymmetric attack.


Security » Application Security : Sessions and Logins : Login Enforcement
Login Pages List
Login Enforcement
Session Tracking
Add
Delete
Add
Delete
Save

Current edited policy: hackit_http (transparent)

Login Enforcement

Expiration Time	Disabled
Authenticated URLs (Wildcards supported) Example: /index.html	/dbquery/user/list-query.php* /dbquery/ts01map-query.php* *.mp4 *.pdf
Logout URLs (Explicit only) Example: /index.html	/logout.php

Figure 8b: Defining a Login-Wall with ASM

For a full explanation of the Login Enforcement feature, see the section “Creating Login Pages” in the ASM configuration guide.

Note: If you are not sure what resources to protect, you can reconnoiter your own applications— see section 3.2.2.

2.4.2.2 Script a Login-Wall

You can make a login-wall with just an LTM iRule by setting a specific cookie at the login page, and then checking that cookie on every other page. Create this iRule, attach it, and test it. Then detach it and keep it in your library to be activated as necessary.

Here’s a link to a login-wall iRule:

https://devcentral.f5.com/wiki/iRules.Simple-Login-Wall-iRule-Redirect-unauthenticated-users-back-to-login-page.ashx

2.4.2.3 Protect Applications with DoS Protection Profiles

The F5 Web Application Firewall, ASM, includes application-specific DoS profiles. These powerful profiles detect DoS conditions by monitoring **server latency** or **http request rates**. ASM can then trigger an optional iRule event as the attack is mitigated.

The mitigation options are:

- Drop the suspicious connections
- Return a JavaScript redirect to the client to enforce that a browser is being used
- Rate-limit by client address or URI

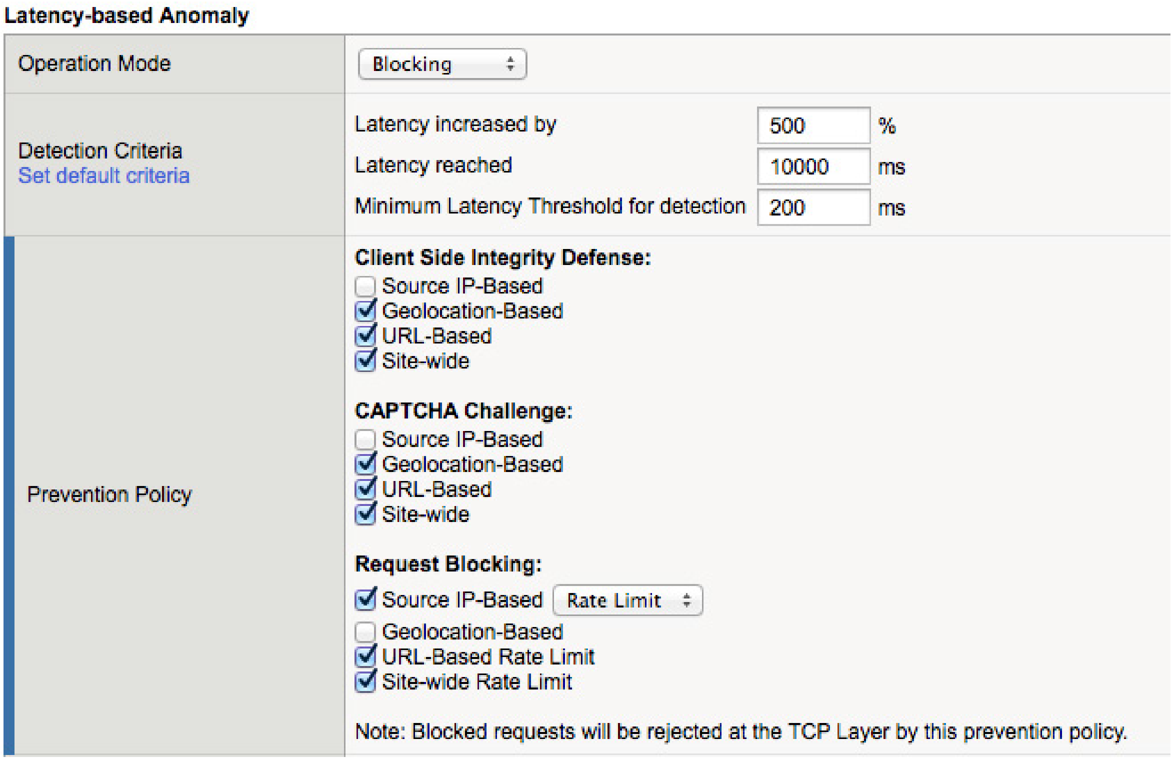
Use the following commands to create a DoS profile and attach it to the application:

```
% tmsh create security dos profile my_dos_prof { application add { Lrule1 { latency-based {  url-rate-
limiting enabled mode blocking }  }  }  }
```

```
%   tmsh modify ltm virtual my_vip1 profiles add {  my_dos_prof }
```

Access this DoS profile from the Security tab and then select DoS Protection. From that screen check Application Security and then configure the L7DOS protection parameters.

Another mitigation option is to add a CAPTCHA challenge when thresholds are reached. In addition, F5 provides a more granular Prevention Policy configuration to allow you to specify which prevention policy is applied based on Source IP, Geolocation, URL, or site-wide. CAPTCHA is the ultimate detection for human detection. You can also customize the CAPTCHA page style to be consistent with that of your application.



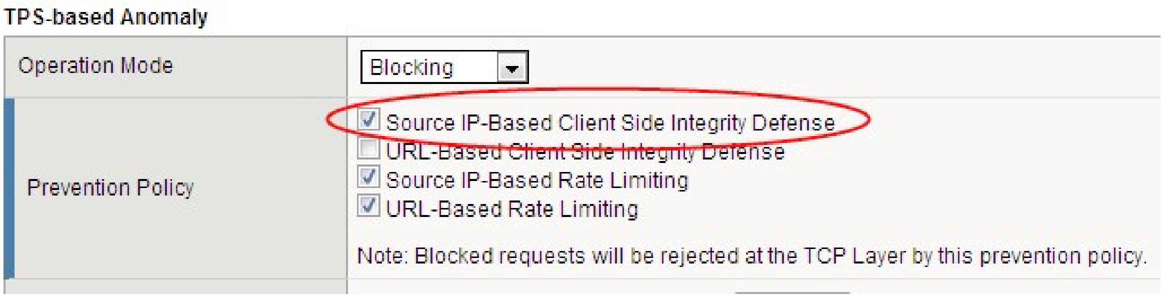
Latency-based Anomaly

Operation Mode	Blocking
Detection Criteria Set default criteria	Latency increased by 500 % Latency reached 10000 ms Minimum Latency Threshold for detection 200 ms
Prevention Policy	**Client Side Integrity Defense:** □ Source IP-Based ☑ Geolocation-Based ☑ URL-Based ☑ Site-wide **CAPTCHA Challenge:** □ Source IP-Based ☑ Geolocation-Based ☑ URL-Based ☑ Site-wide **Request Blocking:** ☑ Source IP-Based Rate Limit □ Geolocation-Based ☑ URL-Based Rate Limit ☑ Site-wide Rate Limit Note: Blocked requests will be rejected at the TCP Layer by this prevention policy.

Figure 9: Configuration for the ASM module's comprehensive L7DOS protection

2.4.2.4 Enforce Real Browsers

There are additional ways that F5 devices can separate real web browsers from probable bots. The easiest way, with ASM, is to create a DoS protection profile and turn on the "Source IP-Based Client Side Integrity Defense" option. This will inject a JavaScript redirect into the client stream and verify each connection the first time that source IP address is seen.



TPS-based Anomaly

Operation Mode	Blocking
Prevention Policy	☑ Source IP-Based Client Side Integrity Defense □ URL-Based Client Side Integrity Defense ☑ Source IP-Based Rate Limiting ☑ URL-Based Rate Limiting Note: Blocked requests will be rejected at the TCP Layer by this prevention policy.

Figure 10: Insert a Javascript redirect to verify a real browser

From the command line:

```
% modify security dos profile my_ddos1 application modify { Lrule1 { tps-based {
ip-client-side-defense enabled  }  }  }
```

2.4.2.5 Configure Heavy URLs to Mitigate Stress Attacks

Heavy URLs are defined as those that may consume considerable server resources per request. Attackers prefer these URLs because they provide the largest return for their efforts as a small number of requests can consume a significant amount of resources on a server as opposed to standard requests. An example of a request would be a site that typically returns small query values but has the potential to return larger queries such as a website search function or a stock quote page. Abuse of the URL can have a request that responds with a very loose search string or a stock quote for a long period in detail. The latter examples would require more time for the server to retrieve all the values and present the information.

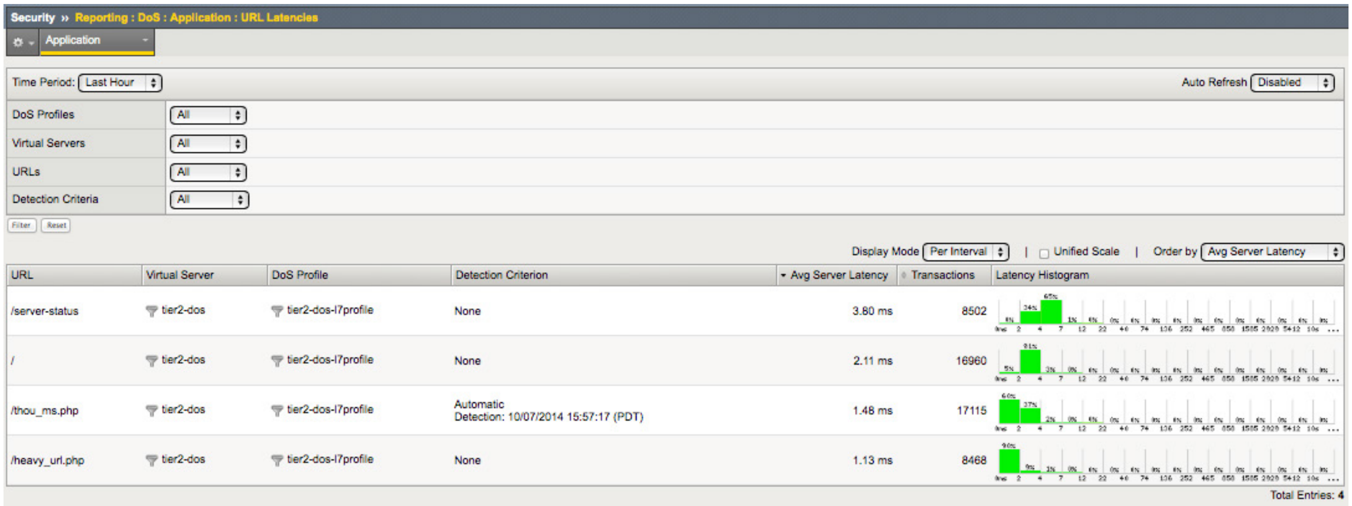
Latency is used as the measurement tool because it best represents resource consumption on the back-end servers. Those cases when the server's resources are stressed—as when, for example, there is a problem with the site or there is a DoS attack, or when the server is under load while processing requests. Typically, when a server becomes stressed and latency on the back end increases, it continues for some time until the stress condition situation resolves itself. This can be the conclusion of the intensive request or termination of the DoS attack.

The idea behind the heavy URL protection is to identify potentially heavy URLs not standard load and when an attack starts to mitigate the traffic destined for those URLs. F5 allows you to configure those URLs manually, or the BIG-IP can attempt to identify potentially heavy URLs automatically. F5 builds a histogram of the response latency for URLs, which you can use to determine the optimal setting for the latency threshold. You can use this histogram to statically configure the specific URLs that you believe may meet the heavy URL description.

```
% tmsh modify security dos profile tier2-dos-profile application modify { tier2-dos-profile { heavy-urls {
include add { /thous_ms.php }}}}
```

And you can allow for the BIG-IP to determine automatic detection of URLs.

```
% tmsh modify security dos profile tier2-dos-profile application modify { tier2-dos-profile { heavy-urls {
automatic-detection enabled }}}
```



Security » Reporting : DoS : Application : URL Latencies

Time Period: Last Hour

Auto Refresh: Disabled

DoS Profiles	All
Virtual Servers	All
URLs	All
Detection Criteria	All

Display Mode: Per Interval | □ Unified Scale | Order by: Avg Server Latency

URL	Virtual Server	DoS Profile	Detection Criterion	Avg Server Latency	Transactions	Latency Histogram
/server-status	tier2-dos	tier2-dos-l7profile	None	3.80 ms	8502	
/	tier2-dos	tier2-dos-l7profile	None	2.11 ms	16960	
/thou_ms.php	tier2-dos	tier2-dos-l7profile	Automatic Detection: 10/07/2014 15:57:17 (PDT)	1.48 ms	17115	
/heavy_url.php	tier2-dos	tier2-dos-l7profile	None	1.13 ms	8468	

Total Entries: 4

Figure 11: URL latencies histogram with automatic detected heavy URL

After a couple of days of standard traffic, you can evaluate the 95th percentile to determine the peak latency. Configure the latency threshold to reflect the highest average across the URLs.

```
% tmsh modify security dos profile tier2-dos-profile application modify { tier2-dos-profile { heavy-urls {
latency-threshold 1000 }}}
```

Lastly, for the heavy URL to be effective, it needs to have a mitigation method. The two methods are client-side integrity challenges via a javascript challenge and rate-limiting.

```
% tmsh modify security dos profile tier2-dos-profile application modify { tier2-dos-profile { latency-based {
ip-client-side-defense  enabled ip-rate-limiting enabled }}}
```

2.4.2.6 Site-wide Mitigation

Site-wide mitigation is designed to detect attacks from large-scale botnets or other attacks originating from multiple IP addresses attacking multiple URLs on the site. Configure the BIG-IP to observe that the entire site is under attack and mitigate against both TPS and latency-based attacks. It is also considered a “last resort” detection and mitigation, meaning the site is experiencing a DDoS attack, but it is not limited to a specific URL or did not arrive from a specific source IP.

```
% tmsh modify security dos profile tier2-dos-profile application modify { tier2-dos-profile { latency-based {
ip-client-side-defense  enabled ip-rate-limiting enabled }}}
```

2.4.2.7 Throttle GET Request Floods via Script

The F5 DevCentral community has developed several powerful iRules that automatically throttle GET requests. Customers are continually refining these to keep up with current attack techniques.

Here is one of the iRules that is simple enough to be represented in this document. (The live version can be found at this DevCentral page: HTTP-Request-Throttle.)

```
when RULE_INIT {

      # Life timer of the subtable object. Defines how long this object exist in the subtable

      set static::maxRate 10

      #  This defines how long is the sliding window to count the requests.
      #  This example allows 10 requests in 3 seconds

      set static::windowSecs 3
      set static::timeout 30
}

when HTTP_REQUEST  {
      if {  [HTTP::method] eq “GET” }  {
              set getCount [table key -count -subtable [IP::client_addr]]
              if {  $getCount <  $static::maxRate }  {
                      incr getCount 1
                      table set -subtable [IP::client_addr] $getCount “ignore”
                      $static::timeout $static::windowSecs
              }  else {
                      HTTP::respond 501 content “Request blockedExceeded requests/sec limit.”
                      return
              }
      }
}
```

Another iRule, which is in fact descended from the above, is an advanced version that also includes a way to manage the banned IP’s address from within the iRule itself:

- **URI-Request Limiter iRule**—Drops excessive HTTP requests to specific URIs or from an IP

https://devcentral.f5.com/wiki/iRules.HTTP-URI-Request-Limiter.ashx

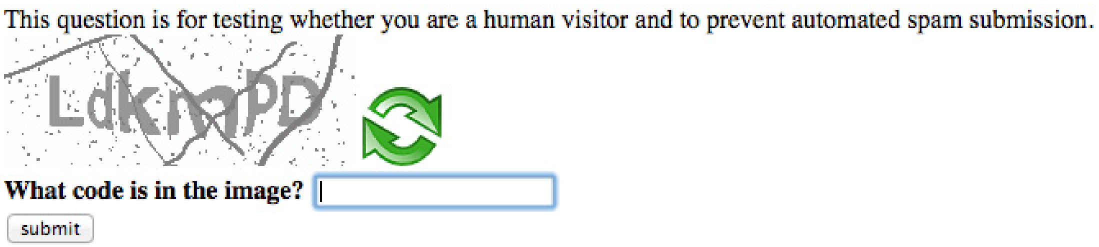
2.4.2.8 Use CAPTCHAs to Eliminate Bots

Another way to mitigate GET floods is by verifying "humanness" by using a CAPTCHA mechanism. The CAPTCHA mechanism shows pictures of scrambled words to the user, who proves his or her humanity by typing the words into a web-form. CAPTCHAs are still one of the best ways to distinguish humans from computers even though hackers and researchers have been trying to "break" them for more than 10 years. Advances in pattern-recognition algorithms seem to bring attackers close to automating the CAPTCHA system. It is F5's experience, though, that the computational work required to "break" a CAPTCHA vastly decreases the asymmetric advantage of a modern DDoS attacker, and this keeps these attacks theoretical for now. This means that CAPTCHAs are still an effective means of repelling botnets.

F5 allows you to configure CAPTCHA as a mitigation method for TPS and latency-based anomalies under the application DoS profiles. This can be added to the DoS profile to mitigate attacks and force clients to answer the challenge before proceeding to the protected content. F5 also allows you to customize the message for the website under the specific DoS Profile.

```
% tmsh modify security dos profile l7_dos application modify {l7_dos { latency-based { url-captcha-challenge
enabled } } }
```

Below is an example of the default CAPTCHA challenge sent to clients to verify their authenticity.



This question is for testing whether you are a human visitor and to prevent automated spam submission.

What code is in the image?

Figure 12: CAPTCHA default page

For those who are attempting to mitigate attacks without ASM, Google offers the reCAPTCHA service, which performs this function while also decoding ancient texts. There is a Google ReCAPTCHA iRule on DevCentral that can be used to provide verification that a human is at the other end of the connection. Download the iRule (approximately 150 lines) and edit it to provide some of the basic information (such as your Google reCAPTCHA key and your DNS server). Make it available on your BIG-IP. Attach it to a virtual server and test it, and then keep it ready for deployment.

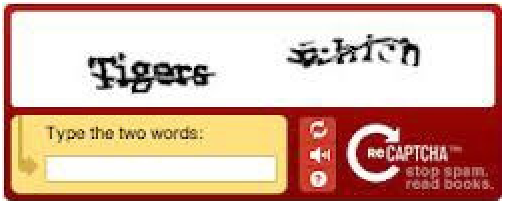


Figure 13: iRule-based CAPTCHA

2.4.2.9 Proactive BOT defense

The DoS profile can now be configured to attempt to detect automated attacks executed by bots before they happen by challenging the clients to prove that they are truly a browser before passing on the request. This configuration sends a javascript challenge that checks the browser's capabilities and adds an ASM cookie to the response. A browser is more than likely to support the embedded javascript and respond with the proper cookie. This mitigation technique can be configured to be always on or enabled only when an attack is detected to reduce the performance requirements.

```
% tmsh modify security dos profile l7_dos application modify { l7_dos { bot-defense { mode during-attacks
grace-period 300 }}}
```

2.4.3 Geolocation Whitelists and Blacklists

F5 gives you the ability to override the DoS profile's **Geolocation Detection Criteria** threshold settings by selecting countries from which to allow or block traffic during a DoS attack. If you know your organization doesn't normally do business with certain countries, or it is determined that the majority of the attacks appear to be coming from a particular country, you can enforce mitigation on those countries. Alternatively, whitelists are configured for countries that you want to allow anyway.

```
%tmsh modify security dos profile l7_dos application modify { l7_dos { geolocations replace-all-with {
Canada }}}
```

2.4.4 Script a Custom Mitigation

If all other techniques must be ruled out, you may find it necessary to write a custom iRule to defend your application from an application-layer attack. These custom iRules typically fall into one of two categories: filtering and indiscriminate blocking.

While this is perhaps the most "manual" of all the techniques in this document, it is also the most powerful and the most used by agile F5 customers. The extreme programmability of F5 iRules gives an administrator the ability to block nearly any kind of attack provided that he or she can script well enough. Security-related iRules protect many organizations today and are one of F5's real differentiators for application-layer DDoS attacks.

If the attack leaves you with some outbound Internet access, search devcentral.f5.com for some of the keywords that might match your attack. You may find that an iRule has already been written for you!

To write your own iRule, first dissect the attack traffic and find a feature about the incoming attack traffic that you can use to distinguish bad traffic from good. Then write an iRule that detects that traffic and drops it. If you are not an iRule author, there are iRules scattered throughout this document (and all over DevCentral) that you can use as examples. Attach your new iRule to the application’s virtual server.

```
 when HTTP_REQUEST  {
  if {  [HTTP::header exists “Referer”] }  {
    if {  not ([HTTP::header “Referer”] contains “\x2F\x2F”) }  {
      drop
    }
  }
 }
```

If you are not able to easily distinguish good traffic from bad, you can write an iRule that discards traffic based on the object being requested. For example, if the attackers are requesting a particular large PDF or MP4 file, you can use an iRule to drop all requests to that object.

```
ltm data-group internal block_uris {
records {
        /faqs/faq.mp4 {  }
        /locator/locations.pdf {  }
        /cgi-bin {  }
    }
    ty pe string
}
```

You can also use external data groups that are hosted outside the BIG-IP.

Then use a simple scrubber iRule to drop requests for URIs that match the data class.

```
when HTTP_REQUEST  {
  set origUri “”
  if {[HTTP::query] eq “”}{
    set origUri “[URI::path [HTTP::path]][URI::basename [HTTP::path]]”
  }  else {
    set origUri “[URI::path [HTTP::path]][URI::basename
[HTTP::path]]?[HTTP::query]”
  }
  if {  [class match -- [ string tolower $origUri ] contains block_uris] }
{
drop
  }
}
```

This is definitely not the best solution, because it will turn away good traffic as well as bad. It may keep your servers alive, but if you have the time and ability to write a rule like the one immediately above, you can usually find something to distinguish good traffic from bad.

2.4.5 Mitigate SSL DDoS at the Application Tier

While it is possible, and sometimes preferable, to terminate SSL at either tier, F5 recommends a physical (nonvirtual) appliance for terminating SSL at the application tier. Many SSL DDoS attacks will be mitigated by the very presence of the SSL acceleration hardware used in F5 physical devices. These include:

- SSL protocol attacks
- SSL replay attacks
- SSL connection floods

Whether or not hardware is used, F5 will also mitigate SSL connection floods with adaptive reaping (section 2.3.9.1) and a high-capacity connection table (section 2.3.8).

The SSL renegotiation attack can be mitigated in one of two ways. In most cases, you can simply temporarily disable the SSL renegotiation feature at the virtual server’s SSL clientssl profile. However, very long-lived connections (such as automatic teller machines or database connections) will still require the ability to renegotiate. For those cases, see the SSL Renegotiation DDoS iRule in the appendix of this document.

Mitigation for connection floods and slow-flow requests should be enabled at the application tier as well. Eviction policies should be configured as in section 2.3.9 Configure

Adaptive Reaping and Eviction Policies to provide additional mitigation and manage the connection table on the application tier devices.

2.4.6 Understand Connection Multiplexing and Port Exhaustion

In general, do not perform functions like connection multiplexing and SNAT at the network tier. These functions and associated extras like the insertion of the X-Forwarded-For header should be processed at the application tier.

2.4.6.1 Connection Multiplexing

A layer 7 DDoS can exhaust back-end resources such as connection tables. One way to combat this effect is by multiplexing the connections through the load balancer. On LTM this feature is called OneConnect and can decrease the number of TCP connections used by an order of magnitude while still maintaining (or even improving) overall requests-per second.

The OneConnect feature should be tested with each application prior to being used as DDoS defense. Some applications may rely on separate connections per user.

2.4.6.2 Port Exhaustion

A SNAT supports approximately 64,000 concurrent connections per destination IP. A high volume of requests can exceed the 64,000 connection limit and result in TCP port exhaustion. You can use a SNAT pool to overcome this limitation. Configure and set the appropriate IP address within the SNAT pool to mitigate the exhaustion.

For example, if your virtual server **vip1** is using simple automap source address translation, you can change it to use a pool of IP addresses with the immediately following commands. This example uses just three addresses to increase the available ports from 64,000 to 192,000.

```
% tmsh create ltm snatpool  ddos_snatpool members add { 10.1.20.161 10.1.20.162
10.1.20.163 }

% tmsh modify ltm virtual vip1 source-address-translation {  pool ddos_snatpool }
```

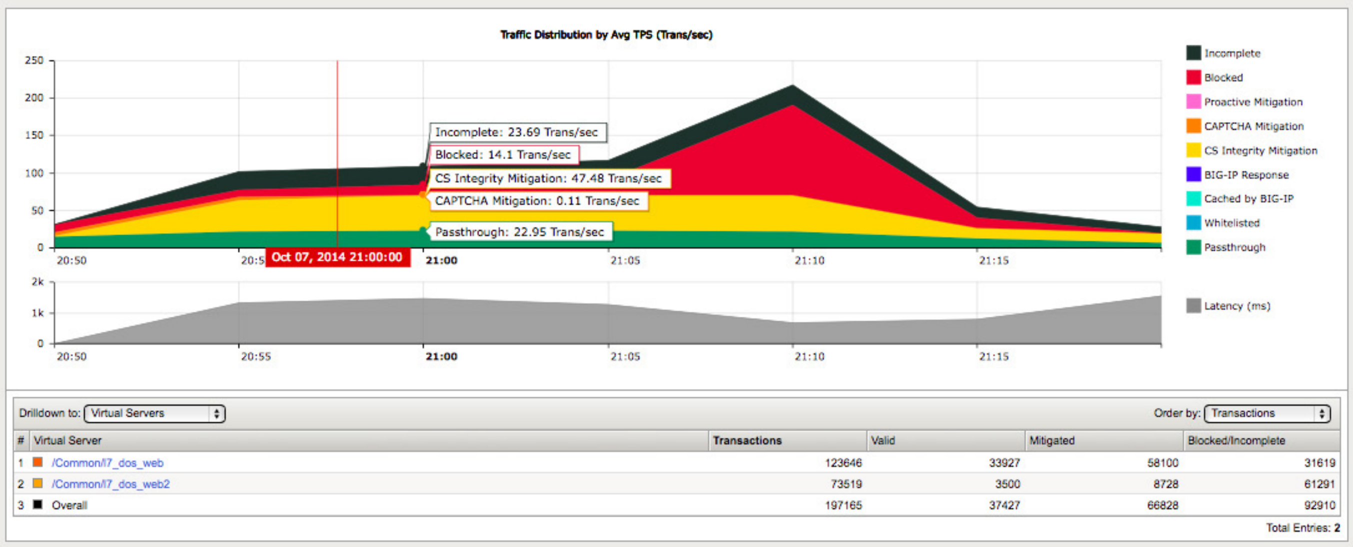
For each address added to the SNAT pool, you may want to assign a discrete timeout value (the default is indefinite). With an idle timeout, BIG-IP can close idle connections and help protect upstream stateful firewalls.

```
% tmsh modify ltm snat-translation 10.128.20.161 { ip-idle-timeout 60
```

Repeat this command for each of the addresses in your SNAT pool (10.1.20.162 and 10.1.20.163 in the example above).

2.3.6 Attack and Mitigation Visibility

F5 now provides many new enhanced DoS reports and the ability to generate your own detailed reports. This allows you to better visualize DoS attacks and mitigation of those attacks. Included are real-time stats that let you answer the age-old question, "Are we under attack?"



Drilldown to: Virtual Servers
Order by: Transactions

#	Virtual Server	Transactions	Valid	Mitigated	Blocked/Incomplete
1	🟧 /Common/l7_dos_web	123646	33927	58100	31619
2	🟧 /Common/l7_dos_web2	73519	3500	8728	61291
3	■ Overall	197165	37427	66828	92910

Total Entries: **2**

Figure 14: Traffic distribution by Avg TPS and mitigations applied

Real-time attack monitoring tracks different attacks and their severity over time, allowing you to determine which attacks may currently be affecting your site. The severity is based on the impact on your application's performance during the duration of the attack.

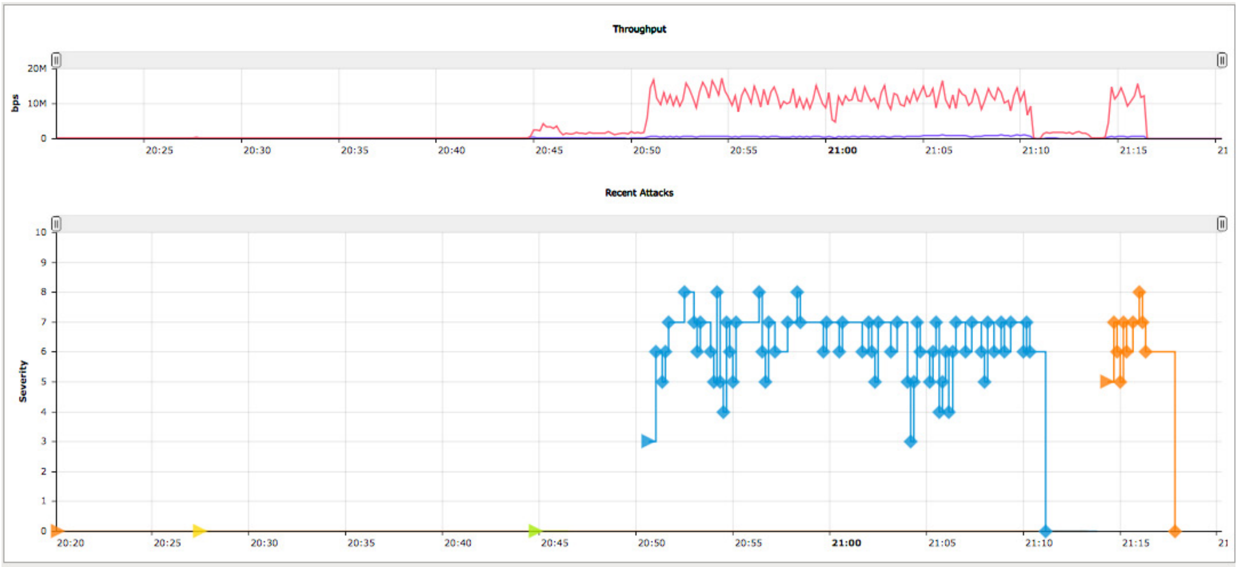
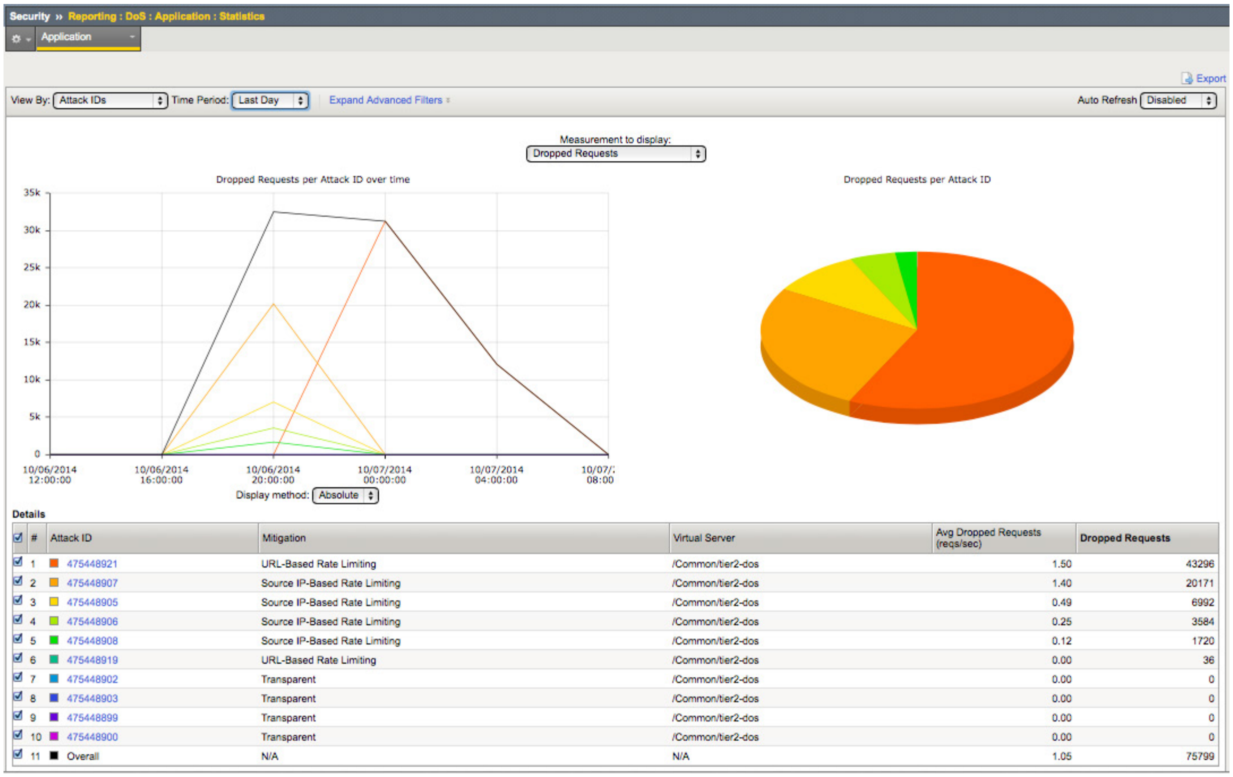


Figure 15: Real-time attack monitoring



View By: Attack IDs Time Period: Last Day

Auto Refresh Disabled

Measurement to display: Dropped Requests

Display method: Absolute

Details

☑	#	Attack ID	Mitigation	Virtual Server	Avg Dropped Requests (reqs/sec)	Dropped Requests
☑	1	■ 475448921	URL-Based Rate Limiting	/Common/tier2-dos	1.50	43296
☑	2	■ 475448907	Source IP-Based Rate Limiting	/Common/tier2-dos	1.40	20171
☑	3	■ 475448905	Source IP-Based Rate Limiting	/Common/tier2-dos	0.49	6992
☑	4	■ 475448906	Source IP-Based Rate Limiting	/Common/tier2-dos	0.25	3584
☑	5	■ 475448908	Source IP-Based Rate Limiting	/Common/tier2-dos	0.12	1720
☑	6	■ 475448919	URL-Based Rate Limiting	/Common/tier2-dos	0.00	36
☑	7	■ 475448902	Transparent	/Common/tier2-dos	0.00	0
☑	8	■ 475448903	Transparent	/Common/tier2-dos	0.00	0
☑	9	■ 475448899	Transparent	/Common/tier2-dos	0.00	0
☑	10	■ 475448900	Transparent	/Common/tier2-dos	0.00	0
☑	11	■ Overall	N/A	N/A	1.05	75799

Figure 16: Attack ID and mitigation reporting

3 More DDoS Recommended Practices

3.1 Mitigate DNS DDoS

DNS is the second-most-targeted service after HTTP. When DNS is disrupted, all external datacenter services (not just a single application) are affected. This single point of total failure, along with the historically underprovisioned DNS infrastructure, makes DNS a very tempting target for attackers. Even when attackers are not specifically targeting DNS, they often inadvertently do: if the attack clients are all querying for the IP of the target host before launching their floods, the result is an indirect attack against DNS.

Because of the relatively simple, UDP-based DNS protocol, a DNS attack is easy to generate, but difficult to defend against.

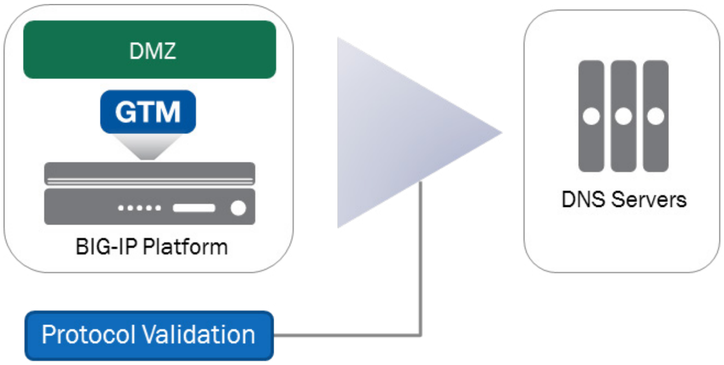


Figure 17: Mitigating DNS DDoS

There are four on-premises strategies for mitigating DNS DDoS attacks:

- Use Protocol Validation
- Detect and prevent DNS floods
- Overprovision DNS Services against NXDOMAIN query floods
- Blacklist as a last resort

3.1.1 Consider the Placement of DNS Services

You may notice that in Figure 18 below the DNS service exists as its own set of devices behind the security perimeter. Often DNS is served from this so-called DMZ between security tiers. This is done to keep DNS independent of the applications that it serves. For example, if that part of the datacenter goes dark, DNS can redirect requests to a secondary datacenter (or the cloud). **F5 recommends this strategy of keeping DNS separate** from the security and application tiers for maximum flexibility and availability.

Some large enterprises with multiple datacenters will go one step further and serve DNS outside the main security perimeter using a combination of F5's GTM DNS Express and the AFM firewall module. The main benefit of this approach is that the DNS services remain available even in the event that the network tier firewalls go offline due to a DDoS attack.

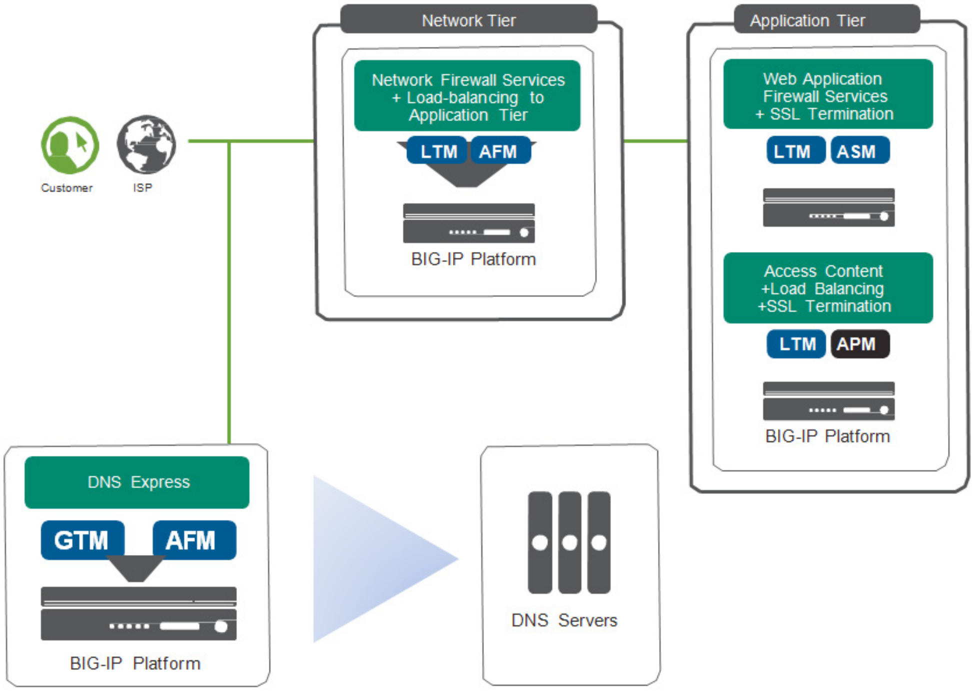

Network Tier
Network Firewall Services + Load-balancing to Application Tier
LTM
AFM
BIG-IP Platform
Application Tier
Web Application Firewall Services + SSL Termination
LTM
ASM
Access Content +Load Balancing +SSL Termination
LTM
APM
BIG-IP Platform
Customer
ISP
DNS Express
GTM
AFM
BIG-IP Platform
DNS Servers

Figure 18: Alternate external DNS architecture

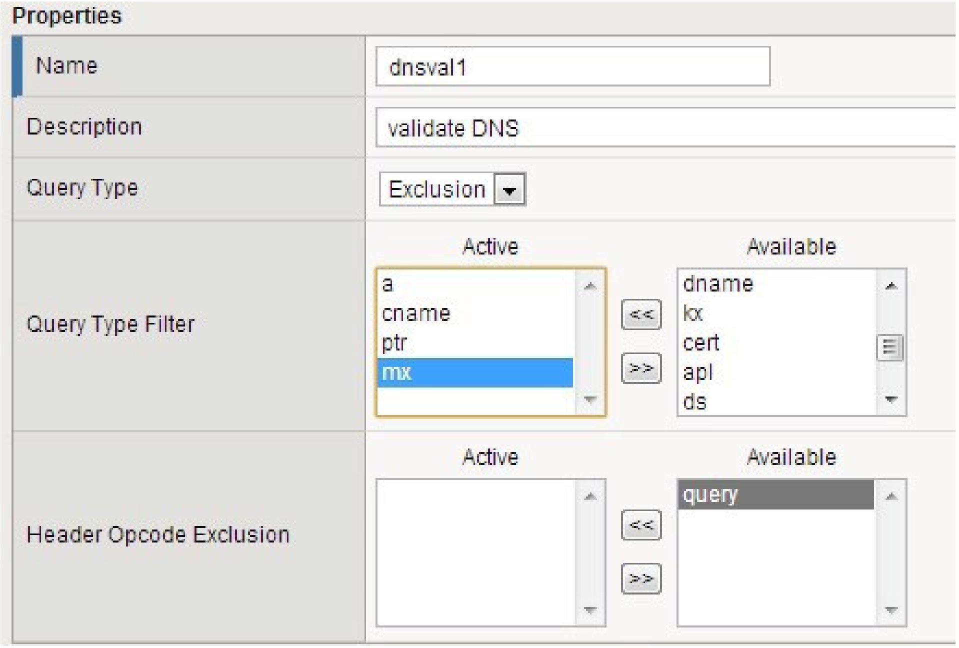
3.1.2 Use Protocol Validation to Protect DNS Services

Regardless of whether you serve DNS inside or outside the DMZ, you can use either GTM or AFM to validate the DNS requests before they hit the DNS server.

If you have GTM performing global server load balancing, it will likely be blocking many DNS DDoS attacks already. You can view the DNS query/response performance from the main dashboard in the GTM GUI. Because GTM is full-proxy for DNS, it will automatically validate every request and discard the invalid ones.

However, you may find that your servers are still overwhelmed by valid-looking requests. If you have the F5 firewall module AFM, you can use a **Protocol Security Profile** to further filter only specific types of DNS requests.

From the **Security** tab, select **Protocol Security** and then **Security Profiles**. Select **DNS** and press the **Create** button. At this screen you can build a protocol security profile to filter or block different types of requests.



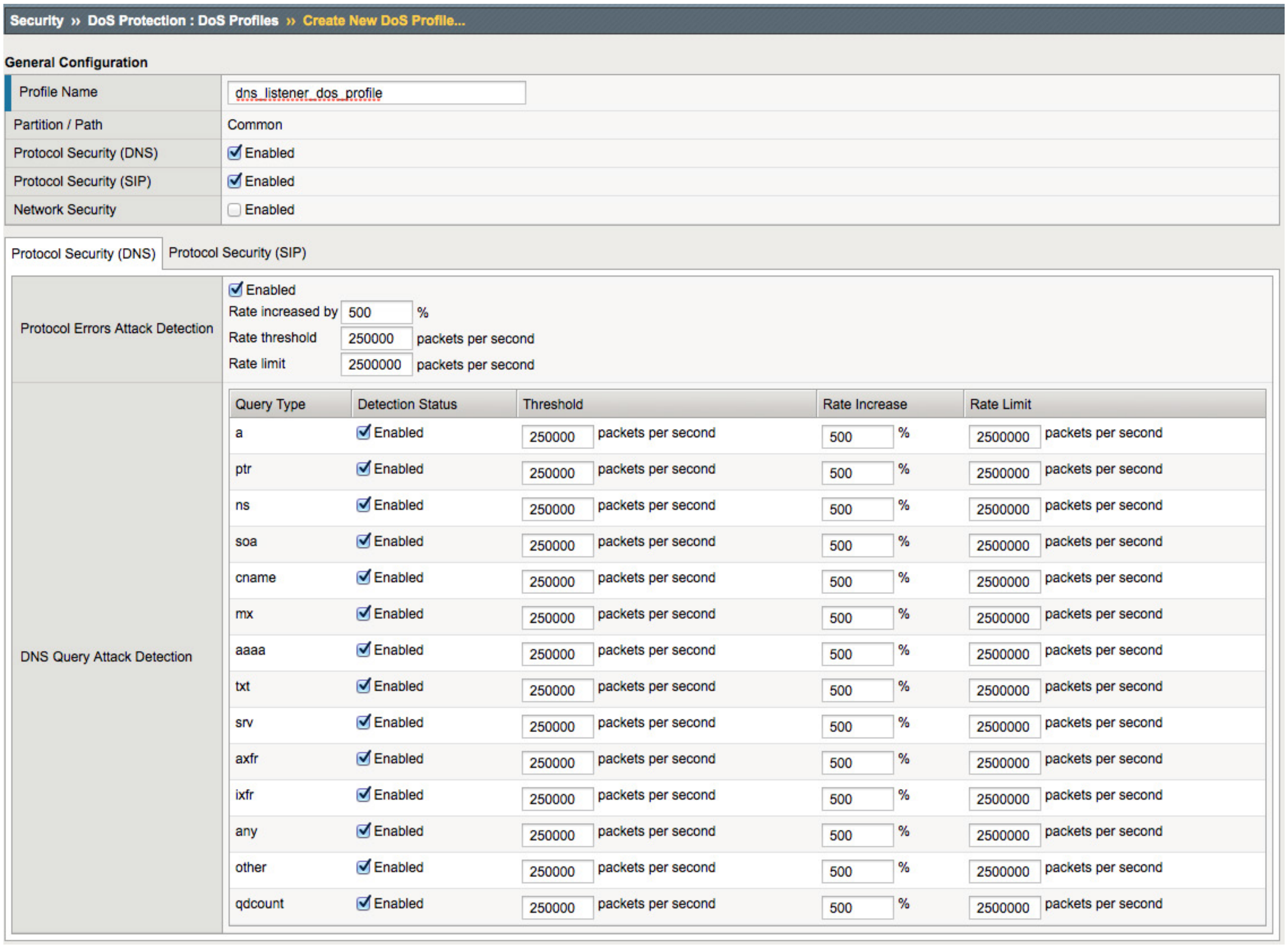
Properties

Name	dnsval1
Description	validate DNS
Query Type	Exclusion
Query Type Filter	Active: a, cname, ptr, mx Available: dname, kx, cert, apl, ds
Header Opcode Exclusion	Active: Available: query

Figure 19: DNS Protocol Validation

3.1.3 Detect and Mitigate DNS Floods

The F5 firewall module AFM has a powerful DNS DDoS function: it can detect DNS floods by record type. From the **Security** tab, select **DoS Protection**, then **DoS Profiles**, and finally **Create**. From the creation screen, click the checkbox for DNS and set and accept the threshold parameters.


Security » DoS Protection : DoS Profiles » Create New DoS Profile...

General Configuration

Profile Name	dns_listener_dos_profile
Partition / Path	Common
Protocol Security (DNS)	☑ Enabled
Protocol Security (SIP)	☑ Enabled
Network Security	□ Enabled

Protocol Security (DNS)
Protocol Security (SIP)

Protocol Errors Attack Detection
☑ Enabled
Rate increased by 500 %
Rate threshold 250000 packets per second
Rate limit 2500000 packets per second

DNS Query Attack Detection

Query Type	Detection Status	Threshold	Rate Increase	Rate Limit
a	☑ Enabled	250000 packets per second	500 %	2500000 packets per second
ptr	☑ Enabled	250000 packets per second	500 %	2500000 packets per second
ns	☑ Enabled	250000 packets per second	500 %	2500000 packets per second
soa	☑ Enabled	250000 packets per second	500 %	2500000 packets per second
cname	☑ Enabled	250000 packets per second	500 %	2500000 packets per second
mx	☑ Enabled	250000 packets per second	500 %	2500000 packets per second
aaaa	☑ Enabled	250000 packets per second	500 %	2500000 packets per second
txt	☑ Enabled	250000 packets per second	500 %	2500000 packets per second
srv	☑ Enabled	250000 packets per second	500 %	2500000 packets per second
axfr	☑ Enabled	250000 packets per second	500 %	2500000 packets per second
ixfr	☑ Enabled	250000 packets per second	500 %	2500000 packets per second
any	☑ Enabled	250000 packets per second	500 %	2500000 packets per second
other	☑ Enabled	250000 packets per second	500 %	2500000 packets per second
qdcount	☑ Enabled	250000 packets per second	500 %	2500000 packets per second

Figure 20: DNS flood protection

The Protocol Errors checkbox means that the system detects malicious or malformed DNS queries, and it displays, in percentage, how much of an increase in DNS query traffic is legal before the system tracks malformed and malicious DNS queries. Each specific query type can then be assigned its own rate limit.

3.1.4 Overprovision DNS Services Against Query Floods

DNS services have been historically underprovisioned. Part of the reason for this is that for many organizations the ownership of DNS has not been a positive development for any particular team. Whatever the real reason, a significant percentage of DNS deployments are underprovisioned to the point where they are unable to withstand even small or midsize DDoS attacks.

F5's recommended way to remedy this is to front the DNS service with the special, high-performance DNS proxy module called DNS Express. DNS Express acts as an absolute resolver in front of the existing DNS servers. It loads the zone information from the servers and then resolves every single request or returns NXDOMAIN. It is not a cache and cannot be emptied via NXDOMAIN query floods.

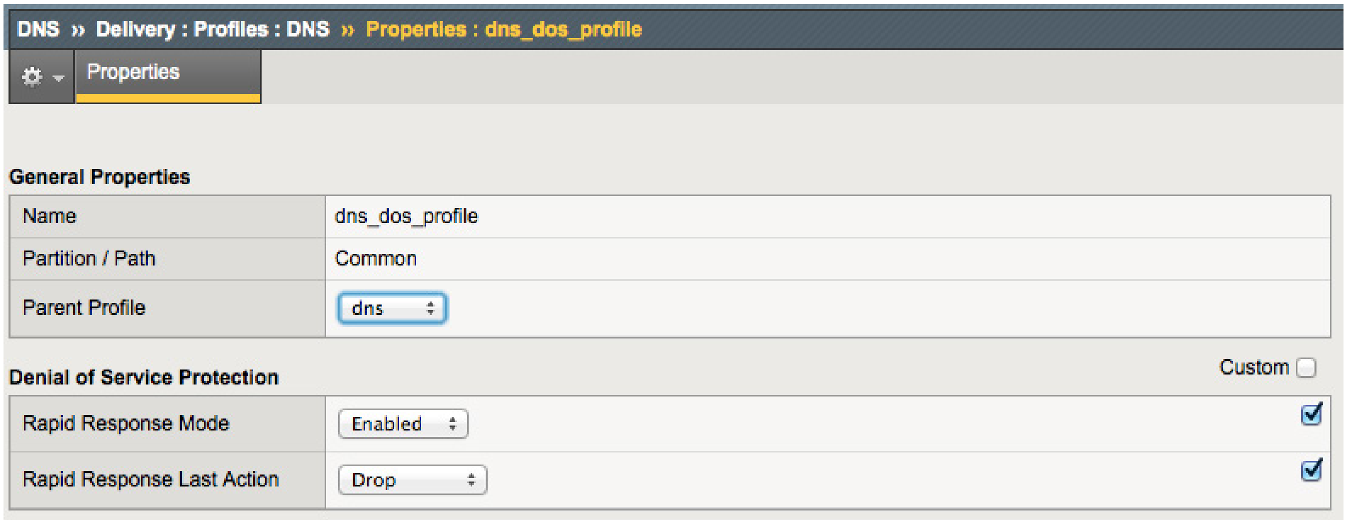
In GTM or DNS Services, DNS Express can serve 250,000 requests per second per CPU and is therefore resistant to all but the most virulent DNS attacks. The DNS servers remain in place to manage the zone data.

3.1.5 Enable DNS Rapid Response

One of the most effective ways of mitigating a DNS attack is to absorb it—that is, take in the request and if the request is valid respond as quickly as possible and silently drop invalid requests. F5 allows you to configure DNS Rapid Response for DNS Express within the DNS profile. This feature allows requests for DNS Express–enabled zones to be processed at up to double the performance possible in previous versions.

However, you must be aware of a couple of considerations before you can enable DNS Rapid Response. The DNS profile with Rapid Response requires UDP as the protocol and Auto Last Hop to be enabled. Additionally, BIG-IP platforms using vCMP and Virtual Edition do not support DNS Rapid Response.

Due to the processing order for DNS Rapid Response, all other DNS features are disabled. These disabled features include DNS iRules, software DoS functions, DNSSEC, DNS Cache, and DNS6-4 with exceptions for DNS Express and Global Server Load Balancing (GSLB), unless the Rapid Response Last Action is set to Allow. And then it is only applicable to DNS Queries that did not match DNS Express zones.



DNS » Delivery : Profiles : DNS » Properties : dns_dos_profile

General Properties

Name	dns_dos_profile
Partition / Path	Common
Parent Profile	dns

Denial of Service Protection Custom ☐

Rapid Response Mode	Enabled	☑
Rapid Response Last Action	Drop	☑

Figure 21: DNS Rapid Response configuration

To configure DNS Rapid Response on an existing DNS profile, issue the following command to enable the setting and the action to take when a query does not match a DNS Express Zone:

```
% tmsh modify ltm profile dns dns_dos_profile enable-rapid-response yes rapid-response-last-action drop
```

The default action of drop will ensure the most expeditious processing of invalid queries while under a DDoS attack.

Next, the profile must be assigned to the listener:

```
% tmsh modify gtm listener listener profiles replace-all-with { dns_dos_profile udp_gtm_dns }
```

Validation of the DNS Rapid Reponse configuration and observation of the statistics associated with the actions for valid and invalid traffic can be done through tmsh.

```
 % tmsh show ltm profile dns f5_dos

-------------------------------------------
 Ltm::DNS Profile: f5_dos
-------------------------------------------
<…truncated…>
 Rapid Response
   Queries                        4004
   Responses                      3803
   Last Action
     Allowed                         0
     Drops                         201
     Truncated (TC)                  0
     No Name (NXDOMAIN)              0
     No Error                        0
     Refused                         0
```

3.1.6 Blacklist as a Last Resort

DNS traffic is traditionally UDP, which is easy to generate and easy to spoof. Conventional layer 3 and 4 defenses, such as blacklisting by source IP, are usually ineffective against a DNS flood. In fact, blocking DNS requests by source IP can be downright dangerous. For example, if you unknowingly block requests from a major ISP, you may deny service to many legitimate users without realizing it.

See “Detecting and Preventing DNS DDoS Attacks” in the BIG-IP Systems: DOS Protection and Protocol Firewall Implementations manual.

3.1.7 Beware of DDoS Attack Participation

3.1.7.1 Unused Query Types

An attacker can trick a DNS service into bombing a third-party target by sending queries for unused services. Use the AFM screens (see Figure 20 above) to disable query types that you aren’t using. Then, queries that come in for these types will be dropped. No response will be provided, thereby helping to avoid participation in a DDoS attack.

This is especially true for MX (mail services) and zone transfers. If your organization does zone transfers at known, specific times, keep the IXFR, AXFR, and ZXFR types disabled at all other times.

3.1.7.2 DNSSEC

DNSSEC is an important evolution in the global domain name service. Ultimately it will cut down on deceptive practices such as phishing. The picture is more complicated for DNS DDoS. DNSSEC responses are sometimes 10–20 times larger than traditional DNS UDP responses. This means that DNSSEC servers are actually tricked into attacking other computers by inadvertently bombarding them with invalid responses.

With GTM, F5 has the highest-performing DNSSEC solution on the market. The capacity of GTM could make for a potent weapon if it were to be used as an attack vector. Therefore, GTM allows you to rate-limit the number of responses to prevent itself from participating in an attack.

```
% tmsh modify sys db dnssec.maxnsec3persec value 10
```

The **dnssec.maxnsec3persec** variable controls the upper limit of NSEC3 authoritative NXDOMAIN messages that GTM will send per second. 0 is unlimited and the default. A more restrictive value, such as between 10 and 100 per second, may prevent GTM itself from being used during an attack.

```
% tmsh modify sys db dnssec.signaturecachensec3 value true
```

Setting the **dnssec.signaturecachensec3** variable to false prevents NXDOMAIN messages from using the GTM cache at all, thus preventing an attacker from filling GTM's cache with "no such domain" responses.

3.2 Additional DDoS Best Practices Preparation Procedures

The time spent preparing for a DDoS attack will increase the effectiveness of your defense. Here are a few more ways that you can prepare your organization for such an attack.

3.2.1 Configure and Verify Logging

During an attack there is a good chance that you will be sending diagnostics and logging anomalies, and traffic spikes. High performance is critical when dealing with a large DDoS attack. Instrumentation is important as well, meaning that you will want to use the High-Speed Logging facility of BIG-IP to send this information to a third-party logging device such as Splunk or a SIEM such as ArcSight.

Note: You **must** use the High-Speed Logging facilities of BIG-IP at the network tier when mitigating a DDoS attack. Do not use the local logging because an intense DDoS attack can overwhelm the local disk-based logging.

3.2.1.1 Set Up High-Speed Logging

1. Create a pool to map to your external log servers (in this case they are syslog). Rewrite as necessary for ArcSight, TrustWave, or whichever SIEM solution your environment supports. Then create the log config objects to format and forward the strings properly:

```
%   tmsh create ltm pool hsl_pool mem bers  add {  10.128.10.250:514 }
%   tmsh create sys log-config destination remote-high-speed-log log_dest_HSL {
pool-name hsl_pool }
%   tmsh create sys log-config destination remote-syslog log_dest_format { format
rfc5424 remote-high-speed-log log_dest_HSL }
%   tmsh create  sys log-config publisher log_pub_ddos {  destinations {  log_dest_HSL log_dest_format }  }
```

2. Create a log profile object using the GUI.

   Access the **Security** > **Event Logs** > **Logging Profiles** page. Create a log profile using the following:

Profile Name	ddos_log_profile
Network Firewall	Enabled
Network Firewall: Publisher	log_pub_ddos
Log Rule Matches	Accept, Drop, and Reject
Log IP Errors	Enabled
Log TCP Errors	Enabled
Log TCP Events	Enabled
Storage Format	field-list Select all Available Items and move them to the Selected Items list

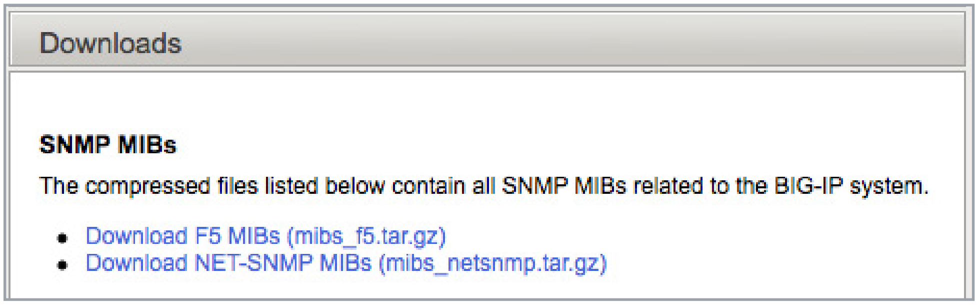
3. Associate that log profile object with the virtual servers protecting your application.

```
%   tmsh modify /ltm virtual vip1 {  security-log-profiles add {  ddos_log_profile }  }
```

3.2.2 Using SNMP to Report and Track DDoS Attacks

SNMP is a primary component of network management solutions. SNMP can now be used to gather statistics for DoS attacks on devices and monitor the performance of device eviction policies configured in section 2.3.9. This can be especially useful to provide an overall picture, particularly in larger networks that may span multiple BIG-IP devices, allowing you to identify and manage the overall DoS threat.

Beyond the standard monitoring, you should be tracking on BIG-IP devices for CPU, memory, and network throughput. Configure your management tool to use the F5-BIGIP-LOCAL-MIB that can be extracted from the mibs_f5.tar.gz file available for download from the Welcome Page on any BIG-IP. Within the mib are additional DoS-specific statistics that should be collected.



Downloads

SNMP MIBs

The compressed files listed below contain all SNMP MIBs related to the BIG-IP system.

- Download F5 MIBs (mibs_f5.tar.gz)
- Download NET-SNMP MIBs (mibs_netsnmp.tar.gz)

Figure 22: Welcome Page downloads section for SNMP MIBs

The following table contains examples of the DoS-specific statistics available via SNMP data collection:

`ltmDosAttackDataStat`	`ltmFlowEvictionPolicyStat`
• Device Name • Vector Name/ID • Attack Detected • Dropped Packet Counter (sec/min/hr) • Total Packets Matched (sec/min/hr) • Whitelist Counter	• Eviction Policy Name • Context Type and Name • Evicted Flow Counter

3.2.3 Reconnoiter Your Own Applications

Modern DDoS attackers will reconnoiter an application days or weeks before launching their DDoS attack. They will spider your website and retrieve the **load-time** and **data-size** for each valid URI. By sorting the resulting dataset, they will quickly isolate your most CPU- or database-intensive queries and your largest objects (such as PDFs and MP4s). During the DDoS attack they will repeatedly query for these objects, overwhelming your infrastructure.

Though section 2.4 will help you mitigate that attack when it happens, you can help yourself beforehand by reconnoitering your own applications. This will give you advanced visibility into what URIs and subsystems will be likely targets, allowing you to make more-informed triage decisions later.

Ideally you have a tool like LoadRunner or another performance-monitoring tool that can provide you with the metrics you need. If you lack this capability, perhaps the simplest way to retrieve the basic table of URL, load-time, and data-size is to run the wget utility, which is available on most Linux distributions. Run it with the following syntax:

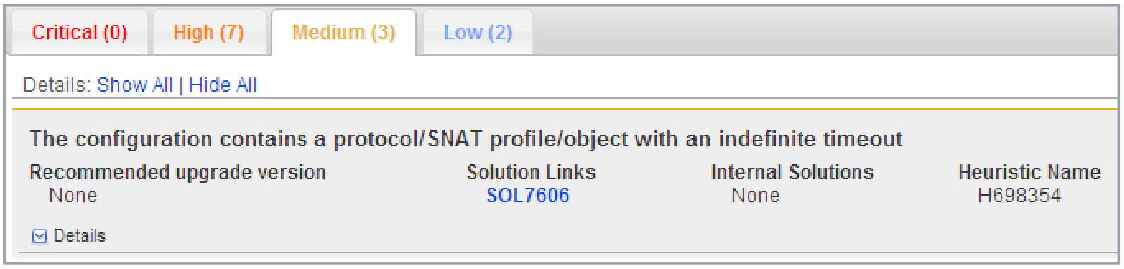
```
%   wget -r --spider http://10.128.10.150 2>&1 | grep saved
2013-08-25 15:44:29 (2.48 MB/s) - `10.128.1.150/index.html’ saved [22304]
2013-08-25 15:44:39 (5.53 MB/s) - `10.128.1.150/index.php’ saved [22304]
2013-08-25 15:44:41 (7.06 MB/s) - `10.128.1.150/sell.php’ saved [41695]
```

The last number (in square brackets) is the data-size of the request. To get the load-time you will have to subtract the times (second field) from the time of the previous request.

3.2.4 Validate the Health of Existing BIG-IP Devices with iHealth

F5 provides a cloud-based diagnostics and heuristics service called iHealth. iHealth will examine an F5 device’s configuration and make recommendations to keep BIG-IP fast, secure, and available. While the majority of the settings may be more applicable to fast and secure, some of these settings can apply to availability and, by extension, to DDoS resilience.

In this example, iHealth is showing that a SNAT pool has been configured without timeout values. This can be a reminder to resource-conscious administrators to ensure that the SNAT pool used for their hardened virtual servers should include idle timeouts to keep the number of connections down and prevent an upstream firewall from tipping.



The configuration contains a protocol/SNAT profile/object with an indefinite timeout

Recommended upgrade version	Solution Links	Internal Solutions	Heuristic Name
None	SOL7606	None	H698354

Figure 23: iHealth reporting

See the iHealth website for more information about iHealth.

3.2.5 Prepare a DDoS Playbook

A DDoS Playbook or Runbook is a procedural manual to assist your IT employees in combatting a DDoS attack. A good playbook will help administrators combat a DDoS attack. The Playbook should be kept current with updated whitelists and contact information.

A few organizations will perform periodic DDoS drills (or even tests) against themselves to keep current and to test the playbook. Try to have your staff practice the procedures from the playbook when key people are not present—attacks do not always happen at the most convenient time.

If you do not have a playbook, contact F5 for one.

3.2.6 Review Defensive Tactics in the Two-Tier Architecture

Some of the defensive tactics described in the previous sections are worth reviewing, especially for administrators using a non-F5 network firewall.

Remember to:

- Configure SNAT pools to mitigate port exhaustion at the network tier.
- Shape traffic at the network tier.
- Aggressively reap TCP connections.
- Blacklist DNS only as a last resort.
- Implement login-walls and CAPTCHAs at the application tier.
- Disable optional CPU-intensive features at the application tier.
- Always use high-speed remote logging.

By implementing the proposed suggestions in this Best Practices guide, you will have done a lot to safeguard your applications against a DDoS attack.

4 Conclusion

Understanding the modern threat spectrum of denial-of-service attacks is important. Understanding how to make use of the defensive equipment that you already have is even more important.

Depending on your resources and needs, you may already have redesigned your network for DDoS resilience. If this is something that you are considering, then pay close attention to the recommended multi-tier architecture described in section 2.1. Even if your network security is not completely built from F5 technology, it still makes sense to tackle layer 4 and layer 7 independently for DDoS purposes.

Architecture
Recommended Practices
Process

By following the recommended practices, you will be preparing your network, applications, and people to be attack-resistant.

The final step in F5's recommended practices for DDoS mitigation is to prepare a DDoS playbook. Such a playbook is a real-time procedural guide for mitigating an attack that includes worksheets and logs. F5 can provide you with a template to get started.

Experts predict that DDoS will be an issue on the Internet for a long time to come. Soon, being prepared will not just be an option but a requirement.

Appendix

Application Attack Taxonomy and Countermeasures

The following section recommends mitigations for specific layer 7 attack vectors. Many of these are the “slow-and-low” type of attacks that can be particularly pernicious.

Contents

Slowloris

Slowloris is a common HTTP vector where an attacker will (very slowly) send small HTTP headers to keep the HTTP session alive (e.g., "**X-a: b**" every 299 seconds). If the virtual server is currently load-balancing at layer 4, consider switching it to layer 7. This will add some native protection when the HTTP profile is added.

```
% tmsh list ltm virtual vip1 ltm virtual vip1 {
      destination 10.128.10.141:http
      profiles {
fastL4 {  }
    }
}
%   tmsh modify ltm virtual vip1 profiles replace-all-with {  tcp http  }
```

This will cause BIG-IP to absorb the Slowloris connections. If you become concerned that too many are building up and causing problems for other devices (such as a firewall), use the following Slowloris iRule to drop any connection that has not completed after 10 seconds (feel free to adjust this number).

```
# Slowloris iRule
when CLIENT _ ACCEPTED  {
      set hsl [HSL::open -proto UDP  -pool hsl _ pool]
set rtimer 0 after 10000 {
              if {  not $rtimer }  {
drop
                      HSL::send $hsl "Dropped [IP::client _ addr] - connection too slow"
              }
      }
}
when HTTP _ REQUEST  {
set rtimer 1
}
```

Keep Dead

This attack is based on consuming CPU and RAM. By using Keep-Alive and the HTTP HEAD method, it can create a flood of requests without triggering a firewall defense that is based on the number of connections opened to the server.

The ASM module can disallow HEAD requests (which are not typically used by browsers). You can reject HEAD requests by configuring the "Allowed Methods" in the application security policy in question.

See solution 12312 for more information.

Low Orbit Ion Cannon (LOIC)

The Low Orbit Ion Cannon is a voluntary botnet tool closely associated with the hacktivist group Anonymous. While the tool uses SYN floods and UDP floods, it is most famous for its layer 7 HTTP floods. Assuming that the SYN and UDP floods have been mitigated (see sections 2.2.2 and 2.2.3), the last step is to mitigate the LOIC GET floods.

Often the fastest way to do this is to filter it on the attack “protest message” included with each LOIC HTTP request. Use Wireshark or tcpdump, or another tool, to isolate the message, and then add that message to a datagroup. Use %**20** to represent spaces. The message may change over time, and you may need to monitor it for the duration of the attack.

```
ltm data-group anonmsgs {
records {
        Somos%20legi {   }
        U%20dun%20goofed {   }
    }
    type string
}
```

Note that you can use external data classes that are hosted outside the BIG-IP—see “**help search data-group**” in the tmsh command shell.

Then use a simple scrubber iRule to drop requests that contain any payloads in that data class:

```
ltm rule loic _ defense _ rule {
    when CLIENT _ ACCEPTED  {
             set hsl [HSL::open -proto UDP  -pool hsl _ pool]
    }
    when HTTP _ REQUEST  {
    if {  [class match [HTTP::uri] contains anonmsgs] }  {
drop
      HSL::send $hsl “Dropped [IP::client _ addr] -  suspected Low  Orbit 
Ion Cannon”
    }
  }
}
```

Slow-POSTs

The heart of the Slow-POST attack relies on sending a POST request with given "content-length," which is typically a large number, and then very slowly sending the message body to the server, while keeping the idle time long. The server will leave the connection open as it continues to receive data. If a large number of these requests are executed against a server, there is potential for exhausting the connection table, which would leave the server unable to respond to further requests.

If you have the ASM module, you can mitigate slow posts with two of the variables found in the ASM system variables screen—navigate to **Security** : **Options** : **Application Security** : **Advanced Configuration** : **System Variables**—and modify the following variables:

`slow _ transaction _ timeout` (defaults to 10 seconds). Lower this value as needed.

`max _ slow _ transactions` (defaults to 25 transactions). Lower this value to 5 or less as needed. If you do not have ASM, then see this LTM iRule to mitigate Slow-POST. It can be used with the Slow-read iRule (just attach them as two separate iRules) because the Slow-read iRule is server-based and the Slow-POST iRule is client-based.

Zero Window Attacks

The Zero Window attack is a difficult-to-detect layer 4 attack. It works by establishing a TCP connection to the target, requesting some data, and then setting the TCP window size to zero. This stalls the connection at the server, cache, or middleware.

If the attacker is setting a TCP zero window length against a BIG-IP, you can use the zero-window- timeout tcp profile value mentioned in section 2.2.2 to mitigate.

Slow-Read Attack

The Slow-Read attack works by sending legitimate HTTP requests and then reading the HTTP responses very slowly from the buffer, aiming to keep as many connections as possible in an active state on the victim.

In BIG-IP 11.3.0, the ASM module’s low-and-slow prevention works on inbound requests such as a Slow POST. For Slow-read, use the following LTM iRule mitigation:

```
when SERVER_CONNECTED  {
TCP::collect
}

when SERVER_DATA  {
set rtimer 0

# Time in milliseconds before HTTP response read is considered slow:
    after 5000 {
        if {  not $rtimer} {
            set hsl [HSL::open -proto UDP  -pool hsl_pool]

# Slow read detected for this server response. Increment the count by adding a
table entry:
# Add the client source IP::port to the subtable with a timeout
            table set -subtable “MyApp” “[IP::client_addr]:[TCP::client_port]” “ignored” 180

#  If we are over the concurrency limit then reject
            if {  [table keys -subtable “MyApp” -count] >  5}   {
                clientside {reject}
                table delete -subtable “MyApp” “[IP::client_addr]:[TCP::client_port]”
                HSL::send $hsl “Dropped [IP::client_addr] –  reading too slow”
            }
        }
    }

    TCP::notify response TCP::release TCP::collect
}

when USER_RESPONSE  {
set rtimer 1
}

when CLIENT_CLOSED  {
    table delete -subtable “MyApp” “[IP::client_addr]:[TCP::client_port]”
}
```

RUDY

R-U-Dead-Yet (RUDY for short) uses Slow-POST and a generic HTTP DoS attack via long-form field submissions. See section 4.4 for the mitigation for the Slow-POST attack.

Apache Killer

The Apache Killer is also known as a Range Attack. When a client browser (such as a mobile handset browser) needs just part of a document, it can request a “range” of the data with an HTTP range header. If the client wants just the first 100 bytes, it could say:

```
Range:bytes=0-100
```

The Apache Killer attack works by requesting multiple, overlapping ranges that confuse web servers like Apache:

```
Range:bytes=0-,5-1,5-2,5-3,…
```

There are three ways to mitigate Apache Killer. You can modify the HTTP profile to simply remove the Range header. For example, if your http profile was named “http_ddos2,” you would run this command:

```
%   tmsh modify ltm profile http http _ ddos2 { header-erase range }
```

A more surgical way to mitigate Apache Killer is with the following iRule, which only removes range requests when more than five ranges are requested.

```
when CLIENT _ ACCEPTED  {
    set hsl [HSL::open -proto UDP  -pool hsl _ pool]
}
when HTTP _ REQUEST  {
    # remove Range requests for CVE-2011-3192 if more than five ranges are requested
    if {  [HTTP::header “Range”] matches _ regex {bytes=(([0-9\- ])+,){5,}} }  {
        HTTP::header remove Range
        HSL::send $hsl “Client [IP::client _ addr] sent more than 5 ranges. Erasing range header.”
    }
}
```

The third method of mitigation using BIG-IP solutions is to use the following ASM attack signature to detect and act upon an attack using this technique:

```
pcre:”/Range:[\t ]*bytes=(([0-9\- ])+,){5,}/Hi”;
```

SSL Renegotiation

If you are seeing a lot of renegotiations happening from specific SSL clients, you might be suffering an SSL renegotiation attack. The easiest way to mitigate it is to disable SSL renegotiation from the virtual server’s associated clientssl profile. However, if you need to support renegotiation for legitimate clients (such as old “step-up” or server-gated-cryptography browsers) while still mitigating the attack, you can use the following iRule, or others like it. This rule closes any connection that attempts more than five renegotiations in a minute:

```
when RULE _ INIT {
    set static::maxquery 5
    set static::mseconds 60000
}
when CLIENT _ ACCEPTED {
    set ssl _ hs _ reqs 0
    set hsl [HSL::open -proto UDP  -pool hsl _ pool]
}
when CLIENTSSL _ HANDSHAKE {
    incr ssl _ hs _ reqs
    after $static::mseconds { if {$ssl _ hs _ reqs > 0} {incr ssl _ hs _ reqs -1} }
    if { $ssl _ hs _ reqs > $static::maxquery } {
after 5000
drop
        HSL::send $hsl “Dropped [IP::client _ addr] -  too many SSL  renegotiations”
    }
}
```

Dirt Jumper iRule

Certain versions of the Dirt Jumper tool do not include a // in their referrer field. Here is a simple iRule to detect and drop Dirt Jumper connections:

```
when CLIENT _ ACCEPTED  {
    set hsl [HSL::open -proto UDP  -pool hsl _ pool]
}
when HTTP _ REQUEST  {
 if {  [HTTP::header exists “Referer”] }  {
   if {  not ([HTTP::header “Referer”] contains “\x2F\x2F”) }  {
     HSL::send $hsl “DDoS Dirt-Jumper HTTP Header Structure missing x2f x2f Referer protocol
identifier from [IP::client _ addr]”
drop
   }
 }

}
```

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

Americas
info@f5.com

Asia-Pacific
apacinfo@f5.com

Europe/Middle-East/Africa
emeainfo@f5.com

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

Solutions for
an application world.