ENLIGHT WAF FEATURES

Contents

Intuitive Dashboard

Dashboard will show statistics and useful information about the health of your eNlight WAF cluster.

You will also find:

- The number of active Listeners (Number of IP addresses/ports are listening).
- The number of active Applications (Web Applications available through WAF).

System Status

CPU, RAM, SWAP and partitions usage evolution are displayed on this page. Evolution of the number of processes running on each WAF Node.

Network Status

Bytes received over time

It is the number of bytes per second received by WAF on all network interfaces.

Bytes sent over time

It is the number of bytes per second sent by WAF on all network interfaces.

Number of entries in the firewall state table

This is a very important stat: It shows the number of active entries in the pf firewall state table.

Incoming packet dropped

It is the total number of incoming packets which were dropped on all network interfaces.

Number of errors while receiving

It is the total number of errors while receiving.

Number of errors while sending

It is the total number of errors while sending.

Reporting

Map

This page shows the number of hits by country for the last 10 minutes.

Access

This page displays several charts which allows you to see general information about your applications.

Charts available on this page:

- Number of hits by status code (line chart)
- HTTP code (pie chart)
- HTTP methods (pie chart)
- Browsers (pie chart)
- Operating systems (pie chart)
- Traffic per URL (data table)
- Average bytes received per request (line chart)
- Average bytes sent per request (line chart)
- Average time elapsed per request (line chart)

Security

This page displays several charts which allows you to see security information about your applications.

Charts available on this page:

- Number of hits by status code (line chart)
- Average score by status code (pie chart)
- Distribution of blocked requests (radar chart)
- Number of blocked requests (bar chart)
- OWASP Top 10 requests (bar chart)
- Reputation tags (pie chart)
- IP list reputation (bar chart)

Packet Filter

This page displays several charts which allows you to see incoming and outgoing traffic captured by Packet Filter.

Charts splitted in two sections (incoming and outgoing) available on this page:

- Number of hits (line chart)
- Source IP (pie chart)
- Destination IP (radar chart)

- Firewall actions (pie chart)
- Requests per destination port (bar chart)

WAF Management

Nodes

From the "WAF Management/Nodes" view you can manage the global WAF Cluster:

1. Active- Active WAF Cluster
2. Active-Passive WAF Cluster

Users

WAF Management UI users group membership

1. Administrator
2. Application Manager
3. Security Manager
4. System Manager

The following table describes permissions associated to groups:

Fonction	Administrator	Application Manager	Security Manager	System Manager
Balancer config	x			x
Proxy Balancer config	x	x		
Listener management	x			x
Network services management	x			x
Portal template management	x	x		
Statistics & diagnostic	x	x	x	x
Users management	x			
ACL management	x		x	
TLS Profiles	x	x		
PKI Management	x			x
Repository management	x	x		
WAF management (policy)	x		x	
WAF management (rules)	x		x	
Cluster, Nodes and updates	x			x
Application management	x	x	x	
URL rewriting	x	x		
Worker profiles	x	x		

Figure 1 User Permissions

Reputation

IP reputation databases for showing reputation of client IPs.

How does reputation work?

A WAF system check in WAF logs if any source IP address is known to have a bad reputation. If a match is found, WAF will modify the log entries to add the corresponding tags:

Can I block traffic based on IP source reputation?

WAF looks for the source IP address and, if found and the tag needs to be blocked, WAF will return a "403 FORBIDDEN" HTTP response.

Log Viewer

There are different types of logs inside WAF:

- Applications WAF logs.
- Internal WAF diagnostic logs (updated every minute).
- Packet filter logs.

Several actions and options are available from the search toolbar:

Search: Shows the query builder form.

Real Time: Log display is updated continuously (every second).

Configuration: Shows the configuration form to customize your display.

Date: Date and time filter.

Query: The active query filter.

Export to csv file: You can export the query result into a CSV file.

Reset: Delete the query filter.

Execute: Execute the search query.

Save as dataset: After a query has been executed, you can save the resulting logs into a "Dataset". This dataset can be used later to build a mathematical model and find anomalies based on SVM algorithms (Support Vector Machine).

Query builder

The query builder let you build a query with AND / OR logic and save it for further reuse.

Detail

When you click on a log line, you've got all the details on the log entry:

Available actions when you "right click" on a line of log

Add IP to blacklist

This will automatically add the source IP address to pf network Firewall blacklist.

Add whitelist/blacklist

You will be able to review the rule automatically created for you and decide whether you want to accept or block the corresponding HTTP request. Once done, the rule will be automatically added to the blacklist / whitelist associated with the Web application. The corresponding rules are available in WAF Ruleset.

WL BL Builder

With this, you can build whitelist/ blacklist rules with all the data of the selected log line.

Load balancer

WAF provides high-performance load-balancing features on incoming request.

You can use this feature to accept incoming connection on any TCP port and load-balance it to another server:

Another WAF node inside the cluster for incoming web traffic.

Another server for another protocol.

Balancing mode: For now, only round robin is supported in WAF.

The HTTP Mode is needed if you need the original client's IP logged by WAF. This mode adds a `X-Forwarded-For' header with the Client's IP address.

You can fine-tune some settings to improve performance / to adapt timeout to backend:

Timeout connect: Maximum timeout for client to connect to WAF, defaults to 500ms.

Timeout server: Maximum timeout for WAF to connect to the backend, defaults to 2s.

Timeout client: Maximum timeout for a client to acknowledge or send data, defaults to 5s.

Max connections: Maximum simultaneous connections, defaults to 10,000.

URL Rewriting

When WAF receives an HTTP request, you can modify the requested URL before processing it.

You can create rules that apply on all applications or only to the specified ones.

If you have multiple rules, WAF will apply them in the following way:

1. Rules that apply to all applications are activated first, in the httpd "" context
2. Rules that apply to a specific application are then activated, in a httpd "" context

Rewriting policy

You can combine multiple conditions and action in a rule policy.

Proxy Balancer

Sometimes a web application runs on multiple web servers, to improve performance and availability.

WAF can load-balance incoming HTTP requests to multiple web backend:

Proxy load-balancer

Unlike load-balancer, Proxy balancer works in layer 7 on HTTP protocol.

You can combine both incoming TCP load-balancer and Proxy balancer to have powerful multi-layered load-balancing.

Topology

Trusted IP for which X-Forwarded-For is acceptable: IP/Block IP/FQDN

These directives list the IP address from which WAF will trust the X-FORWARDED-FOR header.

ACLs

WAF can perform access control before accepting to serve a request.

Here you can create ACLs to allow or deny requests based on several criteria:

- HTTP method,
- Client IP address,
- Apache Expression,
- Apache environment variable

Web Firewall

WAF Ruleset

WAF gives you the possibility to edit security rules. When you create whitelist / blacklist from logs, related rules appear here.

- Manually edit the rules: Writing custom security rules.
- Import OWASP rules: Download security rules package defined by OWASP covering a large panel of known vulnerabilities.

It's possible to fine select those imported OWASP rules. Clicking on the "OWASP_CRS" rule dataset gives you the ability to toggle specific rules. It is also possible to edit those rules.

WAF gives you the ability to use Virtual Patching, which generates a list of security rules from a vulnerability scan report.

There are also some whitelist rulesets for known CMS provided out of the box by WAF.

Anti Session Hijacking

If no session provided, server gives a new session and WAF remembers it. Session sent by client is checked by WAF for valid session id if valid then forwards to webserver. If client provides fake or stolen session, WAF blocks the request.

CSRF Token

Inject a CSRF token in every form, which will be checked upon each POST request. If the CSRF does not match with the one generated by WAF, the anomaly score is increased by the specified amount in the WAF Policy.

User-Agent checking

Capture the User-Agent and check whether it is an:

- Unknown UA
- UA Anonymous
- UA Bot
- UA Browser
- UA Cloud
- UA Console
- UA Crawler

- UA Emailclient
- UA Emailharvester
- UA Mobile
- UA Script

Content-type whitelist

Check whether the Content-Type is one of the following (specified in the WAF Policy):

- application/x-www-form-urlencoded
- multipart/form-data
- text/xml
- application/xml
- application/x-amf
- application/json
- application/json-rpc

Protocol whitelist

Check whether the Content-Type is one of the following (specified in the WAF Policy):

- HTTP/1.0
- HTTP/1.1
- HTTP/2

File extension whitelist

For a file upload, check whether the file extension is included in the list specified in the WAF Policy

Headers whitelist

Check whether sent Header field is not one of the following (specified in the WAF Policy):

- Proxy-Connection
- Lock-Token
- Content-Range
- Translate
- via
- if

WAF Policy

This menu lists the custom policies in effect.

It allows you to create new one on the fly, and you can fine tune them.

Main settings

Enable body inspection

Toggle the body parsing (CPU intensive).

Enable content injection

Toggle the content injection (CPU intensive)

Disable backend compression

Disable compression.

Validate UTF8 Encoding

Toggle the parsing and validation of the UTF8 encoding compliance.

XML Inspection

Validate the XML structure correctness.

Block invalid body

Returns 400 (Invalid request) status code is body is detected as invalid.

Connections engine

Toggle WAF detection mode

- **Detection only:** Process WAF rules without disruptive actions
- **On:** Process WAF rules with disruptive actions
- **Off:** Do not process WAF rules

Audit Engine

Log level, enable/disable or only log if relevant.

Three options are available:

- **On:** log all transactions,
- **Off:** do not log any transaction,
- **RelevantOnly:** only the log transactions that have triggered a warning or an error, or have a status code that is considered to be

Logging Mode

Select the log file, audit_log, error_log or both

Enable Mod Defender: Toggle the activation of Mod Defender.

This directive is required if you want to use:

- **Generic SQL detection**
- **Generic XSS detection**

Enable Generic SQL Detection: Toggle the activation of LibinjectionSQL.

Enable Generic XSS Detection: Toggle the activation of LibinjectionXSS.

Scoring

Each HTTP request begins with an anomaly score of 0 and is incremented each time a rule match.

If a threshold (inbound or outbound) is reaches, corresponding actions are triggered.

Security Level: Increase or decrease the anomaly score threshold depending on the security level.

Critical anomaly score: The score added to the anomaly score when a critical rule match.

Error anomaly score: The score added to the anomaly score when an error rule match.

Warning anomaly score: The score added to the anomaly score when a warning rule match.

Notice anomaly score: The score added to the anomaly score when a notice rule match.

Block if global score exceeds: The score threshold that when reached will block the request.

HTTP Policy

Maximum number of arguments in request: The maximum number of parameters in the request.

Maximum argument name length: The maximum name length for each request parameters.

Maximum arguments value length: The maximum value length for each request parameters.

Maximum arguments value total length: The request arguments total length.

Request body limit: The request body length limit.

Maximum file size, in bytes: The maximum files size allowed to upload.

Maximum combined file size, in bytes: The maximum file size allowed to upload.

Allowed HTTP versions: Allowed HTTP protocol versions.

Allowed request content type: Allowed Content-Type headers.

Restricted extensions: The file extension allowed for requested files.

Restricted headers: Restricted (allowed) headers name.

DOS & BF Protection

You can here define an anti DOS policy by specifying the protected urls (or all), the burst time period, the threshold and the block timeout.

If a client requests the specified URL more than **"DOS Counter threshold"** times within a period of **"DOS Burst Time Slice"** seconds, he will be blocked for a period of **"DOS Block Timeout"** seconds.

Virtual Patching

WAF gives you the ability to use Virtual Patching, which generates a list of security rules from a vulnerability scan report. Virtual patching can understand VTMScan, Acunetix, Qualysguqrd WAS and OWASP ZAP Proxy.

For each application, a blacklist and a whitelist are auto generated.

Number of vulnerabilities processed

Number of security rules generated

Number of unsupported vulnerabilities skipped

Number of bad URLs (Rules not generated)

Learning Datasets

This menu lists previously built datasets.

Learning

When you enable learning mode, you will see learning datasets, listed after SVM one's. They are named following this pattern: <name of app in learning mode>-defender-whitelist

If you think that you have "learned" enough of the web traffic of your backend and wish to generate all the whitelists automatically, you can just click the flash logo.

Generated whitelists rulesets appear in WAF Ruleset; they are named with the following pattern: learning_<app name>_<app id> WL. Inside, whitelist rules are unique.

Packet Filter

This interface gives you the ability to fine tune the packet filter configuration.

General

Firewall settings

Configure Firewall Settings of: The WAF node that you want to configure.

Firewall Status

The summary of pf status, whether it is currently running or not.

You can also reload the service, which is mandatory after each configuration change.

Configuration

You can add packet filter rules here.

You have the possibility to tune

Policy: The action of the rule: Block / Pass

Direction: The direction of the rule: Inbound / Outbound / Both

Log: Toggle to log the event

Inet: IPV4 / IPV6

Protocol: TCP / UDP / ICMP (Ping) / All

Source: The source IP

Destination: The destination IP

Port: The targeted port

Rate limit: Default limit is 100 connections per second.

Comment: The rule comment

Action: Duplicate or delete the current rule

Blacklist / Whitelist

Permanent blacklist

Here you can enter one IP address or network range per line.

All connection coming from any IP of this table will be dropped.

Permanent whitelist

Here you can enter one IP address or network range per line.

All connection coming from any IP of this table will be allowed.

Applications

Add your websites/applications here. From this menu you can manage all your WAF's applications. You can change their configurations and start / stop / reload listeners.

After any configuration change, you need to click on "Reload" in order to synchronize changes.

This operation is safe and won't alter running applications, even if there is a mistake in your configuration: WAF first checks your configuration before stopping running processes.

Internet settings

Manage global application settings as well as settings relative to the "Internet side" of your WAF's applications.

Public URL mapping

Public FQDN: This is the fully qualified domain name of your web application, as users will type in their web browser. For example: www.example.com.

Public Alias: Another fully qualified name for your application. Use ',' if you want to add Multiple alias.

Public directory: The public path of your application. Default is '/'.

HTTP features and performance

Worker profile: The worker profile you want to use for your application.

Enable HTTP/2 protocol: If enabled this will activate HTTP/2 support for this application. Note that it implies you have at least a TLS profile for one of your listener and a correct HTTP/2 configuration in your worker profile.

Enable support of Microsoft RPC-Over-HTTP: If enabled this supports Microsoft RPC Over HTTP protocol.

Backend settings

Manage settings relative to the "Internal(private) side" of your WAF's application.

Application backend settings

.

Private URI: The private URI of your Web application. WAF will contact your application through this URL. It is better to use IP Address here instead of hostname, to avoid useless DNS requests. If your web backend needs a Host header associated to this IP address, you can add a custom 'Host' HTTP Request header in WAF.

Preserve incoming Host Header: By default, WAF will forward to the backend the hostname of the private URI defined above. If you want WAF to forward the public Host header requested by the client, toggle this option.

Timeout: This is the global timeout for communication with the backend, in seconds. Once this timeout is over, WAF will raise a HTTP 504 Gateway Timeout.

Disable connection reuse: This parameter should be used when you want to force proxy to immediately close a connection to the backend after being used, and thus, disable its persistent connection and pool for that backend. This helps in various situations where a firewall between WAF and the backend server (regardless of protocol) tends to silently drop connections or when backends themselves may be under round-robin DNS. To disable connection pooling reuse, set this property value to On.

Send KEEP_ALIVE messages to backend: This parameter should be used when you have a firewall between your WAF and the backend server, which tends to drop inactive connections. This flag will tell the Operating System to send KEEP_ALIVE messages on inactive connections and thus prevent the firewall from dropping the connection. To enable keepalive, set this property value to On.

TTL of inactive connection: Time to live for inactive connections and associated connection pool entries, in seconds. Once reaching this limit, a connection will not be used again; it will be closed at some later time.

Advanced settings

Send Proxy HTTP headers to backend: This determines whether or not proxy related information should be passed to the backend server through X-Forwarded-For, X-Forwarded-Host and X-Forwarded-Server HTTP headers.

Override backend' s HTTP errors: This directive is useful for reverse-proxy setups where you want to have a common look and feel on the error pages seen by the end user. This also allows for included files to get the error code and act accordingly. If enable, WAF will replace application's error page by the one specified in the Portal template.

Rewrite Cookie Path: If enabled (default), WAF will rewrite the 'path' argument of backend's Cookies. For example, if the WAF's public dir is '/public' and your private URI is '/', WAF will rewrite the Cookie "path=/" to "path=/public/".

Network settings

Manage settings relative to the listeners on which your application should be available. You cannot modify or remove a listener while it is running.

IP Address: Select the listener on which you want WAF to listen for incoming connection.

TCP port: The TCP port to listen onto.

SSL Profile: If you want TLS to be enable on the listener, select your TLS Profile

Security settings

Manage the security settings of your application.

Global settings

Use ACLs: Choose here one or more ACLs profiles

Redirect http to https: If enabled, WAF will automatically redirect incoming HTTP request to HTTPS request. For that it will send an HTTP 301 Permanent Redirect response. Not that it can't work if you don't have at least one TLS profile associated to one of your listeners.

Allowed HTTP Method: By default, WAF only allows HEAD, GET and POST requests. If you need support for other HTTP METHOD, simply add them here. There is an auto-completion feature that helps you to add valid HTTP METHOD. For Microsoft RPC-Over-HTTP you will typically need to add the HTTP METHOD "RPC_IN_DATA" and "RPC_OUT_DATA" here.

Enable backend's cookies encryption: If enabled, will encrypt ALL the cookies sent by the backend using a randomly generated key and iv with the selected cipher. (Note: Changing the cipher will invalidate all the current cookies)

Cipher to use for backend's cookies: Choose here the cipher to use for backend's cookies encryption.

Source IP Reputation analysis

The source ip of clients will be checked against the reputation database which is a MaxmindDB file generated daily. If the IP is found, a 403 forbidden will occur.

When an IP is found in the database, you will be able to see its tags in the log viewer.

Block IP with the following tags: Gives you the ability to block some tags associated to source IP addresses.

If reputation blocking is enable, WAF will check if the client ip is found, it will retrieve the corresponding tags.

If one of the tags match a forbidden tag, WAF will send a 403-FORBIDDEN HTTP response.

Enable GeoIP: Toggle this setting if you want WAF to lookup the country of the source IP Address. This is based on GeoLite2 data created by MaxMind.

Block following countries: Enter the 2-letters code of countries you want to deny access from (blacklist).

Only allow following countries: Enter the 2-letters code of country you want to accept. Other countries will be denied (whitelist).

High-precision GeoIP: Toggle this setting if you want to look up the city of the source IP Address.

WAF policy settings

Protect application with WAF profile: Select here a WAF profile that will be used to protect your application against Web attacks.

Enable learning mode: If enabled learning dataset is generated in Dataset.

Block in learning mode: If you want to block attack in learning mode, enable this option.

WAF Ruleset settings

Choose the Rules Set to use: Select WAF ruleset for application security.

URL specific Rules Set to use: You can specify WAF ruleset for specific URL or public dir.

Request header settings

Manage the requested headers, sent from client's web browser to WAF. This feature is useful to add/modify headers sent by the client, before they are transmitted to your web application.

A typical usage of this feature is to add the 'Host' header for backends that use

Response header settings

Manage the response headers, sent from the Web application to client's web browser. This feature is useful to add/modify headers sent by the web application, before they propagate to clients. A typical usage of this feature is to modify response header of the backend.

Content Rewriting

Manage the content rewriting rules. It allows WAF to rewrite any part of your application's response, from headers to body.