ACOS 6.0.8

A10 Next Generation Web Application Firewall Configuration Guide

December, 2025

Information in this document is subject to change without notice.

PATENT PROTECTION

A10 Networks, Inc. products are protected by patents in the U.S. and elsewhere. The following website is provided to satisfy the virtual patent marking provisions of various jurisdictions including the virtual patent marking provisions of the America Invents Act. A10 Networks, Inc. products, including all Thunder Series products, are protected by one or more of U.S. patents and patents pending listed at: a10-virtual-patent-marking.

TRADEMARKS

A10 Networks, Inc. trademarks are listed at: a10-trademarks

CONFIDENTIALITY

This document contains confidential materials proprietary to A10 Networks, Inc. This document and information and ideas herein may not be disclosed, copied, reproduced or distributed to anyone outside A10 Networks, Inc. without prior written consent of A10 Networks, Inc.

DISCLAIMER

This document does not create any express or implied warranty about A10 Networks, Inc. or about its products or services, including but not limited to fitness for a particular use and non-infringement. A10 Networks, Inc. has made reasonable efforts to verify that the information contained herein is accurate, but A10 Networks, Inc. assumes no responsibility for its use. All information is provided "as-is." The product specifications and features described in this publication are based on the latest information available; however, specifications are subject to change without notice, and certain features may not be available upon initial product release. Contact A10 Networks, Inc. for current information regarding its products or services. A10 Networks, Inc. products and services are subject to A10 Networks, Inc. standard terms and conditions.

ENVIRONMENTAL CONSIDERATIONS

Some electronic components may possibly contain dangerous substances. For information on specific component types, please contact the manufacturer of that component. Always consult local authorities for regulations regarding proper disposal of electronic components in your area.

FURTHER INFORMATION

For additional information about A10 products, terms and conditions of delivery, and pricing, contact your nearest A10 Networks, Inc. location, which can be found by visiting www.a10networks.com.

Table of Contents

Getting Started

A10 Networks' Next Generation Web Application Firewall (NGWAF) is implemented based on A10 ADC capabilities empowered by Fastly (a two-time WAF Gartner visionaries' vendor) technology for identification and mitigation of attacks and bad signals.

This document provides information on NGWAF and the step-by-step procedure to configure it.

The following topics are covered in this section:

Overview

A10 Next-Gen WAF (NGWAF) is an application security monitoring system that monitors suspicious and anomalous web traffic and protects against attacks directed at applications and origin servers. It provides superior protection for applications and APIs by delivering the following advantages over legacy WAF:

- **Performance** - Provides superior performance and lightning fast WAF decisions (within less than 2ms).
- **Deployment Time** – Requires less deployment time, has excellent default policies, and needs just one click to enable the blocking mode.
- **Almost zero false positives** - Enables you to quickly turn on NGWAF in the blocking mode.
- **Advanced rate limiting** – Controls the number of requests from potential threats.
- **API and Microservice Protection**
  - **API Brute Forcing Protection** - Identifies and blocks brute forcing sensitive IDs or tokens in APIs attacks such as Unique Identifier Enumeration.
  - **Unauthorized API Access Prevention** - Protects fraud gift card and credit card validations, attempts to obtain patient healthcare records, and more.
  - **API Abuse Mitigation** - Mitigates potential attacks aiming to abuse sign-up systems, emails, and other sensitive actions.
- Provides protection against the following attacks:
  - OWASP Top 10
  - Advanced Web attacks
  - Malicious bots
  - DDoS attacks
  - Volumetric attacks
  - Account Takeover (ATO) and Credential Stuffing - Prevents attackers from using known lists of compromised credentials and breach data dumps for illegal access.

- Blocks bad actors from known malicious sources or disallowed countries / geographies.
- Needs minimal tuning, troubleshooting and maintenance. As a result, dedicated security administrators are not required.

Features Supported in ACOS

A10 Next-Gen WAF (NGWAF) is a module of the HTTP reverse proxy and provides the following features in ACOS:

- Supports the HTTP and HTTPS virtual ports
- Supports the HTTP and HTTP2 proxy
- Supports SSL Onloading and Offloading
- Supports AAM module for authentication. However, the requests are sent to NGWAF only after completing the authentication process.
- Delivers data to Fastly Cloud dashboard for monitoring the attack request records and modifying the NGWAF policies.
- Provides syslogs and debug logs to aid debugging and troubleshooting.
- Provides health monitoring feature to monitor correct NGWAF functioning.

Architecture

A10 Next-Gen WAF (NGWAF) services can be enabled on each partition and comprises of the following three components:

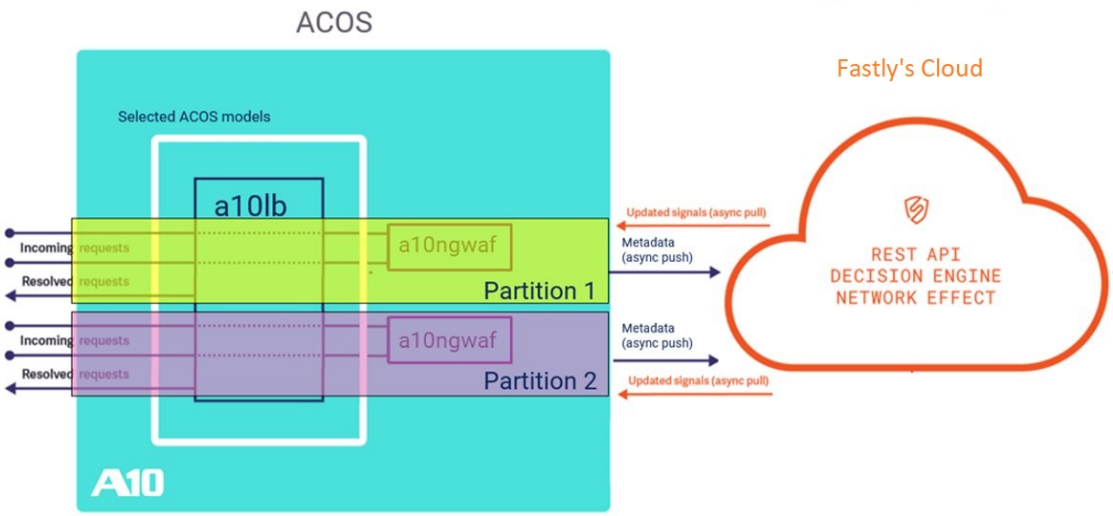
- **A monitoring agent (a10ngwaf)** - It is a small daemon process that analyses and decides whether the inbound requests should be permitted to continue or whether some action should be taken. It also uploads processed request data to the cloud analysis system and downloads new rules and configurations on ACOS.
- **An engine module (a10lb)** – It passes the request data to the monitoring agent for processing.
- **A cloud analysis system (Fastly Cloud)** – The anomalous request data is collected locally and uploaded to the cloud analysis system to perform out-of-band analysis

of inbound traffic. The output from the cloud system is used by the agent locally to make better blocking decisions.

NOTE: Data traffic is not sent to the Fastly cloud; only the information/metadata on the attacks and anomalous requests data is sent. Additionally, the sensitive data (e.g., cookies, form fields) is never sent to the Fastly cloud; it is handled by the monitoring agent.

The following figure demonstrates the NGWAF architecture and workflow.

Figure 1 : ACOS NGWAF Architecture and Workflow



The ACOS NGWAF follows the workflow described below:

1. The Fastly agent runs on one or more control CPU's within ACOS and exposes a Unix Domain Socket.
2. ACOS (a10lb) connects to this listening socket and sends the client requests for inspection to the NGWAF agent (a10ngwaf).
3. When the agent receives a request from the module, it runs through the rule-set and decides whether the request should be forwarded or blocked.
4. The agent then sends the request and its decision back to the module.
5. Additionally, after determining if the request contains an attack, the sanitized and redacted portion of the request is marked as attack or anomaly and then sent to the Fastly cloud.

NOTE: If NGWAF is not operational, the requests are forwarded to the destination without inspection.

Platforms Supported

A10 Next-Gen WAF (NGWAF) is supported on the following A10 devices:

- Thunder Platforms (physical appliances):
  - Thunder 3350-E, Thunder 3350, Thunder 3350S
  - Thunder 1060S
  - Thunder 1040-F, Thunder 1040QSSL
  - Thunder 5840, Thunder 5840S
  - Thunder 5840-11, Thunder 5840-11S
  - Thunder 5440, Thunder 4440
  - Thunder 6655S
  - Thunder 7655S (multi-PU)
- Multi-tenant Virtual Platform (Thunder MVP) deployed on:
  - Dell OEM R640
  - Dell OEM R740
  - Dell OEM R760
- Virtual Thunder (vThunder) on VMware ESXi 6.5,7 and KVM with the following bandwidth licenses:
  - 20 Gbps
  - 40 Gbps
  - 100 Gbps

NOTE: For the above-listed devices, NGWAF is currently supported only for ACOS versions 6.0.0 TR, 6.0.0-P1, 6.0.1 GA, 6.0.2 and higher.

ACOS and Fastly Agent Compatibility

For all ACOS releases, A10 Next-Gen WAF is tested with the Fastly agent (also referred to as NGWAF agent). The tested versions are listed in the following table:

ACOS Releases	NGWAF Agent Versions
6.0.2	4.45.0
6.0.2-P1	
6.0.3	4.49.0
6.0.3-P1	4.51.0
6.0.3-P2	
6.0.4	4.53.0
6.0.6	4.61.0
6.0.7	4.64.0

Limitations

Following are limitations of A10 Next-Gen WAF (NGWAF) in ACOS:

- Maximum 8 partitions can be enabled.
- One partition can have only one NGWAF agent. Therefore, all the virtual ports on a partition share the same statistics on the Fastly Dashboard.
- NGWAF does not have a learning mode.
- NGWAF does not examine HTTP responses.
- ACOS cannot connect to the Fastly Cloud portal via L3V data interface. Please check Prerequisites to configure NGWAF correctly.

NOTE: For Fastly NGWAF limitations, refer to https://docs.fastly.com/products/signal-sciences-next-gen-waf#limitations

Configuring A10 Next-Gen WAF

This section describes the steps to import the A10 Next-Gen WAF (NGWAF) module and configure the NGWAF agent on a partition.

The following topics are covered in this section:

Prerequisites

Before configuring the A10 Next-Gen WAF (NGWAF) agent, ensure the following:

1. A10 Thunder device is connected to the internet and can reach the following cloud domains via port 443/TCP:
   - c.signalsciences.net
   - wafconf.signalsciences.net
   - sigsci-agent-wafconf.s3.amaonaws.com
   - sigsci-agent-wafconf-us-west-2.s3.amazonaws.com
   - dl.signalsciences.net
2. The default route is set so that the Thunder device can access the Internet.

   ```
   ACOS(config)# ping 8.8.8.8
   ```

   The NGWAF agent accesses the internet either via management or data interface. However, the NGWAF agent does not support **`ip default-gateway`** command under the management port. So, if you are using the management port, you need to specify explicit static routes to access the DNS server at 8.8.8.8 as well as the Amazon Web Services (AWS) Class A network 52.0.0.0/8 using the **`route`** command as shown below (and in Step-4):

   ```
   ACOS(config)# ip route 8.8.8.8 /32 192.0.2.1
   ```

3. A valid DNS server is configured.

   ```
   ACOS(config)# ip dns primary 8.8.8.8
   ```

4. The IP route is set so that the NGWAF agent can communicate with Fastly's cloud engine hosted on AWS.

   ```
   ACOS(config)# ip route 52.0.0.0 /8  192.0.2.1
   ```

5. A user account is created on the Fastly website.

   To create this account, the Corp Admin needs to add you as a Corp User and then assign you to the same site to which the agent will be registering. For more information, refer to Corp-Management.

6. A valid license is installed through A10's Global License Manager (GLM) in order to use the A10 Next-Gen WAF (NGWAF). For installation steps, see Installing A10 Next-Gen WAF License.

NOTE: The above network settings must be configured on the shared partition even if NGWAF is enabled on the L3V partition.

Installing License

A valid license needs to be installed through A10's Global License Manager (GLM) in order to use A10 Next-Gen WAF (NGWAF). The following NGWAF licenses are available:

- Next-Gen Web App Firewall License
- Next-Gen Web App Firewall Pool License (ACOS 6.0.6 onwards) - Allocates SKU bandwidth to multiple Thunder or vThunder devices. For example, a 20G NG-WAF Pool license can allocate 10G to a 10G vThunder device and allocate the remaining 10G among ten 1G vThunder devices. This license can also be shared between Thunder and vThunder devices. For installation steps, see Capacity Flex Pool Licensing Guide.

NOTE: In case a device has both, the ADC product license and the NGWAF pool license, between the two licensed bandwidths, the lower bandwidth is considered and used to limit traffic.

Installing A10 Next-Gen WAF License

To install the license, you first need to obtain the NGWAF license token from your GLM account or from a sales representative. After obtaining the token, follow the steps mentioned below to activate the license. In this example, `vTh704a5c654` is used as the token.

1. Log in to your ACOS device and enter the configuration mode.
2. Configure your ACOS device with a valid domain name server (DNS).

```
ACOS(config)# ip dns primary 8.8.8.8
```

3. Configure the user management port interface.

```
ACOS(config)# glm use-mgmt-port
```

4. Configure the license by specifying the NGWAF Token.

```
ACOS(config)# glm token vTh704a5c654
```

In case of the NGWAF pool license, you must activate the ACOS UUID on the GLM dashboard and set the required bandwidth. Alternately, if you set the token from the CLI, the license will be activated with the default bandwidth only, based on the available resources in the pool.

5. Enable the connection to GLM.

```
ACOS(config)# glm enable-requests
```

6. Send the license request to the GLM.

```
ACOS(config)# glm send license-request
NGWAF License successfully updated, please log out and log back in to
access license features
```

7. Save the configuration by executing the **`write mem`** command.
8. Verify if the license is installed successfully, by entering the following command:

```
ACOS(config)# show license-info
Host ID          : 82BAB415734A4017FE11A781C0203F8DF3C0B618
Serial No        : N/A
USB ID           : Not Available
Billing Serials  : vTh17bad151f0000, vTh704a5c6540000
Token            : Not Available
Product          : ADC
Platform         : vThunder
Burst            : Disabled
Version          : N/A Green
GLM Ping Interval In Hours : 24
-----------------------------------------------------------------
Enabled Licenses Expiry Date (UTC) Notes
-----------------------------------------------------------------
SLB              None
CGN              None
GSLB             None
```

```
RC                  None
DAF                 None
WAF                 None
AAM                 None
FP                  None
QOSMOS              N/A     Requires an additional QOSMOS license.
WEBROOT_TI          N/A     Requires an additional Webroot Threat Intel
license.
IPSEC_VPN           N/A     Requires an additional IPsec VPN license.
A10_TI              N/A     Requires an additional A10 Threat Intel license.
SECURE_GAMING       N/A     Requires an additional Secure gaming license.
PRIVILEGE_SHELL_ROOT N/A    Requires an additional Privilege shell root
license.
NGWAF               01-September-2023
```

Installing A10 Next-Gen WAF Pool License

The A10 Next-Gen WAF Pool License is a Flex Pool license. It creates a pool of bandwidth that can be applied to multiple Thunder and/or vThunder devices. The bandwidth can be assigned to a device through the GLM activation page by using the device's UUID. The device then receives the license when it connects to the GLM.

To install the license with the default bandwidth using the ACOS CLI, see Installing A10 Next-Gen WAF License. For details on activating, updating, revoking, or unrevoking the A10 Next-Gen WAF Pool license, see Capacity Flex Pool Licensing Guide.

To verify if the NGWAF pool license is installed successfully from the CLI, enter the `show license-info` command. It displays the configured NGWAF BW.

```
ACOS(config)# show license-info
Host ID          : 82BAB111111A4017FE11A781C0203F8DF3C0B723
Serial No        : N/A
USB ID           : Not Available
Billing Serials  : vThxxxxxxxxxxx
Token            : Not Available
Product          : CFW
Platform         : vThunder
Burst            : Disabled
Version          : Thunder Unlimited
```

```
GLM Ping Interval In Hours : 24
------------------------------------------------------------
Enabled Licenses   Expiry Date (UTC)   Notes
------------------------------------------------------------
SLB                None
CGN                None
GSLB               None
RC                 None
WEBROOT_TI         N/A    Requires an additional Webroot Threat Intel
license.
IPSEC_VPN          N/A    Requires an additional IPsec VPN license.
......
NGWAF              30-September-2025
8 cores allowed    None
1Mbps BW (Egress)  None
50000 NGWAF Mbps BW (Egress) 30-September-2025
60000 MB memory allowed   None
```

Enforcing NGWAF Pool License Bandwidth

The NGWAF pool license is introduced from ACOS version 6.0.6 onwards. This section describes the NGWAF pool license bandwidth enforcement, the way the license display varies on ACOS versions prior to 6.0.6, the license application with product license, and the way it is displayed on the Fastly dashboard.

- ACOS versions prior to 6.0.6 + new NGWAF pool license = no change of display in Fastly dashboard.
- ACOS version 6.0.6 + old NGWAF license + product license bandwidth = Fastly dashboard displays the product license bandwidth.
- ACOS version 6.0.6 + old NGWAF license + no product license bandwidth = Fastly dashboard displays the bandwidth as 1 Mbps.
- ACOS version 6.0.6 + new NGWAF pool license = Fastly dashboard displays the NGWAF pool license bandwidth.
- ACOS version 6.0.6 + new NGWAF license + product license bandwidth = The lower bandwidth from the two licenses is considered. If the NGWAF license is not present, then the product license value is considered.
- ACOS version 6.0.6 + NGWAF is running, if the license token or bandwidth changes,

NGWAF restarts within ten minutes to update the license token and bandwidth. The new license token and bandwidth is updated to the Fastly dashboard after NGWAF receives a new request.

Displaying Next-Gen WAF Pool License on Fastly Dashboard

The Fastly dashboard displays the NG-WAF pool license's unique token ID, SKU bandwidth and the number of activations on those many numbers of ADCs with their associated ADC bandwidth. The reporting on Fastly dashboard is done in the following format in the **Server** field in **Agents** page:

<NGWAF licence token> : <UUID> : <Applied TH/vTH NG-WAF BW>

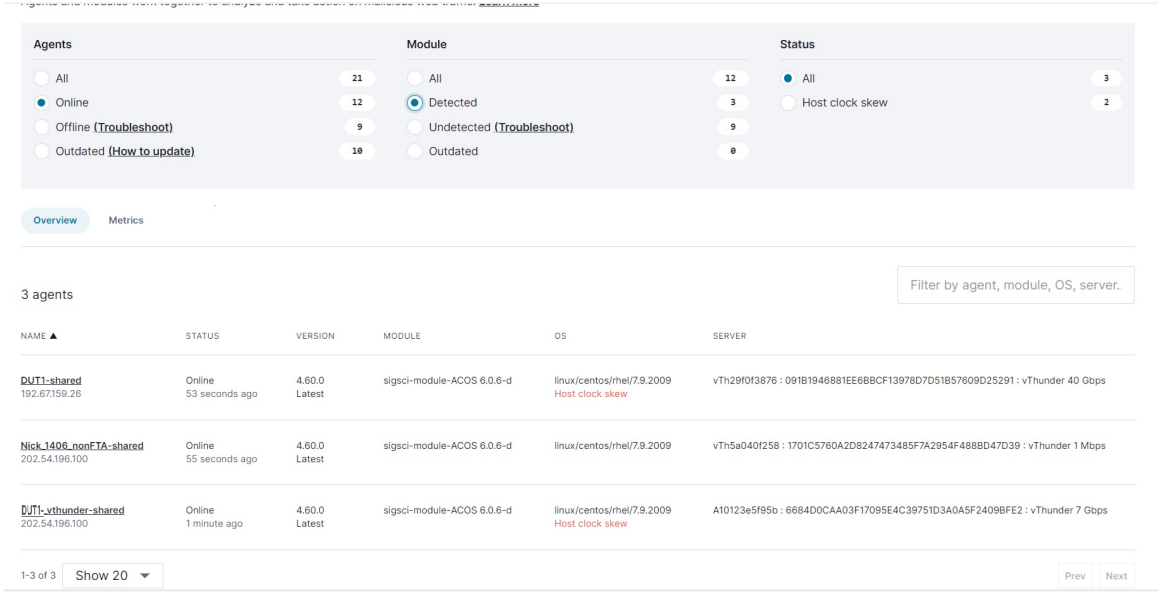
In this format, Applied TH/vTH NG-WAF BW is the bandwidth limit on the ADC. It represents the NG-WAF bandwidth specified at the time of license activation. If NG-WAF bandwidth is not specified at the time of activation, the original ADC product bandwidth is displayed.

While NGWAF is running, if the license token or bandwidth changes, NGWAF restarts within ten minutes to update the license token and bandwidth. The new license token and bandwidth is updated to the Fastly dashboard after NGWAF receives a new request.

Example

The following example indicates the Fastly dashboard view of the device in the format: <NGWAF License token> : <UUID> : <Applied vTH NG-WAF BW>

Figure 2 : Fastly Dashboard Agents Tab



Importing A10 Next-Gen WAF Module

After installing a valid A10 Next-Gen WAF (NGWAF) license, import the NGWAF module (binary file) on ACOS by executing the following command in the shared partition:

```
ACOS(config)# import ng-waf-module use-mgmt-port
https://dl.signalsciences.net/sigsci-agent/sigsci-agent_latest.tar.gz
```

When prompted for **username** and **password**, press enter (password is not required).

NOTE: You are recommended to use the management port for importing the NGWAF module.

For environments where all communication is routed through an explicit proxy, see Importing NG-WAF Files with Explicit Proxy Settings.

Importing NG-WAF Files with Explicit Proxy Settings

In environments with restricted direct internet access and where all communication is routed through an explicit proxy, you can configure the **proxy** parameter of the **import ng-waf-module** and **import ng-waf-custom-page** commands to download NGWAF files. This parameter allows you to specify explicit HTTP proxy settings in the *host:port* format, supporting the usage of IPv4 address, IPv6 address, or hostname as the *host*.

Key Considerations

- An IPv6 address must be encased in a bracket, for example, *[a:b::1]:8080*.
- Only HTTP or HTTPS URLs can be specified for file downloads.
- The **use-mgmt-port** and **overwrite** parameters can also be specified along with the **proxy** parameter.

CLI Configuration

- To import the latest NGWAF agent using proxy settings:

```
ACOS(config)# import ng-waf-module proxy <host:port> <remote_file_path>
```

Example command demonstrating IPv6 address usage:

```
ACOS(config)# import ng-waf-module proxy [2001:db8::1]:3128
https://dl.signalsciences.net/sigsci-agent/sigsci-agent_latest.tar.gz
```

Example demonstrating NGWAF agent upgrade:

```
ACOS(config)# import ng-waf-module overwrite proxy 10.12.10.157:3128
use-mgmt-port https://dl.signalsciences.net/sigsci-agent/sigsci-agent_
latest.tar.gz
```

- To import a custom HTML page for NGWAF using proxy settings:

```
ACOS(config)# import ng-waf-custom-page <html_page> proxy <host:port>
<remote_file>
```

Example command demonstrating hostname usage:

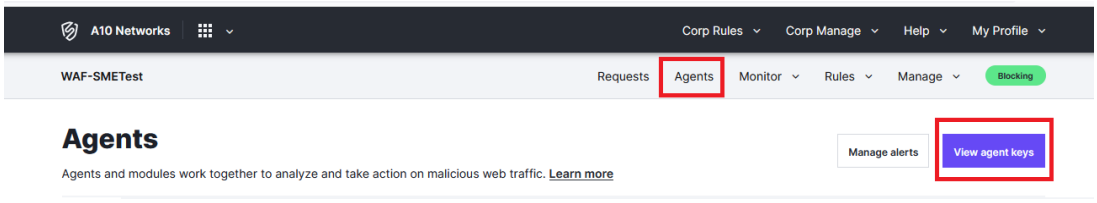
```
ACOS(config)# import ng-waf-custom-page mypage  proxy  myproxy.com:443
 https://www.google.com
```

Configuring A10 Next-Gen WAF Agent

After importing the A10 Next-Gen WAF (NGWAF) module, follow the steps given below to configure the NGWAF agent on a partition:

1. Obtain the **agent key ID** and **secret access key** from the Fastly website. https://dashboard.signalsciences.net/login?next=/

Figure 3 : Fastly Dashboard Agent Key



After logging in, navigate to the **Agents** tab and click **View agent keys** to obtain the keys.

2. On ACOS, configure the NGWAF agent by specifying the **agent key ID** and **secret access key** in the following manner:

```
ACOS(config)# license-manager ng-waf-module access-key-id xxxxxxxx
secret-access-key xxxxxxxxxxx
```

NOTE: This is a per-partition command. The agent can be configured for a maximum of 8 partitions.

3. Enable NGWAF on the virtual port using the following commands:

```
ACOS(config)# slb virtual-server vip 192.168.15.10
ACOS(config-slb vserver))#  port 80 http
ACOS(config-slb vserver-vport)#  service group sg
ACOS(config-slb vserver-vport)#  ng-waf
ACOS(config-slb vserver-vport)#  exit
ACOS(config-slb vserver))#  port 443 https
ACOS(config-slb vserver-vport)#  service group sg
ACOS(config-slb vserver-vport)#  ng-waf
```

NOTE: To disable NGWAF, use the **no ng-waf** command. However, please note that the agent remains active until the **no ng-waf** command is applied to all the virtual ports within the partition.

4. To verify if the above command executed successfully, check the output of the following **show** command:

```
ACOS(config)# show process system  detail | inc a10ngwaf
/a10/bin/a10ngwaf --config /a10 running   32562  0.0    0.5    730232
 84404
```

5. To check the status of the NGWAF agent, execute the following **show** command in the shared partition:

```
ACOS(config)# show ng-waf status
Agent Version: 4.22.0

NGWAF Status:       <partition shared>
```

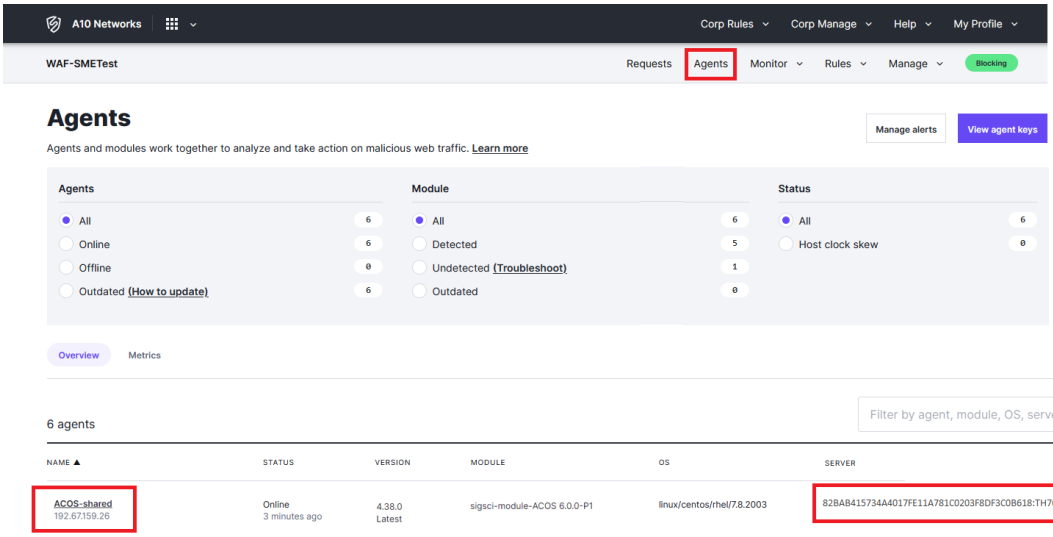
```
Current status:     RUNNING
Agent name:         ACOS_Device-shared
Access key ID:      e9779xxxxxxxxx
Secret access key:  9VF9Uxxxxxxxxx

NGWAF Status:       <partition p2>
Current status:     DOWN
Agent name:         ACOS_Device-p2
Access key ID:      11111xxxxxxxxx
Secret access key:  2xxxxxxxxxx
```

NOTE: The output of the show command depends upon where the command is executed. If it is executed on the shared partition, the status of the NGWAF agents on all partitions is displayed. If it is executed on the L3V partition, only the status of the NGWAF agent on that partition is displayed.

6. You can also check the NGWAF agent on the Fastly Dashboard by navigating to the **Agents** tab as shown below,

Figure 4 : Fastly Dashboard Agents Tab



The **SERVER** field displays the information of the ACOS device (the agent is connected to). The information is displayed in the format **DeviceID:PlatformDetails**, where, **Device ID** is the Universally Unique Identifier (UUID), and **PlatformDetails** refer to the A10 platform information. In case of

Thunder platforms, this field displays the model number (for example, TH5840, TH7655S), whereas, in case of vThunder, the bandwidth license information is displayed.

- Example for Thunder device: **82BAB415734A4017FE11A781C0203F8DF3C0B618:TH7655S**. Here, **82BAB415734A4017FE11A781C0203F8DF3C0B618** is the UUID and **TH7655S** implies that the agent is connected to the Thunder 7655S platform.
- Example for vThunder device: **61CAB615834C6518FE12A892B0405D7EF5C0A737:vThunder 40000 Mbps**. Here, **61CAB615834C6518FE12A892B0405D7EF5C0A737** is the UUID and **vThunder 40000 Mbps** implies that the agent is connected to a vThunder running at 40000 Mbps throughput.

For the complete list of platforms supported by NGWAF, see Platforms Supported.

Points to be considered while configuring NGWAF

- The agent's name is the hostname of your Thunder device + the partition name.
- The Fastly Dashboard will display two agents, and two separate hostnames if you configure the same agent keys for two partitions.
- The **Module** and **Server** fields are empty at the beginning. You need to send a non-existing URI to the virtual port to trigger a 404 error so that the agent sends an update to the Fastly cloud and these fields get updated.
- Agent versions are based on the **Module** version installed. So, a Thunder device with multiple partitions or agents will have agents of the same version.
- Starting January 31, 2023, only NGWAF agent versions 4.16.0 and above will be supported. Thereafter, every quarter, agents older than two years will be deprecated. These agents will not receive rule updates and blocking decisions.

Configuring A10 Next-Gen WAF on Virtual Chassis System

The procedure to configure A10 Next-Gen WAF (NGWAF) on a Virtual Chassis System (VCS) is same as that mentioned in Configuring A10 Next-Gen WAF Agent. However, on a VCS, configuring NGWAF using the `license manager ng-waf-module`

command is not synced (although importing is synced). So, if you execute the **`import ng-waf-module`** command on the Master, the command is executed automatically on the Blade and the NGWAF module is downloaded. But if you execute the **`license manager ng-waf-module`** command on the Master, the command is not executed automatically on the Blade. Therefore, on a VCS, first, you need to execute the **`device-context`** command to specify the Thunder device and then execute the **`license-manager ng-waf-module`** command to specify the agent-key. For a configuration example, see CLI Configuration.

Configuring NGWAF on Multiple Partitions

On the Fastly Dashboard, a corporation can contain several sites and users. Each site contains different rules and NGWAF agents that are configured using the site's access and secret keys. In the case of a Thunder device having multiple partitions, if you want each partition to belong to a different site, you need to configure NGWAF agents belonging to different sites on each partition.

CLI Configuration

Consider a scenario where you need to configure VCS and NGWAF on two Thunder devices that have two partitions each. The following steps provide the configuration details:

1. Configure and enable VCS and VRRP on the two Thunder devices.
2. Log in to the floating IP (VCS management IP).

   For steps 1 and 2, refer to the *Configuring ACOS Virtual Chassis Systems guide* and *Configuring VRRP-A High Availability guide*.

3. Download the NGWAF binary file on both the devices using the **`import ng-waf-module`** command:

   ```
   ACOS# import ng-waf-module use-mgmt-port
   https://dl.signalsciences.net/sigsci-agent/sigsci-agent_latest.tar.gz
   ```

   For more information, refer to Importing A10 Next-Gen WAF Module.

   NOTE: It may take several minutes to upload the binary file to both the devices.

4. On the first device, set agent keys for two partitions using the **`device-context`**

1 command as shown:

```
ACOS(config)# device-context 1
All the following configuration will go to device 1
ACOS(config)# license-manager ng-waf-module access-key-id wwwwwww
secret-access-key xxxxxxxx
ACOS(config)# active-partition l3v
Current active partition: l3v
ACOS[l3v](config)# license-manager ng-waf-module access-key-id yyyyyyy
secret-access-key zzzzzzzz
```

5. On the second device, set agent keys for two partitions using the **device-context 2** command as shown:

```
ACOS[l3v](config:1)# active-partition shared
Current active partition: shared
ACOS(config:1)# device-context 2
All the following configuration will go to device 2
ACOS(config:2)# license-manager ng-waf-module access-key-id wwwwwww
secret-access-key xxxxxxxx
This operation applied to device 2
ACOS(config:2)# active-partition l3v
Current active partition: l3v
ACOS[l3v](config:2)# license-manager ng-waf-module access-key-id
yyyyyyy secret-access-key zzzzzzzz
This operation applied to device 2
```

Upgrading A10 Next-Gen WAF Module

A10 Next-Gen WAF (NGWAF) is not upgraded automatically. To upgrade it to the latest version, import and overwrite the existing NGWAF module by executing the following command,

```
ACOS(config)# import ng-waf-module overwrite use-mgmt-port
https://dl.signalsciences.net/sigsci-agent/sigsci-agent_latest.tar.gz
```

When prompted for the **username** and **password**, press enter (password not required).

For environments where all communication is routed through an explicit proxy, see Importing NG-WAF Files with Explicit Proxy Settings.

Additionally, all previous NGWAF versions can be downloaded from https://dl.signalsciences.net/?prefix=sigsci-agent/. You can also upgrade or downgrade to a specific version by providing the correct download link while importing. For example, to change NGWAF version to v4.20.0, execute the following command,

```
ACOS(config)# import ng-waf-module overwrite use-mgmt-port
https://dl.signalsciences.net/sigsci-agent/4.20.0/linux/sigsci-agent_
4.20.0.tar.gz
```

NOTE: Starting January 31, 2023, only NGWAF agent versions 4.16.0 and above will be supported. Thereafter, every quarter, agents older than two years will be deprecated. These agents will not receive rule updates and blocking decisions.

Additional A10 Next-Gen WAF Configurations

This section provides some additional configurations to customize and override default A10 Next-Gen WAF (NGWAF) settings in ACOS.

NOTE: Although the NGWAF commands in this section are listed under SLB Common, they can also be configured separately for each partition. Therefore, the shared partition and the L3V partitions can have different NGWAF settings.

The following topics are covered:

Enabling Management Interface Usage

A10 Next-Gen WAF (NGWAF) connects to the Fastly Cloud through the data interface by default. For better traffic management, you can connect to the Fastly Cloud through the management interface.

CLI Configuration

- To enable management interface usage for NGWAF:

```
ACOS (config)# slb common
ACOS (config-common)# ng-waf use-mgmt-port
```

- To disable management interface usage for NGWAF:

```
ACOS (config)# slb common
ACOS (config-common)# no ng-waf use-mgmt-port
```

NOTE: This command is only available on the shared partition but is applied to all partitions.

Configuring Explicit Proxy Server Usage

The default proxy-free connection to the Fastly Cloud is vulnerable to internet-borne threats. For better protection, you can configure an explicit proxy server and connect to the Fastly Cloud through it. The proxy server adds an extra layer of security to the connection, thereby protecting against malicious traffic.

CLI Configuration

The **use-https-proxy** command allows you to connect to the Fastly Cloud through an IPv4 or IPv6 explicit proxy server. The following example demonstrates the steps to connect through an IPv6 explicit proxy server:

1. Set up a proxy server for the required IPv6 address. The IP address *2022:192::33* is used in this example.
2. Configure the **use-https-proxy** command under **slb common** by specifying the IPv6 address:

```
ACOS(config)# slb common
ACOS(config-common)# ng-waf use-https-proxy 2022:192::33 3128
```

3. Enable **ng-waf** on the virtual port:

```
ACOS(config)# slb virtual-server vip 172.16.1.142
ACOS(config-slb vserver))# port 443 https
ACOS(config-slb vserver-vport)# ng-waf
```

4. Verify the proxy server usage by checking the Syslog output using the **show log** command.

   Example output:

```
May 15 2023 19:32:35 Info [WAF]:ngwaf is enabled on virtual server vip
vport 443 via [2022:192::33]:3128 CEF: May 15 19:32:35 Temp-vth700-
ip141  CEF:0|A10|CFW|6.0.1-d|909727124728840195|NGWAF enabling on
vport succeeded|2|cs1=vip cs1Label=Virtual Server Name dpt=80 cs2=
[2022:192::33]:3128 cs2Label=Proxy Server Name
```

The above output indicates that NGWAF is connected to the Fastly Cloud through the explicit proxy *2022:192::33*.

NOTE:

- Ensure that the configured explicit proxy server is reachable.
- NGWAF (if enabled) will restart automatically after configuring or deleting the explicit proxy server.

Enabling Cache Function

A10 Next-Gen WAF (NGWAF) cache function can be configured to reduce NGWAF processing time. When this feature is enabled, the checked and forwarded requests are stored in the cache. Thereafter, similar requests are forwarded directly without checking.

NOTE: Blocked requests are not stored in the cache.

CLI Configuration

- To enable NGWAF cache function, configure the **pre-process-enable** command. You can also specify the expiration time for the cache entries (1 to 480 minutes). The default value is 1 minute.

  In the following example, the cache expiration time is set to 5 minutes:

```
ACOS (config)# slb common
ACOS (config-common)# ng-waf pre-process-enable 5
```

- To view the number of cache entries, use the **show ng-waf status** command:

```
ACOS(config)# show ng-waf status
Agent Version: 4.33.0
NGWAF status:           <partition pt1>
```

```
Current status:          RUNNING
Agent name:              ACOS-vth700-ip141-pt1
Access key ID:           e9779xxxxxxxxxxxxx
Secret access key:       9VF9Uxxxxxxxxxxxxx
Cache entries:           1
Tracked custom signals: 0
```

- To clear all NGWAF cache entries, use the **clear ng-waf cache** command:
  - To clear the cache entries in the current partition:

    ```
    ACOS(config)# clear ng-waf cache all
    ```

  - To clear the cache entries for a specific virtual port:

    ```
    ACOS(config)# clear ng-waf cache vip 80
    ```

  Additionally, all cache entries are cleared automatically in the following scenarios:

  - The partition is disabled or deleted.
  - The virtual port is deleted.
  - NGWAF is disabled.

Configuring Custom Response Code

When a request is blocked by A10 Next-Gen WAF (NGWAF), the Thunder device returns a default response code 410. NGWAF provides you with the flexibility to modify the response code based on your requirements. You need to configure the **attack-resp-code** command to specify an HTTP status code (400 to 599) to be returned when a request is blocked.

Additionally, you can also set a custom response code on the Fastly Dashboard to block and redirect requests. However, this setting overrides the configuration on the Thunder device i.e., if you set a response code on the Fastly Dashboard, the response code configured on the Thunder device is ignored. For more information, see Agent Response Codes .

If no response code is set on the Fastly Dashboard or the Thunder device, the status code 410 is returned by default.

CLI Configuration

- To configure a custom response code on the Thunder device:

```
ACOS (config)# slb common
ACOS (config-common)# ng-waf attack-resp-code 456
```

- To configure a custom response code on the Fastly Dashboard, see Fastly Custom Response Code.

Configuring Custom Response Message

When a request is blocked by A10 Next-Gen WAF (NGWAF), the Thunder device displays the following default message:

```
<html><title>Request Denied!</title><body><center>
<h1>Request Denied!</h1>
<p>If you have any questions, please contact the admin.</p>
</center></body></html>
```

NGWAF provides you with the flexibility to modify this message. You can display a custom message and also add some relevant information such as the attack type, the time, the session ID, and the request URI.

To enable this feature, you need to configure the **attack-resp-message** command that allows you to set a block message directly or an HTML webpage that contains the message to be displayed. The reserved keywords can also be specified in the message to display the additional information.

The following table describes the available keywords:

Keywords	Description
$a10_ngwaf_attack_type$	The reason for blocking the request or the attack type. For example, XSS, SQLI, or custom signals. For all supported values, refer to Table 1. These values are based on the system signals received from Fastly.
$a10_ngwaf_time$	The time when the request is blocked.

Keywords	Description
$a10_ngwaf_session_id$	The session ID.
$a10_ngwaf_uri$	The request URI.

NOTE: The keywords are case-sensitive and only 4 keywords can be used in a message or a custom webpage.

CLI Configuration

- To set a custom message when a request is blocked:

```
ACOS(config)# slb common
ACOS(config-common)# ng-waf attack-resp-message "The request is
denied!\n"
```

- To set a custom message using the reserved keywords:

```
ACOS(config)# slb common
ACOS(config-common)# ng-waf attack-resp-message "The request is
blocked!\n Additional Information\n URI:$a10_ngwaf_uri$, time:$a10_
ngwaf_time$, attack:$a10_ngwaf_attack_type$, session ID:$a10_ngwaf_
session_id$\n"
```

  Example output with keywords replaced:

```
The request is blocked!
Additional Information
URI:/pruducts?category=Gifts'--, time:2022-09-20-T06:49:43Z,
attack:SITE-FLAGGED-IP,SQLI, session ID:632962885477430a344ea0ee
```

- To configure a custom webpage when a request is blocked:

```
ACOS(config)# slb common
ACOS(config-common)# ng-waf attack-resp-message custom-page mypage
```

NOTE: Ensure that you import the HTML webpage to the shared partition or an L3V partition before executing this command.

- To import a custom HTML webpage for NGWAF, use the **import ng-waf-custom-page** command:

```
ACOS(config)# import ng-waf-custom-page mypage
scp://hostname@192.168.1.101/ngwaf/index.html
```

NOTE: The HTML file size should not exceed 100 KB.

For environments where all communication is routed through an explicit proxy, see Importing NG-WAF Files with Explicit Proxy Settings to download the HTML webpage for NGWAF.

- To view the custom HTML webpages, use the **show ng-waf custom-page** command:

```
ACOS(config)# show ng-waf custom-page
FILE NAME                                   FILE SIZE (byte)
-----------------------------------------------------------
mypage                                          536
mypage2                                         800
```

- To delete a custom HTML webpage, use the **delete ng-waf-custom-page** command:

```
ACOS(config)# delete ng-waf-custom-page mypage
```

Table 1 : Various attack types and anomalies (system signals)

Attack and Anomaly Names	Description
Attacks	
USERAGENT	Attack Tooling - Indicates automated software usage to identify security vulnerabilities
AWS-SSRF	Server Side Request Forgery - This attack is used to obtain Amazon Web Services (AWS) keys and gain access to S3 data.
BACKDOOR	Backdoor Signal
CMDEXE	Command Execution - This attack attempts to gain control by executing system commands
XSS	Cross Site Scripting attack
TRAVERSAL	Directory Traversal attack
LOG4J-JNDI	Log4J JNDI - This attack attempt to exploit the Log4Shell

Attack and Anomaly Names	Description
	vulnerability (present in Log4J versions earlier than 2.16.0).
SQLI	SQL Injection attack
Anomalies	
ABNORMALPATH	Abnormal Path - Indicates that the original path differs from the normalized path
BHH	Bad Hop Headers - Indicates HTTP smuggling attempts
BLOCKED	Blocked Requests - The requests blocked by Signal Sciences
CODEINJECTION	Code Injection PHP - Indicates that PHP commands are used to gain control or damage a target system
COMPRESSED	Compression Detected - Indicates that the POST request body is compressed and cannot be inspected.
DATACENTER	Datacenter Traffic - Indicates non-organic requests that originate from identified host providers
DOUBLEENCODING	Double Encoding
DUPLICATE-HEADERS	Duplicate Header Names - Request having duplicate header field names
FORCEFULBROWSING	Forceful Browsing - Indicates the failed attempts to access admin pages
GRAPHQL-API	GraphQL API request
GRAPHQL-DUPLICATE-VARIABLES	GraphQL request that contains duplicated variables
GRAPHQL-IDE	GraphQL IDE - Request originating from a GraphQL Interactive Development Environment (IDE)
GRAPHQL-INTROSPECTION	GraphQL Introspection - Indicates the attempt to obtain the schema of a GraphQL API
GRAPHQL-DEPTH	GraphQL Max Depth - Indicates requests that reached or exceeded the maximum depth allowed on the server for

Attack and Anomaly Names	Description
	GraphQL API queries
GRAPHQL-MISSING-REQUIRED-OPERATION-NAME	GraphQL Missing Required Operation Name - Indicates requests that have multiple GraphQL operations but do not define which operation to execute
GRAPHQL-SYNTAX	GraphQL Syntax - Indicates request that contains invalid GraphQL syntax
GRAPHQL-UNDEFINED-VARIABLES	GraphQL Undefined Variable - Indicates requests made to a GraphQL API containing undefined variables that are not expected by the function
HTTP403	HTTP 403 Errors - Request resulting in an HTTP 403 error (Forbidden)
HTTP404	HTTP 404 Errors - Request resulting in an HTTP 404 error (Not Found)
HTTP429	HTTP 429 Errors - Request resulting in an HTTP 429 error (Too many requests)
HTTP4XX	HTTP 4XX Errors - Request resulting in an HTTP 4XX error (other than 403, 404 and 429)
HTTP500	HTTP 500 Errors - Request resulting in an HTTP 500 error (Internal Server error)
HTTP503	HTTP 503 Errors - Request resulting in an HTTP 503 error (Service Unavailable)
HTTP5XX	HTTP 5XX Errors - Request resulting in an HTTP 5XX error (other than 500 and 503)
RESPONSESPLIT	HTTP Response Splitting - Indicates requests where CRLF characters are submitted as input to the application to inject headers into the HTTP response
NOTUTF8	Invalid Encoding - Requests with invalid encoding
JSON-ERROR	JSON Encoding Error - Requests with JSON encoding error
MALFORMED-DATA	Malformed Data - Requests with a malformed POST, PUT or PATCH request body

Attack and Anomaly Names	Description
SANS	Malicious IP Traffic - Requests from IP addresses with malicious activity
SIGSCI-IP	Network Effect - Subsequent requests from IPs that were marked malicious previously
NO-CONTENT-TYPE	Missing **Content-Type** request header - Indicates requests with **Content-Type** header missing
NOUA	No User Agent- Indicates automated and malicious requests using fake or missing User-Agents
NULLBYTE	Null Bytes - Requests with null bytes (malformed request).
PRIVATEFILE	Private Files - Requests with private files that could leak sensitive information
SCANNER	Scanner - Requests that identify popular scanning services and tools
IMPOSTOR	SearchBot Impostor
SITE-FLAGGED-IP	Site Flagged IP - Requests received from an IP that was flagged for exceeding attack thresholds for a specific site
TORNODE	Tor Traffic - Requests with Tor traffic that conceals a user's identity
WEAKTLS	Weak TLS - Requests with weak TLS connections
XML-ERROR	XML Encoding Error - Requests with XML encoding errors

For more information on system signals, see Diving Into System Signals.

Enabling Monitor Mode

A10 Next-Gen WAF (NGWAF) provides a monitor mode that allows you to monitor requests instead of blocking them. When this mode is enabled, all the requests are monitored and allowed to pass. However, the malicious requests, that would otherwise be blocked, are highlighted in the syslogs and debug logs using the **`marked`** keyword.

This mode proves to be useful if you want to analyze and evaluate network traffic.

NOTE:

- The monitor mode does not offer active protection.
- The monitor mode overrides the Fastly rules configured to block and redirect requests, i.e., the Fastly rules are ignored when the monitor mode is enabled on the Thunder device.

Consider the following syslog when NGWAF monitor mode is not enabled. The malicious request is blocked and is highlighted as **`HTTP request is blocked by ng-waf`**.

```
Sep 21 17:32:29 ACOS_Device  CEF:0|A10|SSLI|6.0.0-
d|900719925474099218|HTTP request is blocked by ng-waf|5|src=172.16.1.160
spt=40806 dst=172.16.1.142 dpt=443 cs1=SITE-FLAGGED-IP,SQLI
cs1Label=Attack Type cs2=172.16.1.142 cs2Label=Hostname
cs3=/pruducts?category=Gifts'-- cs3Label=uri
```

However, when NGWAF monitor mode is enabled, the request is not blocked and is highlighted as **`HTTP request is marked by ng-waf`**.

```
Sep 21 17:32:29 ACOS_Device  CEF:0|A10|SSLI|6.0.0-
d|900719925474099218|HTTP request is marked by ng-waf|5|src=172.16.1.160
spt=40806 dst=172.16.1.142 dpt=443 cs1=SITE-FLAGGED-IP,SQLI
cs1Label=Attack Type cs2=172.16.1.142 cs2Label=Hostname
cs3=/pruducts?category=Gifts'-- cs3Label=uri
```

CLI Configuration

- To enable NGWAF monitor mode, configure the **`monitor-mode-enable`** command:

```
ACOS (config)# slb common
ACOS (config-common)# ng-waf monitor-mode-enable
```

- To view the number of **`marked`** requests, use the **`show ng-waf`** command:

```
ACOS(config)# show ng-waf vip 443
Requests
 Received                                4
 Marked                                  1
Attacks
 SQL Injection                           1
```

```
Anomalies
 Site Flagged IP                          1
```

Tracking Custom Signals

Requests that are immediately blocked or allowed by rules (configured on the Fastly dashboard) are not visible in the console. To add visibility to such requests, rules are configured to add custom signals to the requests. The Thunder device can track these custom signals using enhanced show commands. However, these commands can track only the first 64 custom signals. To reserve and track important custom signals, you need to create a class-list and bind it to NGWAF.

CLI Configuration

- To reserve and track certain custom signals, follow the steps:

  1. Create a string type class-list (*signal_list* in this example) and add the signal names to it:

```
ACOS(config)# class-list signal_list string
ACOS(config-class list)# str site.login
ACOS(config-class list)# str site.error
ACOS(config-class list)# str site.logout
```

NOTE: The class-list cannot have more than 64 entries.

  2. Bind the created class-list to NGWAF:

```
ACOS(config)# slb common
ACOS(config-common)# ng-waf custom-signal-clist signal_list
```

- To view the custom signals that are triggered when requests are blocked:

```
ACOS(config-slb vserver-vport)# show ng-waf vip 80
Requests
 Received                  40
 Forwarded                 22
 Blocked                   15
 Error                     3
Attacks
```

```
 Command Execution          8
 Cross Site Scripting       3
 Directory Traversal        8
 SQL Injection              3
Anomalies
 Private Files              1
Custom Signals
 site.login                 1
 site.error                 1
 site.logout                1
```

NOTE: Only the first 64 signal counters are displayed; the later counters (65th and above) are counted as **Unknown**.

- To view the number of custom signals that are being tracked:

```
ACOS(config)# show ng-waf status
Agent Version: 4.22.0

NGWAF Status:             <partition shared>
Current status:           RUNNING
Agent name:               ACOS_Device-shared
Access key ID:            e9779xxxxxxxxx
Secret access key:        9VF9Uxxxxxxxxx
Cache entries:            1
Tracked custom signals:   3
```

- To view all the custom signals that are being tracked:

```
ACOS(config)# show ng-waf custom-signals
Custom Signals
site.login
site.error
site.logout
```

This command displays the signals reserved in class-list as well as the signals triggered by the blocked requests.

- To clear the NGWAF custom signals:

```
ACOS(config)# clear ng-waf custom-signals
```

This command clears all custom signals defined on the Fastly dashboard except the signals that are reserved and tracked using the class-list.

For more information on custom signals, see Exploring Custom Signals.

Switching Agent Modes

The decision to block or log a request depends upon the mode configured for the NGWAF agent on the Fastly Dashboard. The following three modes are available for the agents:

- **Blocking** – The agent checks and identifies whether a request contains an attack and blocks the request if it crosses the threshold of malicious activity. Attacks are blocked by returning a unique HTTP 410 response code.

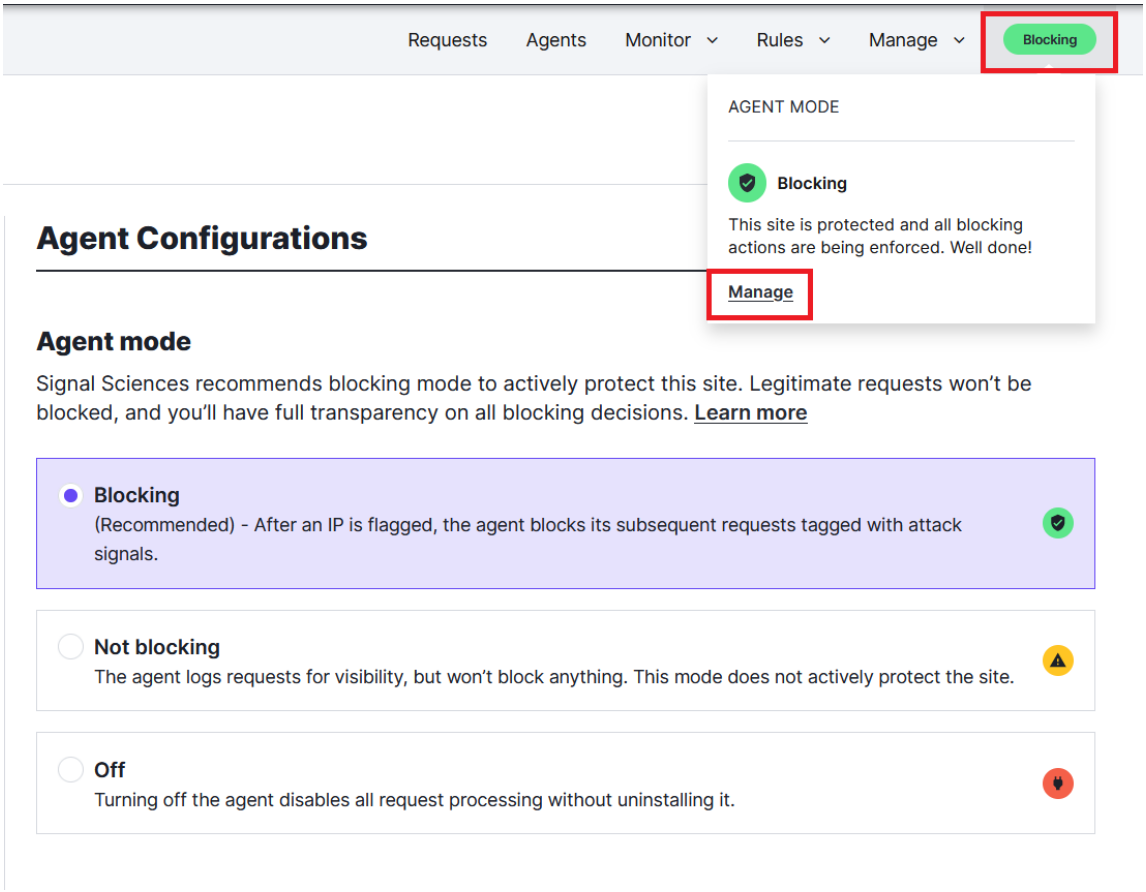
  NOTE: This is the default mode for all agents, and it does not block legitimate requests.

- **Not Blocking** – The agent does not block any request but logs them for visibility. This is a monitor mode that does not offer active protection.
  - **Off** – The agent is turned OFF and request logging is disabled. Additionally, data is not sent to the cloud for analysis.

    NOTE: Enabling this mode does not uninstall the NGWAF agent.

On the Fastly Dashboard, the NGWAF agent mode can be changed by clicking the mode on the top navigation and then clicking **Manage** as shown in Figure 5.

Figure 5 : Agent Mode



These agent modes can also be configured using ACOS command line interface (CLI). The following table provides the details:

Fastly Dashboard	ACOS Command Line Interface
Blocking Mode (Default)	This is the default mode. It is enabled when **`ng-waf`** is applied to a virtual port. However, ensure that **`ng-waf monitor-mode-enable`** is not configured under **`slb common`**.
Non-Blocking Mode (Monitor mode)	This mode is enabled when **`ng-waf`** is applied to the virtual port and **`ng-waf monitor-mode-enable`** is configured under **`slb common`**. For more information, see Enabling Monitor Mode.
Off	**The agent is turned OFF when `ng-waf` is not applied to**

Fastly Dashboard	ACOS Command Line Interface
	any of the virtual ports (on any partition).

Configuring Rules on Fastly Dashboard

The **Rules** tab of the Fastly Dashboard allows you to configure rules to block, forward, and tag requests and exclude system signals for certain conditions. This section describes the various features provided to configure the rules.

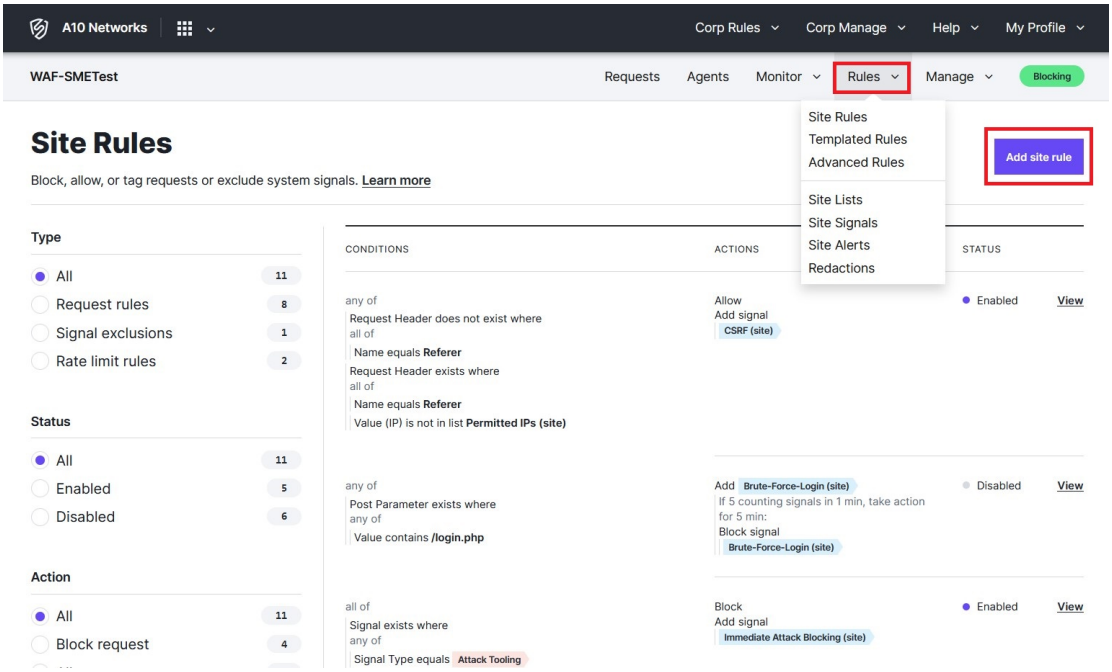
The following topics are covered:

Site Rules and Corp Rules

Rules can be configured at the following levels:

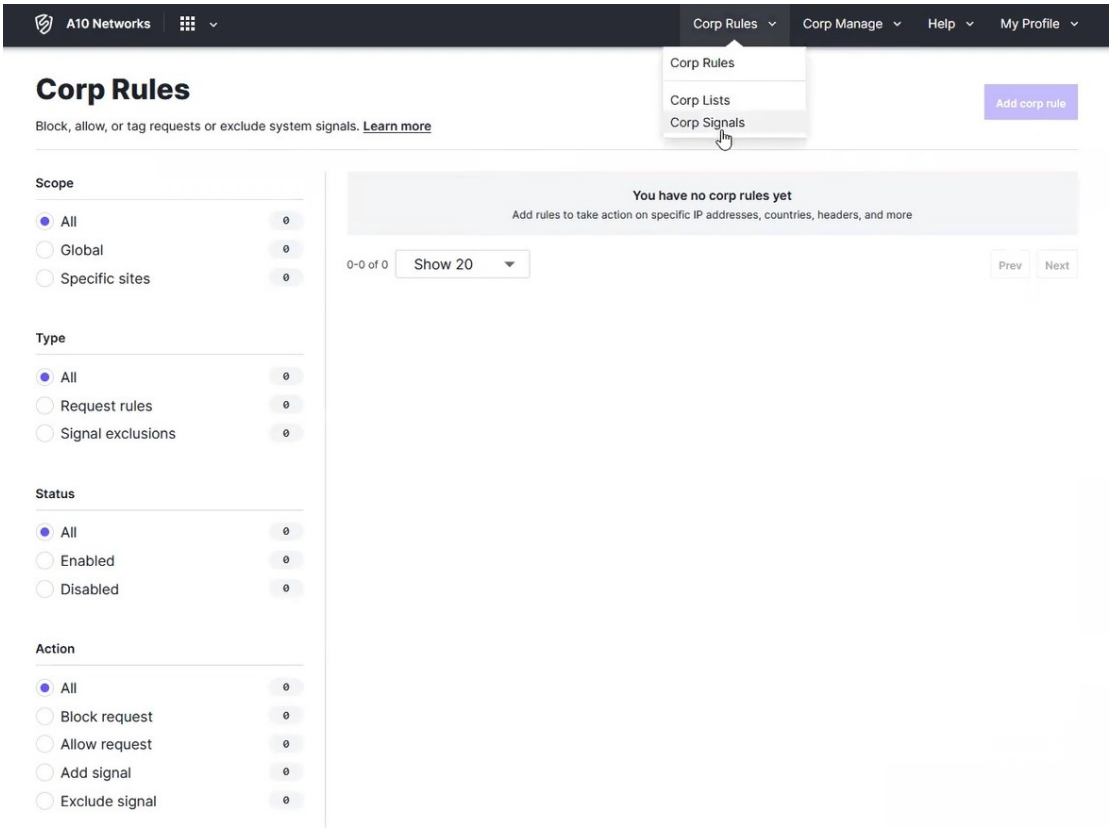
- **Individual sites**: These are known as **site rules** and apply to the agents in that site. They can be managed by navigating to **Rules > Site Rules** tab (as shown in Figure 6).

Figure 6 : Site Rules



- **Corporation as a whole**: These are known as **corp rules** and they apply to all the agents across all sites of the Corporation. They can be managed by navigating to **Corp Rules > Corp Rules** tab (as shown in Figure 7). For more information, refer to Corp Management.

Figure 7 : Corp Rules

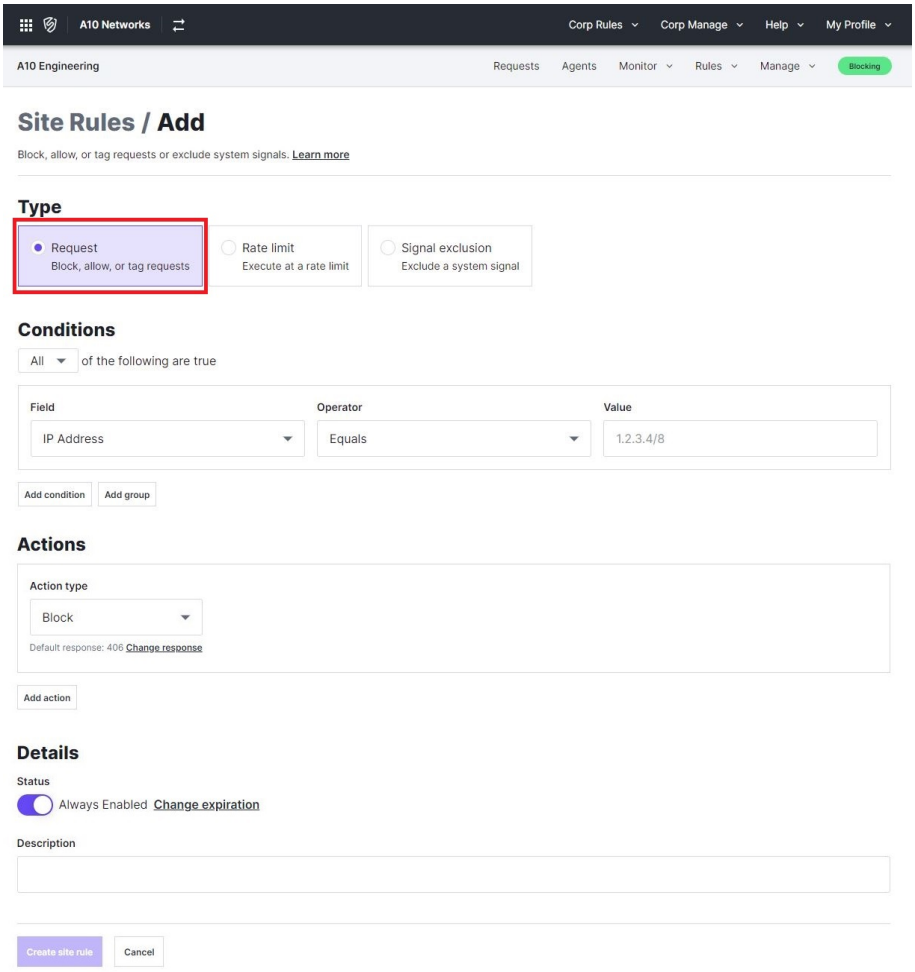


Rule Types

Rules can be created by clicking **Add Site Rule** on the **Site Rules** page (as shown in Figure 6). The following types of rules can be configured and downloaded to the NGWAF agent:

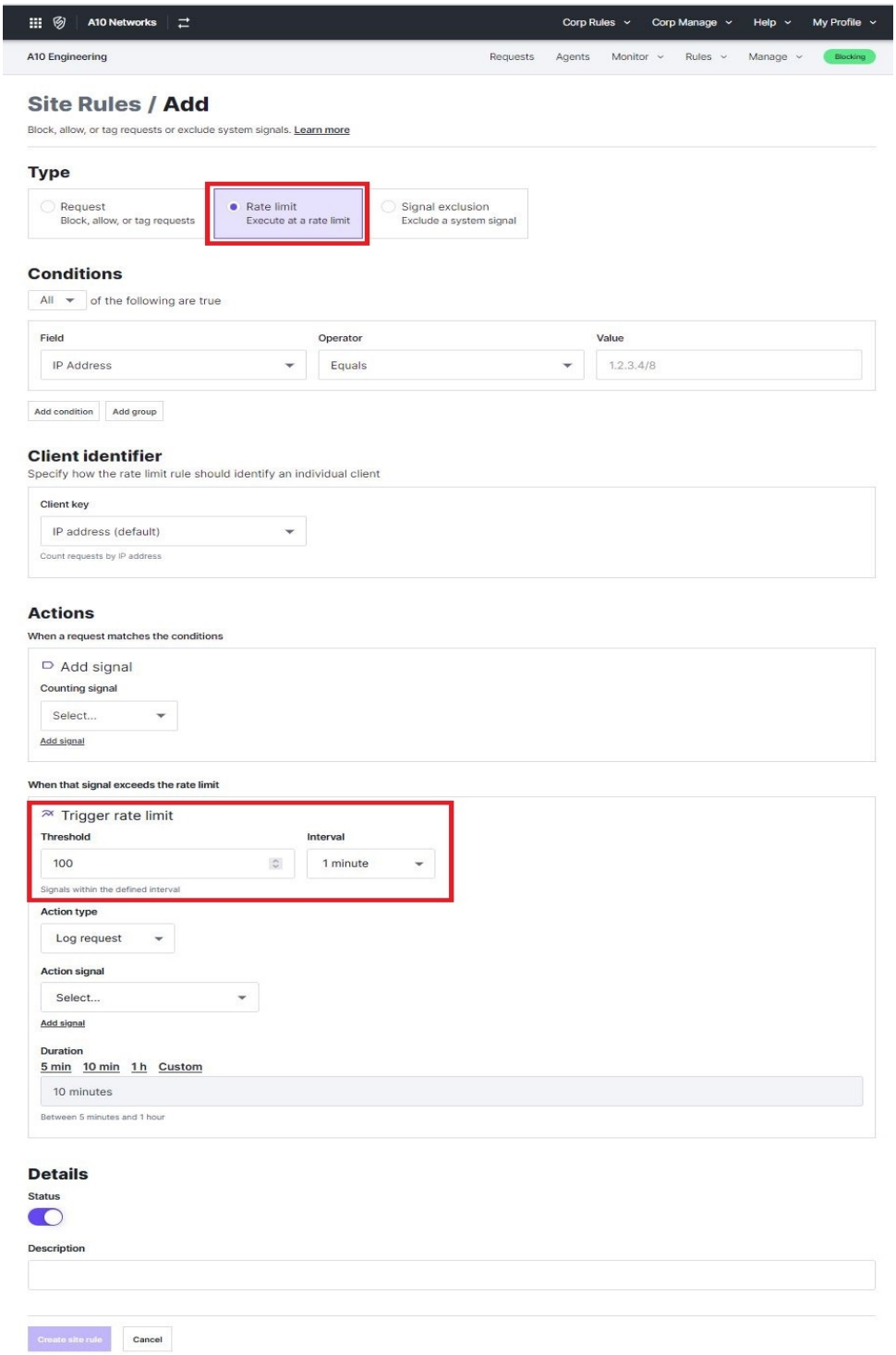
- Request Rules: These rules allow you to define certain conditions and either block, forward, or tag requests indefinitely or for a specific period of time. Requests that cross the threshold are flagged and are either blocked or forwarded. The threshold can only be triggered when the device shares data at the frequency of checks to the Fastly cloud.

Figure 8 : Request Rule



- Rate Limit Rules: This is a premium feature that determines how many requests are forwarded before logging or blocking of subsequent requests when a user-defined threshold is crossed. Since the threshold is triggered directly on the agent, requests are limited to the exact threshold and remain limited until the configured duration is complete.

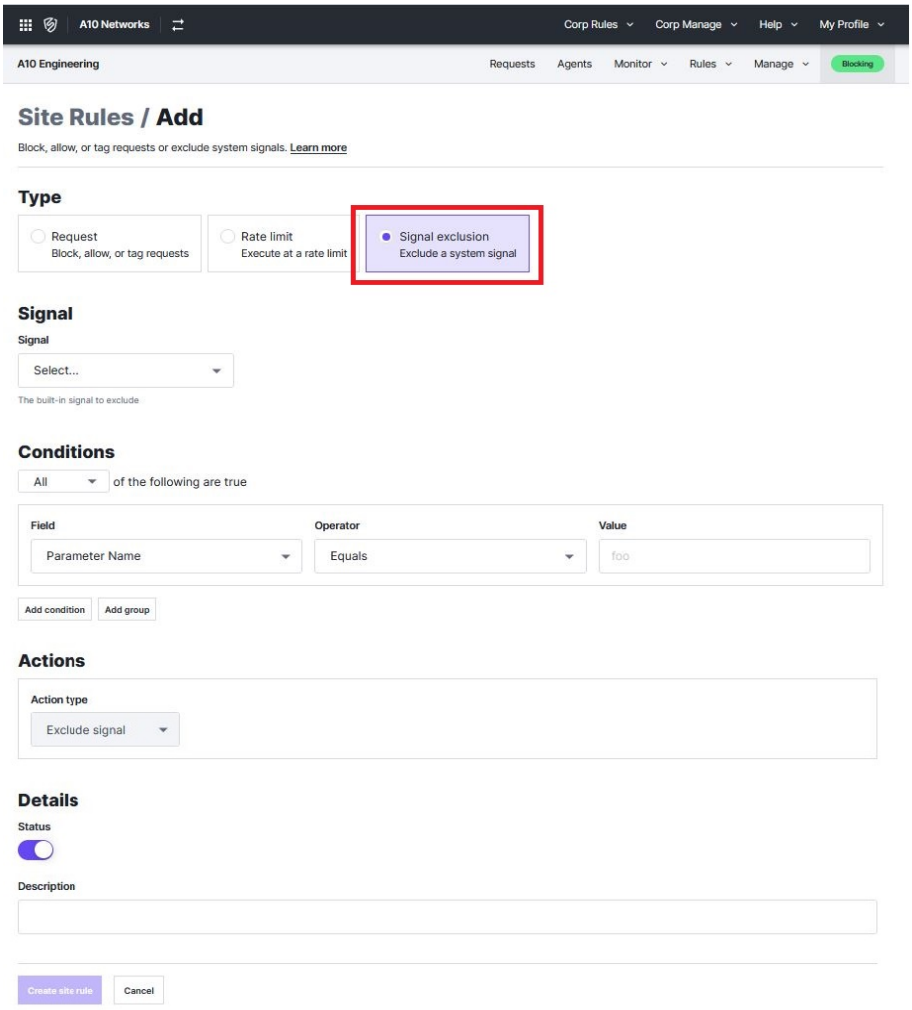
Figure 9 : Rate Limit Rules



Additionally, Templated Rules can also be used to gain visibility into registrations, logins, and virtual patches within your application by configuring simple rules.

- Signal Exclusion Rules: These rules allow you to define certain conditions to exclude a specific system signal.

Figure 10 : Signal Exclusion Rules



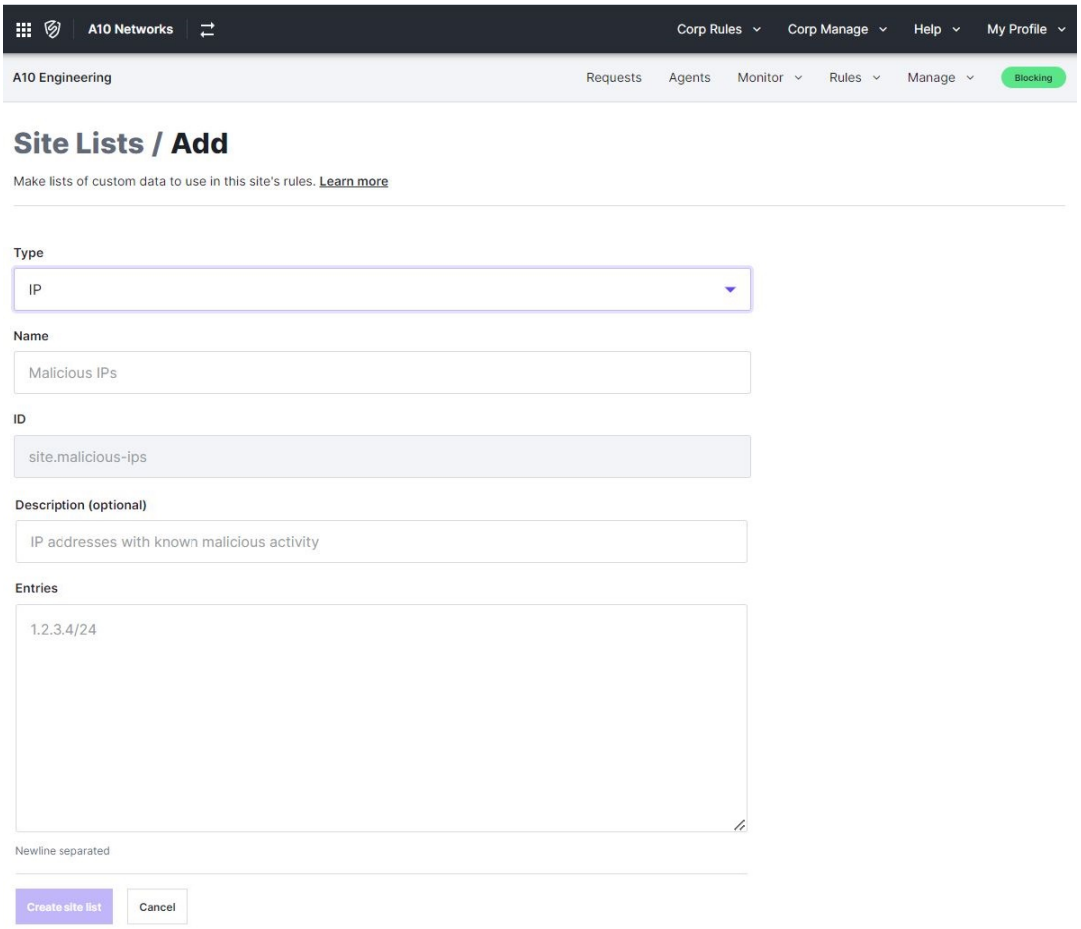
For more information on rules, refer to https://docs.fastly.com/signalsciences/using-signal-sciences/features/rules/

Advanced Features

The **Rules** tab also provides the following advanced features for configuring rules:

- Site Lists: Lists are used to create and maintain sets of data (countries, IP addresses, strings and wildcards) that can be employed while creating rules and can be easily reused across multiple rules. Lists can be created for individual sites (known as Site Lists) as well as for the Corporation as a whole (known as Corp Lists) to be used in multiple sites.

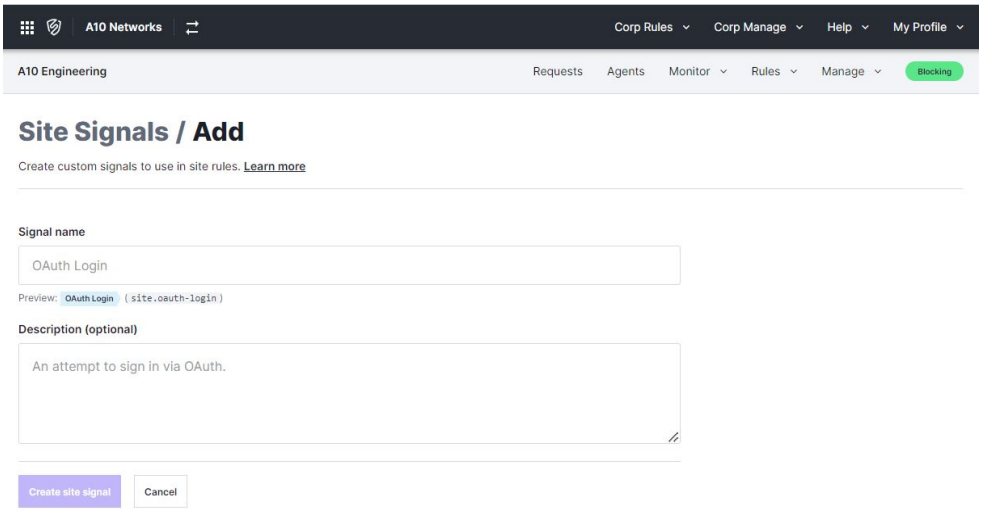
  Figure 11 : Site Lists



- Site Signal: To add visibility to blocked or allowed requests, you can configure a rule to add a signal to the requests. Signals can be created on individual sites

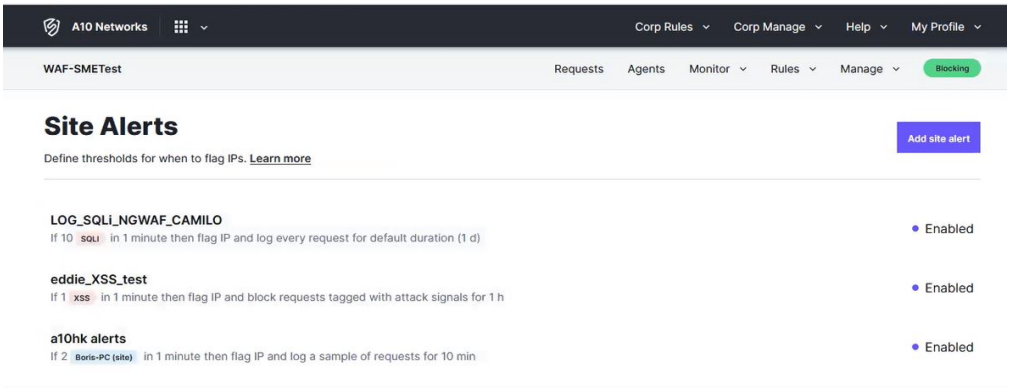
(known as Site Signals) as well as the Corporation as a whole (known as Corp Signals) to be used in multiple sites.

Figure 12 : Site Signals



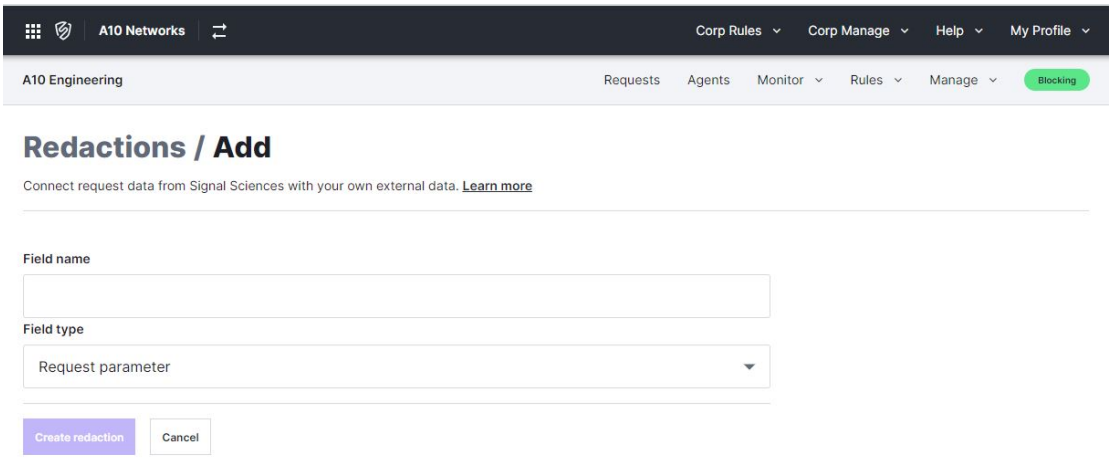
- Site Alert: This feature allows you to define thresholds for flagging an IP address and treating subsequent requests from that IP.

Figure 13 : Site Alert



- Redaction: To maintain data privacy, sensitive data (e.g., sensitive headers, parameters and patterns) is automatically redacted from requests before it reaches the platform backend. The sensitive data includes tokens, credentials, and known patterns such as credit card and social security numbers. In addition to these redactions, you can also specify additional fields to be redacted from requests.

Figure 14 : Redactions



Agent Response Codes

This section provides a brief overview of agent response codes and the usage of custom response codes to redirect requests.

The following topics are covered:

Overview

Agent response codes reflect Next-Gen WAF agent's decisions to forward or block requests to your web application.

When a request is sent, the NG-WAF agent evaluates it against rules and site alerts to determine the appropriate action. Based on this evaluation, the agent assigns a response code indicating whether the request is forwarded or blocked. Agent response codes help troubleshoot issues, monitor security, and track web application performance. For more information, see About Agent Response Codes.

There are two types of agent response codes:

- **System agent response codes**: These codes (value less than or equal to 300) indicate if a request is forwarded or processed incorrectly.

- **Custom agent response codes**: These codes allow you to specify an HTTP status code to be returned when a request is blocked or redirected.

  Supported custom response codes:

  - 301-302 - To indicate redirection of requests.
  - 400-599 - To indicate blocking of requests. By default, all blocked requests receive an agent response code of 406.

Redirecting Requests Using Custom Response Codes

The custom response codes 301 and 302 can be used to redirect requests to the specified location.

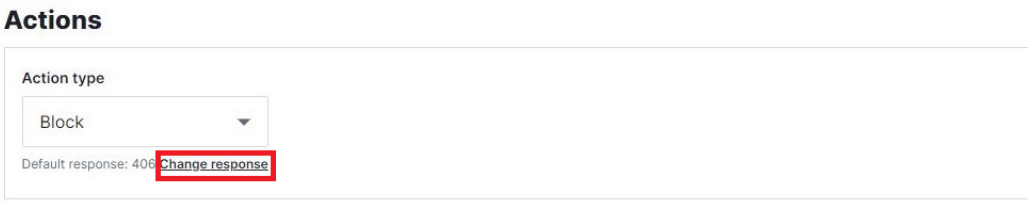
To redirect requests, perform the following steps on the Fastly dashboard:

1. Log in to the Next-Gen WAF control panel.
2. From the **Rules** menu, select **Site Rules**.
3. Click **Add site rule**.
4. Enter the appropriate details in the **Type**, **Conditions**, and **Client Identifier** sections.
5. Perform the following steps in the **Actions** section:

   a. From the **Action type** menu, select the action **Block** to block the requests when the specified condition matches.

   The **Change response** link appears below the drop-down menu, as shown in Figure 15.

   Figure 15 : Setting Actions in Fastly Rules



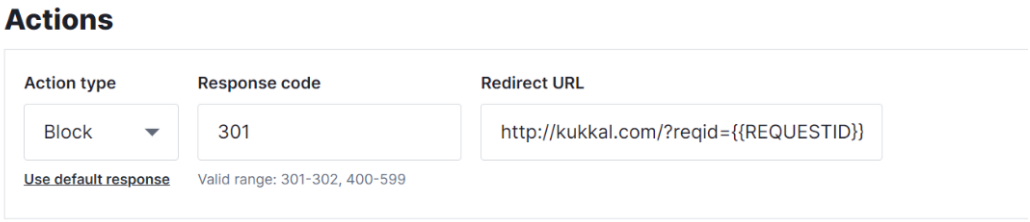
   b. Click **Change response**.

   The **Response code** field is displayed.

c. In the **Response code** field, enter the custom response code 301 or 302.

The **Redirect URL** field is displayed as shown in Figure 16.

Figure 16 : Setting Redirection URL in Fastly Rules



d. In the **Redirect URL** field, enter the absolute or relative URL of the redirect location.

e. Optionally, you can also include the `{{REQUESTID}}` placeholder along with the redirect URL to pass the request ID to the new location.

For example: `https://www.example.com/?reqid={{REQUESTID}}`

When the redirection occurs, the `{{REQUESTID}}` placeholder is replaced with the actual ID of the request. This allows easy tracking of the request in Syslog as well as on the Fastly dashboard.

Viewing Redirected Requests in ACOS

In ACOS, the agent's response is parsed and the redirect response is sent to the client. Additionally, you can use show commands and log messages to view the redirected requests.

- To view the number of redirected requests for a specific virtual port, use the **`show ng-waf`** command:

```
ACOS(config)# show ng-waf vip 80
Requests
 Received                        10
 Redirected                      1

Anomalies
 Blocked Requests                 1
```

```
Custom Signals
 SUSPECTED-BOT                    1
 site.daniel-test                 1
```

- To view the logs, use the **`show log`** command:

  Syslog format

  ```
  Dec 22 11:53:45 Daniel-vth700-ip141 a10lb: [WAF]<3> request redirected
  by ng-waf for SIP 172.16.1.160 SPort 49180 DIP 172.16.1.142 DPort 443
  UserName NULL Attack SUSPECTED-BOT,site.daniel-test-redirected,BLOCKED
  Hostname 172.16.1.142 URI /daniel-test-redirected SessionID
  65850849b24e2c130fb5b0e9 RedirectedURL
  https://myserver.com/?reqid=65850849b24e2c130fb5b0e9
  ```

  In this log, the string **`https://myserver.com`** indicates the redirected URL and **`65850849b24e2c130fb5b0e9`** indicates the request ID.

  CEF Format

  ```
  Dec 22 11:53:45 Daniel-vth700-ip141  CEF:0|A10|CFW|6.0.4-
  d|909727124728840213|HTTP request is redirected by ng-
  waf|5|src=172.16.1.160 spt=49180 dst=172.16.1.142 dpt=443
  cs1=SUSPECTED-BOT,site.daniel-test-redirected,BLOCKED cs1Label=Attack
  Type cs2=172.16.1.142 cs2Label=Hostname cs3=/daniel-test-redirected
  cs3Label=URI cs4=65850849b24e2c130fb5b0e9 cs4Label=Session ID
  cs5=https://myserver.com/?reqid=65850849b24e2c130fb5b0e9
  cs5Label=Redirected URL
  ```

  In this log, the **`cs5`** string **`https://myserver.com/`** indicates the redirected URL and **`65850849b24e2c130fb5b0e9`** indicates the request ID.

Limitations

- The maximum length for a redirect URL cannot exceed 16384 characters.
- Request redirection does not function when **`ng-waf monitor-mode-enable`** is configured on the ACOS device because, in the monitor mode, Fastly site rules for blocking and redirecting requests are ignored.

Logs and Statistics

This section describes the logs and statistics for A10 Next-Gen WAF (NGWAF).

The following topics are covered in this section:

NGWAF Logging

Configuration changes, events, or issues are logged and can be viewed using the **`show log`** command or on the remote log server. However, if a request is blocked by NGWAF, logs are only sent to the remote syslog server. The local log only record the start and stop activities of the NGWAF Agent process.

The log messages provide the following information about an event:

- Date and time.
- Device on which the event occurred.
- Application or service involved.
- The type of event that occurred. For example, **`An HTTP request was blocked by ng-waf`**.
- Source IP address (along with port number) of the request, which is extracted from the IP header.

  When NGWAF is positioned behind a proxy, load balancer, or a NAT device, the Source IP address may correspond to that of the intermediate device rather than the original client's IP address.
- Real Source IP address (along with port number) of the request.

This is the actual IP address of the client, which is extracted from the X-Forwarded-For (XFF) header.

NOTE: Since the XFF header is not formally standardized, some deviations in the IP address format may occur. Additionally, NGWAF does not validate the IP address format presented in this header; instead, it parses the content and logs it.

- Destination IP address and destination port.
- The attack type, for example, `SUSPECTED-BOT`, `SQLI`.
- Action taken when the event occurs, for example, `BLOCKED`.
- Hostname of the destination server.
- The URI or endpoint targeted in the request.
- Session ID associated.

  The session ID provides valuable contextual information about each session, which helps troubleshoot issues and improves security analysis.

ACOS NGWAF log messages are sent to the logging servers in Syslog or CEF format (based on the configuration).

Syslog Format

Syslog is a standard for message logging used for system monitoring, troubleshooting, and security analysis. It consists of a header and a message body. The header includes information such as the severity level, timestamp, and hostname of the device generating the log. The message body contains the actual log message, including events, errors, warnings, or other information.

Sample log messages

- **Log with Real Source IP address and Session ID**

```
2024-02-20T18:01:25-05:00 1406_nonFTA a10logd: [WAF]<3> request blocked
by ng-waf for SIP 11.0.0.1 SPort 46892 RealSource 10.23.14.13 DIP
11.0.0.33 DPort 443 Attack SUSPECTED-BOT,site.tanmayee-test-002,BLOCKED
Hostname 11.0.0.33 URI /tanmayee-test-002 SessionID
65d4e8f51dfb008fc1a2893a
```

  In this log, **10.23.14.13** indicates the real source IP address and **65d4e9f11dfb008fc1a2893b** indicates the session ID.

- **Log indicating Request Redirection**

```
Dec 22 11:53:45 Daniel-vth700-ip141 a10lb: [WAF]<3> request redirected
by ng-waf for SIP 172.16.1.160 SPort 49180 DIP 172.16.1.142 DPort 443
UserName NULL Attack SUSPECTED-BOT,site.daniel-test-redirected,BLOCKED
Hostname 172.16.1.142 URI /daniel-test-redirected SessionID
65850849b24e2c130fb5b0e9 RedirectedURL
https://myserver.com/?reqid=65850849b24e2c130fb5b0e9
```

  In this log, **https://myserver.com** indicates the redirected URL.

Common Event Format (CEF)

CEF is a standard log format used for transmitting security event information from devices for monitoring and analysis. It consists of a set of standard and optional fields. The standard fields include event name, severity level, source and destination IP addresses, timestamp, and device hostname. The custom fields are included to provide additional information about the event.

Sample log messages

- **Log indicating an SQLI attack**

```
Sep 24 08:25:49 ACOS_Device CEF:0|A10|CFW|5.3.0-
d|900719925474099208|HTTP request is blocked by ng-waf|5|src=172.16.1.70
spt=60972 dst=192.168.91.105 dpt=80 cs1=SQLI cs1Label=Attack Type
cs2=192.168.91.105 cs2Label=Hostname cs3=/pruducts?category=Gifts'--
cs3Label=uri
```

- **Log indicating a Command Execution and Directory Traversal attack**

```
Sep 24 08:25:54 ACOS_Device CEF:0|A10|CFW|5.3.0-
d|900719925474099208|HTTP request is blocked by ng-waf|5|src=172.16.1.70
spt=60973 dst=192.168.91.105 dpt=80 cs1=CMDEXE,TRAVERSAL cs1Label=Attack
Type cs2=192.168.91.105 cs2Label=Hostname cs3=/ cs3Label=uri
```

- **Log with Real Source IP address and Session ID**

```
2024-02-20T18:05:37-05:00 1406_nonFTA  CEF:0|A10|SSLI|6.0.4-
d|909727124728840200|HTTP request is blocked by ng-waf|5|src=11.0.0.1
spt=40514 cs1=10.23.14.13 cs1Label=Real Source dst=11.0.0.33 dpt=443
cs2=SUSPECTED-BOT,site.tanmayee-test-002,BLOCKED cs2Label=Attack Type
cs3=11.0.0.33 cs3Label=Hostname cs4=/tanmayee-test-002 cs4Label=URI
cs5=65d4e9f11dfb008fc1a2893b cs5Label=Session ID
```

The **cs1** string (**10.23.14.13**) indicates the real source IP address and the **cs5** string (**65d4e9f11dfb008fc1a2893b**) indicates the session ID.

- **Log indicating Request Redirection**

```
Dec 22 11:53:45 Daniel-vth700-ip141  CEF:0|A10|CFW|6.0.4-
d|909727124728840213|HTTP request is redirected by ng-
waf|5|src=172.16.1.160 spt=49180 dst=172.16.1.142 dpt=443 cs1=SUSPECTED-
BOT,site.daniel-test-redirected,BLOCKED cs1Label=Attack Type
cs2=172.16.1.142 cs2Label=Hostname cs3=/daniel-test-redirected
cs3Label=URI cs4=65850849b24e2c130fb5b0e9 cs4Label=Session ID
cs5=https://myserver.com/?reqid=65850849b24e2c130fb5b0e9
cs5Label=Redirected URL
```

The **cs5** string (**https://myserver.com/**) indicates the redirected URL and **65850849b24e2c130fb5b0e9** indicates the request ID.

A10 Next-Gen WAF Statistics

To view the A10 Next-Gen WAF (NGWAF) statistics for a specified virtual port, use the **show ng-waf** command as shown below,

```
ACOS(config)# show ng-waf vip 80
Requests
  Received              40
  Redirected             2
  Forwarded             22
  Blocked               15
  Error                  3
Attacks
  Command Execution      8
  Cross Site Scripting   3
  Directory Traversal    8
  SQL Injection          3
Anomalies
  Private Files          1
Custom Signals
  site.tp                2
```

The following table describes all possible fields of the command output.

Table 2 : **show ng-waf** field descriptions

Field	*Description*
Requests - Counters for the HTTP requests processed	
Received	The total number of HTTP requests received. It is the sum of the Forwarded, Blocked, Bypassed and Erroneous requests.
Redirected	**The number of HTTP requests blocked and redirected.**
Forwarded	The number of HTTP requests forwarded.
Blocked	The number of HTTP requests blocked.
Bypassed	The number of HTTP requests bypassed. A request is bypassed under the following conditions: the request matches the NGWAF cache, the request encounters

Field	*Description*
	resource allocation problems, or when the NGWAF service is restarted.
Error	The number of errors occurred while processing the requests.
Marked	The number of HTTP requests highlighted as **`marked`** when NGWAF monitor mode is enabled.
Attacks - Counters for the attacks	
Attack Tooling	The number of Attack Tooling attacks detected.
AWS SSRF	The number of Server Side Request Forgery (SSRF) attacks detected to obtain Amazon Web Services (AWS) keys and gain access.
Backdoor	The number of Backdoor Signal attacks detected.
Command Execution	The number of Command Execution attacks detected.
Cross Site Scripting	The number of Cross Site Scripting attacks detected.
Directory Traversal	The number of Directory Traversal attacks detected.
GraphQL Max Depth	The number of requests that reach or exceed the maximum depth allowed on the server for GraphQL API queries.
Log4J JNDI	The number or Log4J JNDI attacks that attempt to exploit the Log4Shell vulnerability (present in Log4J versions earlier than 2.16.0).
SQL Injection	The number of SQL Injection attacks detected.
Anomalies - Counter for the anomalous requests	
Abnormal Path	The number or times the original path differs from the normalized path.
Bad Hop Headers	The number of HTTP smuggling attempts.
Blocked Requests	The number of requests blocked by Fastly.
Code	The number of requests where PHP code commands are used to

Field	*Description*
Injection PHP	gain control or damage a target system.
Compression Detected	The number of requests where the POST request body is compressed and cannot be inspected.
Datacenter Traffic	The number of non-organic requests that originate from identified host providers.
Double Encoding	The number of double encoded requests.
Duplicate Header Names	The number of requests having duplicate header field names.
Forceful Browsing	The number of failed attempts to access admin pages.
GraphQL Duplicate Variables	The number of requests that contain duplicated variables.
GraphQL IDE	The number of requests originating from a GraphQL Interactive Development Environment (IDE).
GraphQL Introspection	The number of attempts to obtain the schema of a GraphQL API. The schema can be used to identify which resources are available, thereby informing subsequent attacks.
GraphQL Max Depth	The number of requests that reached or exceeded the maximum depth allowed on the server for GraphQL API queries.
GraphQL Missing Required Operation Name	The number of requests that have multiple GraphQL operations but do not define which operation to execute.
GraphQL Undefined Variable	The number of requests made to a GraphQL API containing undefined variables that are not expected by the function.
HTTP 403 Errors	The number of requests resulting in an HTTP 403 (Forbidden error) status code.
HTTP 404	The number of requests resulting in an HTTP 404 (Not Found

Field	*Description*
Errors	error) status code.
HTTP 429 Errors	The number of requests resulting in an HTTP 429 (Too Many Requests) status code.
HTTP 4XX Errors	The number of requests resulting in HTTP 4xx errors (other than 403, 404 and 429) commonly known as client request errors.
HTTP 500 Errors	The number of requests resulting in an HTTP 500 (Internal Server error) status code.
HTTP 503 Errors	The number of requests resulting in an HTTP 503 (Service Unavailable) status code.
HTTP 5XX Errors	The number of requests resulting in HTTP 5xx errors (other than 500 and 503) commonly known as server related issues.
HTTP Response Splitting	The number of requests where CRLF characters are submitted as input to the application to inject headers into the HTTP response.
Invalid Encoding	The number of requests with Invalid Encoding.
JSON Encoding Error	The number of requests with JSON encoding error.
Malformed Data in the response body	The number of requests with a malformed POST, PUT or PATCH request body.
Malicious IP Traffic	The number of requests from IP addresses with malicious activity.
Missing Content-Type request header	The number of POST, PUT or PATCH request types that do not have Content-Type request header.
Network Effect	The number of subsequent requests from IPs that were marked malicious previously.
No User Agent	The number of automated and malicious requests using fake or missing User-Agents.

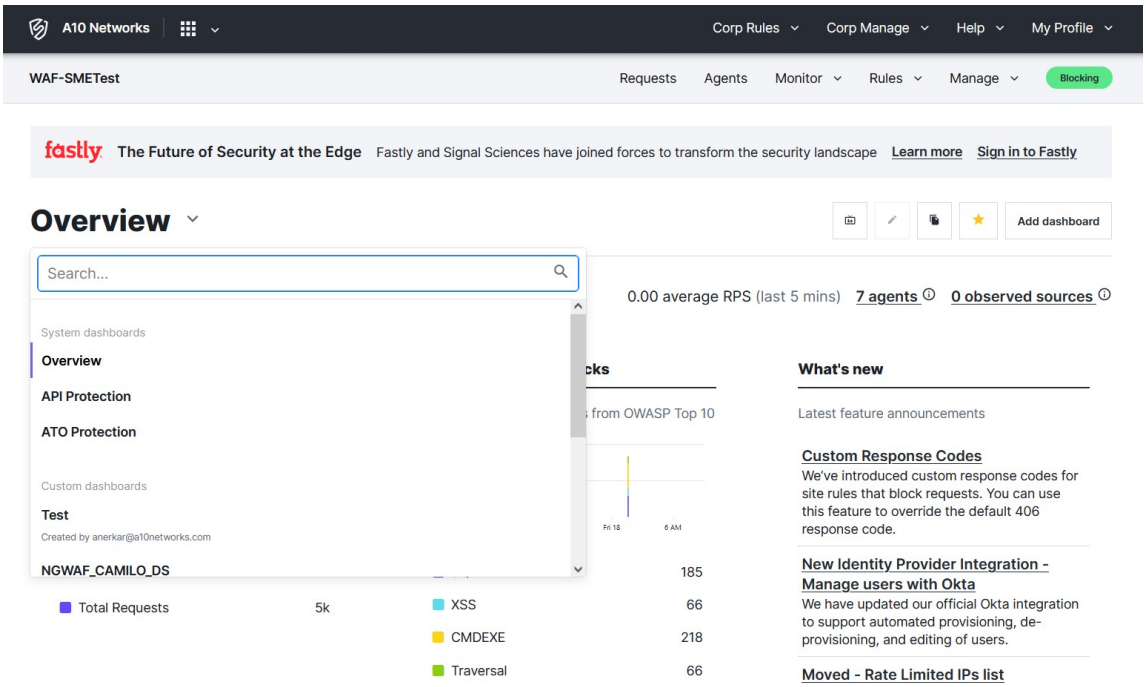
Field	*Description*
Null Byte	The number of requests with Null Bytes (malformed request).
Private Files	The number of requests with private files that could leak sensitive information.
Scanner	The number of requests that identify popular scanning services and tools.
SearchBot Impostor	The number of requests with Search bot impostors.
Site Flagged IP	The number of requests received from an IP that was flagged for exceeding attack thresholds for a specific site.
Tor Traffic	The number of requests with Tor traffic that conceal a user's identity.
Weak TLS	The number of requests with weak TLS connections.
XML Encoding Error	The number of requests with XML encoding errors.
Other	The number of requests other than those mentioned above.

For further details, refer to https://docs.fastly.com/signalsciences/faq/system-tags/.

Fastly Dashboard

The Fastly Dashboard is a web interface that can be used to monitor anomalous web traffic and check the actions performed by the NGWAF for certain requests. The dashboard can be accessed by logging into https://dashboard.signalsciences.net/. You can navigate to the dashboard by clicking the drop-down selector and selecting the preferred dashboard you'd like to view.

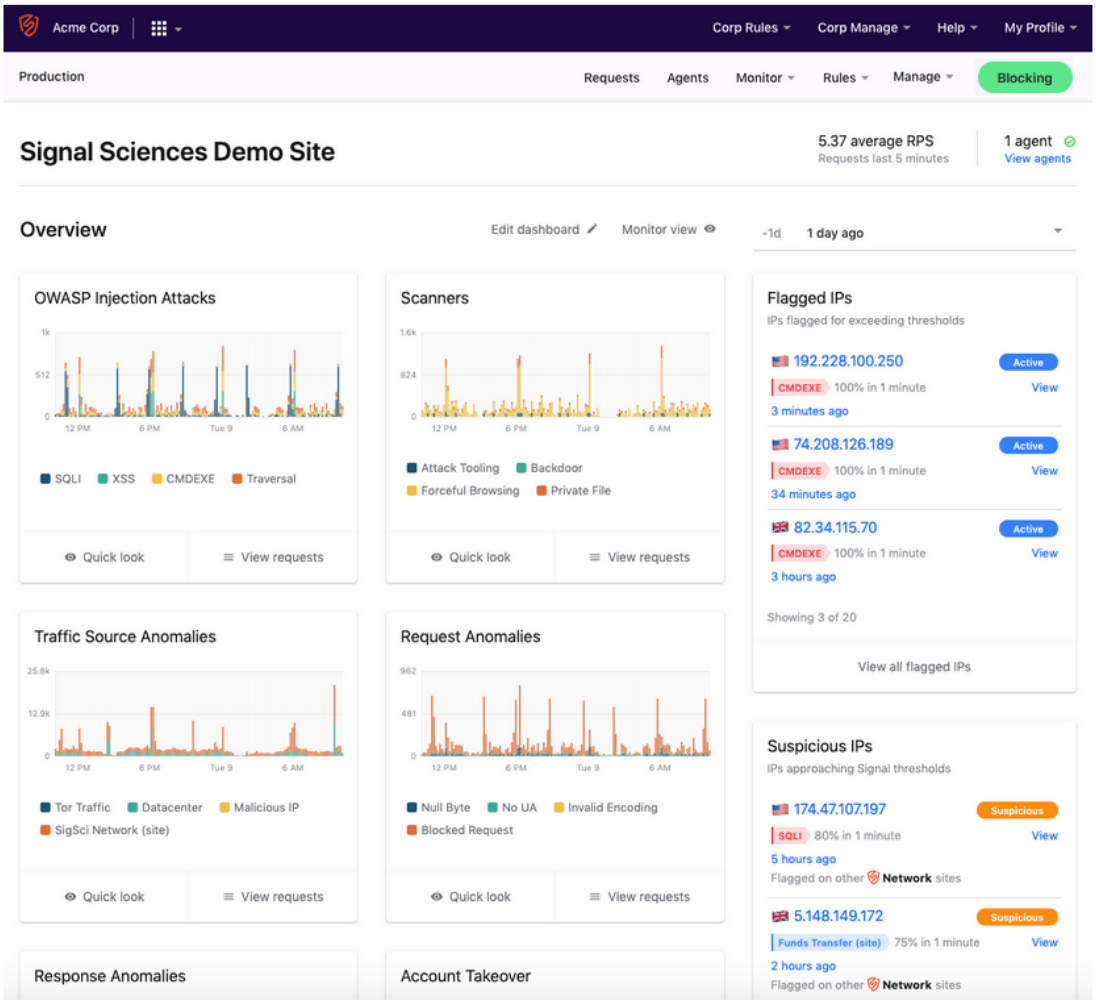
Figure 17 : Fastly Dashboard



The popular dashboards are:

- **Overview** - This dashboard provides a quick glance or overview about the attacks or irregularities that are being managed by the NGWAF. These include graphs for OWASP Injection Attacks and different types of Anomalies. You can get granular details from these graphs, by clicking the requests or highlighting the time-period you are interested in.

Figure 18 : Fastly Dashboard - Overview

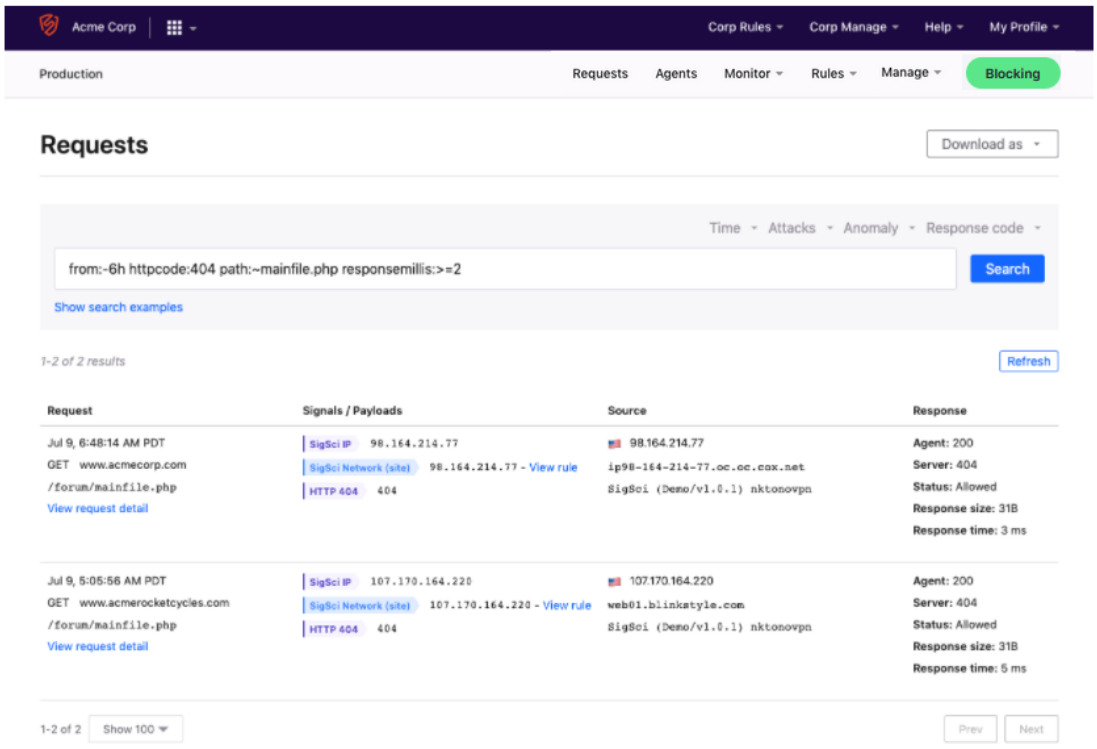


- **Signals Dashboard** - To open this view, navigate to **Monitor > Signals Dashboard**. This view contains breakdowns of the default and custom site signals being tracked in your deployment. This dashboard provides a more detailed view of the activity happening in your environment.

Figure 19 : Signals Dashboard

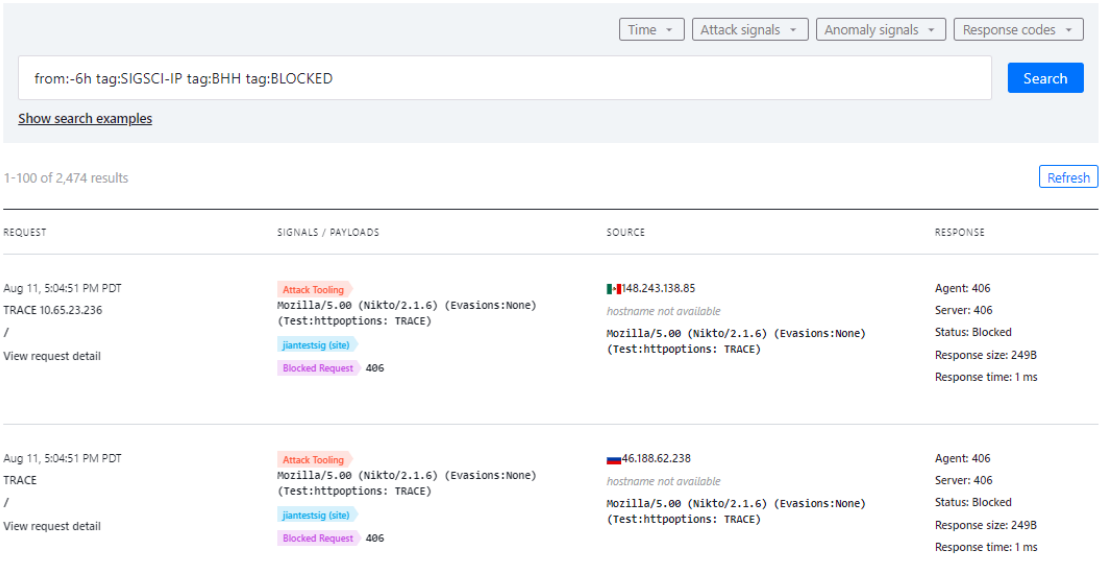
- **Requests** - This view is a very powerful interface for finding information on the different types of requests that are coming through. The requests that are sent to the Cloud are going to be either threats or anomalous tagged requests.

Figure 20 : Fastly Dashboard – Requests View



Additionally, logs can also be checked on the Fastly Dashboard as shown below,

Figure 21 : Fastly Dashboard – Logs view

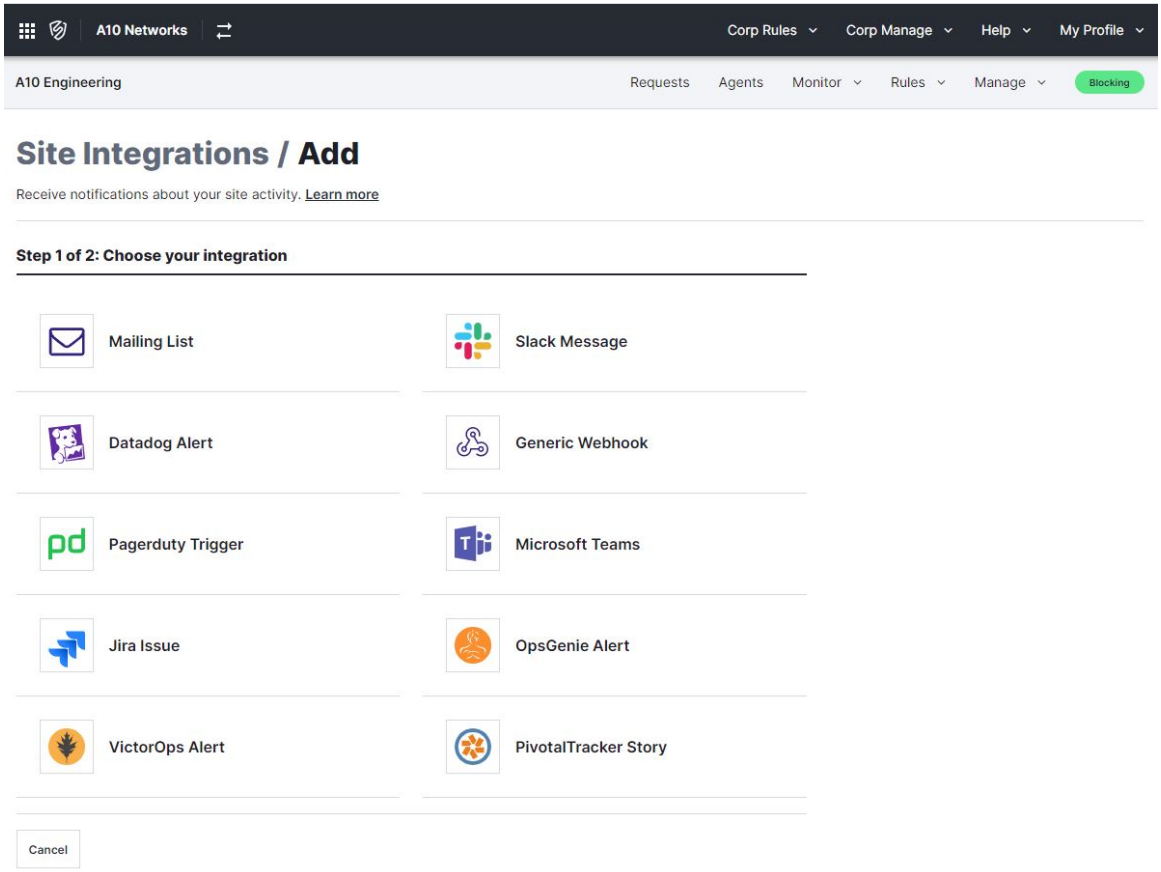


For more details, refer to https://dashboard.signalsciences.net/.

Site Integration

Third party products and services (as shown in Figure 22) can be integrated with Fastly to receive notifications about site activity. For more information, refer to https://docs.fastly.com/signalsciences/integrations/.

Figure 22 : Site Integration



Contact Us