Administration Manual

Web Security Manager 4.4

www.alertlogic.com

support@alertlogic.com

August, 2015

Web Security Manager Web Application Firewall xi
1. Getting started 1
1. Connect to the Web Security Manager web management interface 2
1.1. Navigating Web Security Manager web management interface 2
2. Basic system configuration 4
3. Website configuration 5
4. Testing if it works 8
4.1. Change / configure DNS for the website. 8
4.2. Test connectivity 8
5. View the website deny log 9
6. Change default passwords 10
6.1. admin user 10
6.2. operator user 10
7. Getting help 11
2. Dashboards 13
1. Deny Log 14
1.1. Interactive graph 14
1.2. Interactive list 14
2. Learning 18
3. System 19
3.1. System status 19
3.2. Interfaces 19
3.3. Modules 19
3.4. Disk I/O 20
3.5. Disk 20
3.6. Read-only monitor access 20
3.6.1. As HTML 20
3.6.2. XML format 20
4. Traffic 21
4.1. Interfaces 21
4.2. Traffic by website 21
3. Services 23
1. Websites 24
1.1. Websites list 24
1.1.1. Defined websites 24
1.2. Adding a website 24
1.2.1. Virtual web server 24
1.2.2. Real web servers 25
1.2.3. Default Proxy 26
1.2.4. Initial operating mode 27
1.2.5. Removing a proxy 27
1.3. Global 27
1.3.1. Source based blocking 27
1.3.2. Server ID 28
1.3.3. HTTP request throttling 28
1.3.4. HTTP connection limiting 30
1.3.5. SSL 31
1.3.6. HTTP global request limit 33
1.3.7. HTTP error log level 33

1.3.8. HTTP global access logging 33
2. Network 34
2.1. Blacklisted Source IPs 34
2.2. Network blocking bypass 35
2.2.1. Allowing an IP address to bypass network controls 35
2.3. DoS mitigation 36
2.4. Attack source Auto blocking 36
2.5. Network routing 37
4. Application Delivery Controller (ADC) 39
1. Virtual host 40
1.1. Deployment 40
1.1.1. Reverse Proxy 40
1.1.2. Routing Proxy 40
1.2. Virtual web server 41
1.3. SSL Certificate 42
1.3.1. Importing the SSL certificate 42
1.3.2. Exporting certificate from web server 43
1.4. Virtual host aliases 43
1.4.1. Wildcards 44
1.4.2. Default Proxy 44
1.5. Timeouts 44
1.6. HTTP Request and Connection Throttling 45
1.6.1. HTTP request throttling 45
1.6.2. HTTP connection throttling 46
1.7. Client Source IP 46
1.7.1. X-Forwarded-For 46
1.7.2. Other X-headers 47
1.7.3. Trusted proxy 47
1.7.4. Transparent Proxy 49
1.8. Redirects 50
1.8.1. Match types 50
1.8.2. Prefix match 50
1.8.3. Regex match 52
1.8.4. Vhost regex match 53
1.8.5. Examples summary 54
1.9. Lower button bar 54
2. Load balancing 55
2.1. Real web servers 55
2.2. Timeouts 56
2.3. Load balancing settings 56
2.4. Health Checking 58
2.5. Insert request headers 60
2.5.1. Request header variables 61
2.6. Advanced settings 61
2.7. Lower button panel 62
3. Caching 63
3.1. Static Caching 63
3.2. Dynamic caching 64
3.3. Lower button bar 65

4. Acceleration 66
4.1. Compression 66
4.1.1. Compression level 66
4.1.2. Compress response content-types 66
4.1.3. Exceptions 66
4.2. TCP connection reuse 67
5. Statistics 69
5.1. Interval selection 69
5.2. Summary section 69
5.3. Compression and served from cache graph 70
5.4. Requests total and served from cache graph 70
5.5. Original data and data sent graph 71
5.6. Lower button bar 71
5. Web application firewall (WAF) 73
1. Policy 74
1.1. Validation order and scope 74
1.2. Basic operation 75
1.2.1. WAF operating mode definitions 75
1.2.2. Request parsing 77
1.2.3. Attack class criticality 80
1.2.4. Source IP tracking and blocking 80
1.2.5. External notification 82
1.2.6. Deny log settings 83
1.2.7. Access log settings 86
1.2.8. Mirror proxy policy from master 89
1.3. Protocol restrictions 89
1.3.1. Allowed HTTP methods, protocol versions and web services 89
1.3.2. Headers, restrict length and number 91
1.3.3. Cookies, restrict length and number 92
1.3.4. Request, restrict length and number 93
1.3.5. File uploads, restrict size and number 94
1.3.6. Request parameters, restrict size and number 95
1.4. Website global policy 97
1.4.1. Validate static requests separately 97
1.4.2. URL path validation 98
1.4.3. Denied URL paths 99
1.4.4. Query and Cookie validation 100
1.4.5. Headers validation 102
1.4.6. Attack signatures usage 103
1.4.7. Session and CSRF protection 104
1.4.8. Trusted clients - IP whitelisting 107
1.4.9. Trusted domains 108
1.4.10. Evasion protection 109
1.4.11. Time restricted access 110
1.4.12. Input validation classes 111
1.5. Web applications 113
1.5.1. Web application settings 113
1.5.2. Global violation action override 114
1.5.3. Methods allowed 114

1.5.4. Session protection 115
1.5.5. Parameters 116
1.6. Output filter 118
1.6.1. Backend server cloaking 118
1.6.2. Output headers validation and rewriting 119
1.6.3. Output body validation and rewriting 120
1.7. Authentication 122
1.7.1. SSL client authentication 122
1.7.2. SSL client Certificate Revocation Lists (CRLs) 123
1.7.3. SSL client authorization 124
1.8. Regular expressions 125
1.8.1. What are regular expressions 125
1.8.2. Metacharacters 126
1.8.3. Repetition 126
1.8.4. Special notations with \ 127
1.8.5. Character sets [...] 127
1.8.6. Lookaround 128
1.8.7. Examples 128
1.8.8. Further reading 130
2. Deny and error handling 132
2.1. Deny action 132
2.2. Error messages 132
2.2.1. Document not found (error 40x) 132
2.2.2. Authentication required (error 403) 135
2.2.3. Server error (error 50x) 137
2.3. Lower button bar 139
3. Learning 140
3.1. Learning data 140
3.1.1. Applications learned 140
3.1.2. Global parameters learned 141
3.1.3. Static content learned 142
3.1.4. Tools 142
3.1.5. Lower button bar 144
3.2. Learning status 144
3.2.1. Learning progress indicators 144
3.2.2. Policy history 145
3.2.3. Resulting policy 145
3.2.4. Sample run information 147
3.2.5. Lower button bar 147
3.3. Learning settings 148
3.3.1. Policy generation options 148
3.3.2. Global parameters 150
3.3.3. Policy verification 151
3.3.4. Learning thresholds 152
3.3.5. Learn data sampling 156
3.3.6. Lower button bar 157
4. Log 158
4.1. Deny log 158
4.1.1. Specifying filter criteria 158

4.1.2. Blocked and failed requests 160
4.1.3. Lower button bar 165
4.2. Access log 165
4.3. Access log files 165
5. Reports 167
5.1. Reports 167
5.2. Generated reports 167
6. System reference 169
1. Clustering 170
1.1. Cluster virtual IP configuration 170
1.2. Synchronization configuration 170
1.3. Cluster configuration examples 173
1.3.1. Configuring a fail-over cluster 173
1.4. VRRP Interfaces 174
1.5. Fail-over status information 175
2. Configuration 176
2.1. Network 176
2.2. Static routes 177
2.3. Syslog - logging to external host 178
2.3.1. Mapping of Web Security Manager System Logs to Syslog facilities 179
2.4. SNMP 179
2.5. Date and Time 180
2.6. Admin contact 181
2.7. Email system alerts 181
2.8. Forward HTTP proxy 182
2.9. Backup configuration 183
2.9.1. FTP configuration 183
2.9.2. SCP configuration 184
2.10. Auto-backup 185
2.11. Remote access 186
2.12. Management GUI 186
2.12.1. Password requirements 186
2.12.2. Login and session restrictions 187
2.12.3. SSL certificate 188
2.13. FIPS 140-2 validated mode 189
2.13.1. Validation of FIPS mode 189
2.13.2. Enabling FIPS 140-2 validated mode 190
3. Information 191
3.1. System 191
3.2. Web Security Manager 191
3.3. Devices 191
3.4. Disks 191
3.5. Currently logged in users 192
4. Interfaces 193
4.1. IP configuration 193
4.2. Role 194
4.3. Media settings 195
5. Logs 196

6. Maintenance 197
6.1. Backup and restore 197
6.1.1. Best effort - restoring to different platforms 197
6.1.2. Local backup 197
6.1.3. Restore 197
6.2. Website templates list 198
6.3. Databases 198
6.4. Website access logs list 198
7. Tools 200
7.1. Network tools 200
7.1.1. TCP connect test 200
7.1.2. Network debug 200
7.2. Reboot and Shutdown 201
7.3. Technical information for support 201
7.4. License information 201
8. Updates 203
8.1. Updates available for installation 203
8.1.1. Installing updates 203
8.2. Installed updates 203
8.3. Configuring for updates 203
9. Users 204
9.1. User accounts 204
9.1.1. Built in user accounts 204
9.1.2. Additional accounts 204
9.2. Current user 204
9.3. System users 204
7. The command line interface 207
1. Accessing CLI 208
1.1. Console access 208
1.2. SSH access 208
2. Command reference 209
2.1. show interfaces 209
2.2. show interface 209
2.3. show gateway 209
2.4. show hostname 209
2.5. show routes 209
2.6. show version 210
2.7. set gateway 210
2.8. set interface 210
2.9. set password 210
2.10. set user 210
2.11. system backup run 210
2.12. system cache flush 210
2.13. system ping 211
2.14. system updates fetch 211
2.15. system updates query pending 211
2.16. system updates query installed 211
2.17. system updates install 211
2.18. system status 211

2.19. system restart 212
2.20. system shutdown 212
2.21. system reboot 212
2.22. system remotesupport 212
2.22.1. View remote support status 212
2.22.2. Enable remote support 213
2.22.3. Disable remote support 213
2.23. quit 213
8. Network deployment 215
1. Simple single-homed Web Security Manager implementation 216
2. Firewalled single-homed Web Security Manager implementation 217
3. Firewalled Web Security Manager implementation with a fail-over/backup Web Security Manager 218
4. Dual-homed performance optimized Web Security Manager implementation 219
9. Frequently Asked Questions 221
1. Deployment 222
2. Client issues 223
3. SSL Certificates 224
4. Troubleshooting 225
5. Clustering 226
6. Accessing Web Security Manager management interfaces 227
7. Learning 228
8. Filtering 229

Web Security Manager Web Application Firewall

Web Security Manager Web Application Firewall is implemented in the network as a filtering gateway which validates all requests to the web systems.

Web Security Manager defends against all OWASP Top Ten vulnerabilities, supports XML web services and provides full PCI DSS Section 6.6 requirements compliance.

The following modules are included providing acceleration, scalability and proactive protection of web systems:

Load Balancer

Enabling scalability and acceleration of even complex SSL-enabled stateful web applications.

Web Accelerator and cache

Reducing traffic cost, improving response time and off-loading web servers.

Web Application Firewall

Proactive protection of web servers and web applications by employing a positive security model providing defenses against all OWASP top ten vulnerabilities.

Web Security Manager includes a hardened OS and installs on most standard hardware. The Web Security Manager software appliance installer turns a piece of general purpose application server hardware into a dedicated application acceleration and security gateway within minutes - with minimal interaction.

The Web Security Manager software appliance combines the flexibility and scalability advantages of software with the security advantages and administrative simplicity from dedicated hardware appliances.

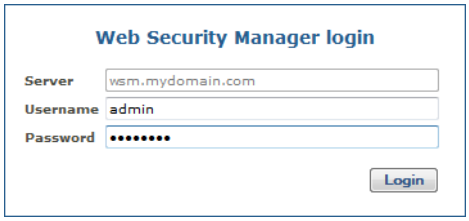
Automated application profiling, adaptive learning, positive and negative filtering and support for XML based web services allow for out of the box protection against attacks from malicious hackers and worms.

As the website is learned Web Security Manager gradually turns towards a positive, white-list based, policy providing protection against attacks targeting undisclosed vulnerabilities in standard software and custom built applications.

Chapter 1

Getting started

1. Connect to the Web Security Manager web management interface



Access the Web Security Manager management interface by opening a web-browser and entering URL https://websecuritymanager_ip_address:4849 (note HTTPS). The management address in the example installation is: https://192.168.3.20:4849.

If you are accessing the management interface for the first time, you will be asked for a license key.

Enter the license key provided in your License key and support contract information letter (PDF) and click the "Activate" button. After successfully entering the license key, you are asked to agree to the Web Security Manager license agreement. After you have read and agreed to the license agreement, you are redirected to the Web Security Manager management login screen.

Log in using username “admin” and password [last nine characters of license key in reverse order]. Please change the password after the initial login. Instructions for changing your password are found below.

1.1. Navigating Web Security Manager web management interface

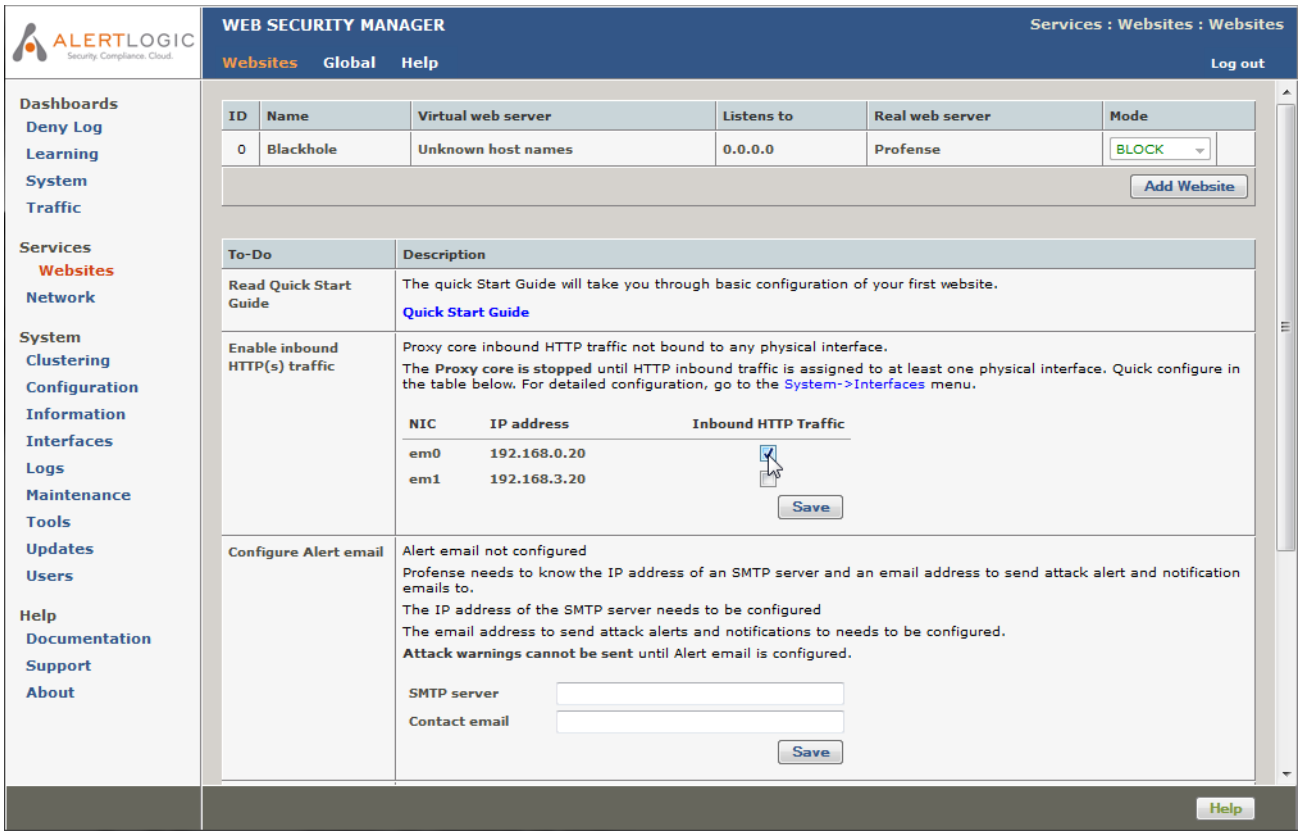


Figure 1.1. The management interface

After successful login, you will be presented with the management interface website overview page. The management interface is divided into 4 main sections:

Dashboards

A quick overview of denied requests, traffic, system status and learning progression.

Services

Configuration and management tool for all website proxies, including policy, caching, acceleration, load balancing, HTTP request throttling and DoS mitigation settings.

To add a website or to select a website for management click **Services** → **Websites**

System

Configuration of system parameters like network interfaces, IP addresses, fail-over, network settings (DNS, NTP, SMTP), viewing of system logs and status information, including administration of updates, backup and configuration restore.

Main (vertical) menu system is on the left side of the screen. Content assigned to the menu item is displayed on the right side of the screen. An additional horizontal menu system appears where applicable.

Help

Access to help and support related information including documentation, version information and support links. The complete manual is available in HTML and PDF versions on the **Documentation** page.

On any page, clicking on **Help** in the horizontal menu will display the manual reference section specific for that page.

2. Basic system configuration

To make sure essential system configuration tasks are not forgotten, a to-do list basic system configuration tasks is displayed. When an item is done it will disappear from the list. When the first website is added the "read Quick Start Guide" item will disappear.

Enable inbound HTTP(s) traffic	Select which network interfaces you want to respond to inbound HTTP/HTTPS requests from clients.
Configure Alert email	Web Security Manager needs to know an SMTP server and an email address it can send log warnings, update notifications, etc. to. SMTP server: Enter the address of an SMTP server that is reachable and accepts SMTP requests from Web Security Manager. Contact email: Enter the email address to send notifications to. This item can be skipped but it is recommended.
Configure DNS	IP address of one or more DNS servers. ***Valid input*** IP addresses Use space to separate multiple hosts (only one required). ***Input example*** `192.168.0.1`
Configure time synchronization	IP address or host name of an NTP server. Remember to set up at least one DNS server if you enter a host name here. ***Valid input*** IP address or fully qualified domain name. Use space to separate multiple hosts (only one required). ***Input example*** `time.nist.gov`

3. Website configuration

Now configure a website.

1. Select **Services** → **Websites** in the left menu pane. This will take you to the **websites overview** page.
2. Click on the **Add Website** button.

The **Services** → **Add** page is displayed.

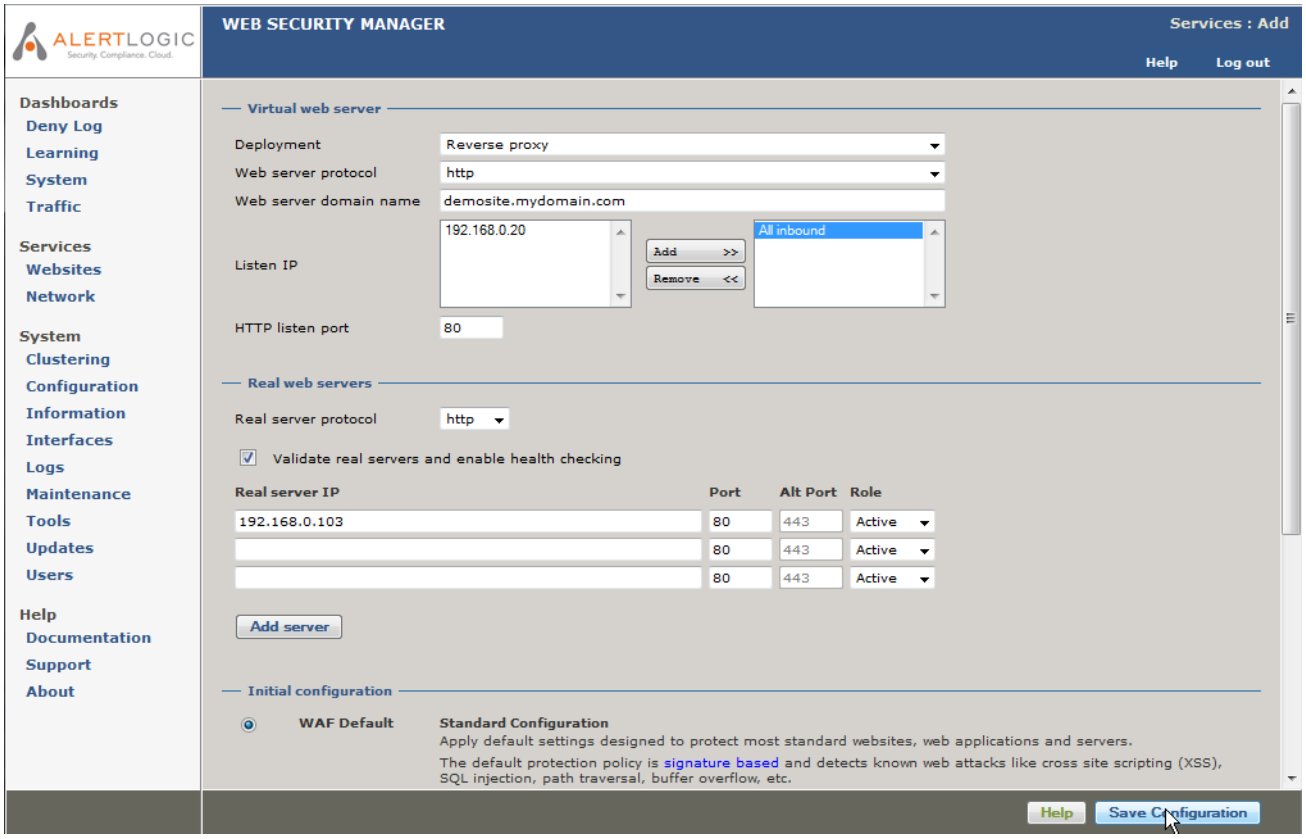


Figure 1.2. Add website page

In the **Virtual web server** section you configure the part of the website proxy that the clients connect to.

1. In **Deployment** select either **Reverse proxy** or **Routing proxy**.

   Both deployments terminate client requests and proxies requests to the backend real server but while Reverse proxy requires an IP address to be configured on the WSM node the Routing proxy deployment routes traffic to the backend server but intercepts traffic for the configured ports, processes it and proxies it to the backend.

   For routing proxy deployments make sure that IP forwarding us enabled in **Services** → **Network** → **Network routing**.

2. In **Web server protocol** select either **HTTP**, **HTTPS** or **Both**. The latter will create a website proxy that responds to both HTTP and HTTPS requests.

   When selecting **HTTPS** or **Both** as the protocol a temporary certificate will be generated. When the new proxy is created the certificate can be replaced by importing the real certificate

in **Services** → **Websites** → **ADC** → **Virtual host**. Click Help in that section to get instructions.

3. In **Web server domain name** enter the address of the web server you want to protect. The address is the one users enter in the browser to go to the website.

   In the example demosite.mydomain.com is entered.

4. In **Listen IP** select the IP address(es) you want the web server to respond to. For HTTP websites All inbound can be selected. This will configure the website proxy to respond to all IP addresses that are configured to accept inbound requests. For HTTPS proxies it is mandatory to select a specific IP address.

5. In **HTTP(s) listen port** select the port(s) you want the website to listen to. For HTTP proxies the default is 80 and for HTTPS proxies the default is 443. When creating a website proxy that serves both HTTP and HTTPS two input fields will appear.

In the **Real web servers** section you configure how the website proxy communicates with the backend web servers.

1. In **Real web server** enter the address of the web server you want Web Security Manager to redirect allowed client requests to. This address is the address of the web server you want to protect. In the example 192.168.0.103 is entered.

2. In Real server protocol select the protocol you want Web Security Manager to use when connecting to the backend web servers. If you want the traffic to the backend web servers to be encrypted select HTTPS otherwise leave it at the default HTTP.

   Note that you should only select HTTPS if it is necessary. HTTPS puts an extra burden on the backend web servers.

3. Decide on real servers health checking. When **Validate real servers and enable health checking** is checked Web Security Manager will connect to the backend servers automatically find a suitable target page to use for health checking. If health checking is not enabled backend server status will not be monitored by Web Security Manager.

4. For each backend web server that is serving the website (demosite.mydomain.com in this example) enter the IP address and port in the real servers list.

   Real server IP and Port: the IP address / port combination the web server is listening on. Typically Address:80 for HTTP servers and Address:443 for HTTPS servers.

   Role: Select Active, Backup or Down. Active means that requests will be forwarded to the server. When Backup is selected the server will only be used if no Active servers are in operation. Down means that the server should not be used - for instance if it is down for maintenance.

Finally, In the **Initial configuration** section, select the initial configuration template to apply to the website proxy.

Now click the **Save Configuration** button in the lower right corner of the page.

This will save your configurations and take you back to the **websites overview** page.

1. Click the blinking link **apply changes** that appears in the upper right corner of the page to apply those changes to your configuration of Web Security Manager.

Figure 1.3. Websites overview page

The Web Security Manager Web Application Firewall is now protecting the configured website.

4. Testing if it works

Now test your newly configured website.

4.1. Change / configure DNS for the website.

For testing purposes, make the website domain name resolve to the Web Security Manager IP address for example by adding the IP address and domain name to the hosts file on your PC.

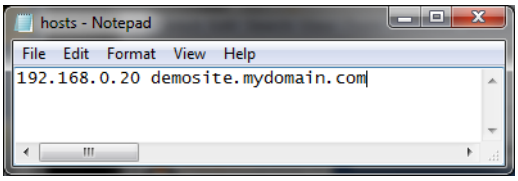


Figure 1.4. Editing the hosts file

4.2. Test connectivity

In a new browser page (or tab) enter the address of the website you configured.

You should see the home page of the website and it should be served by Web Security Manager.

To check that Web Security Manager is serving the content, enter an URL that will match an attack signature. To match the path traversal signature (for instance) append the parameter print=../../../etc/somefile to a page.

http://demosite.mydomain.com/testpage.php?print=../../../etc/somefile

If the page is served through Web Security Manager you will get:

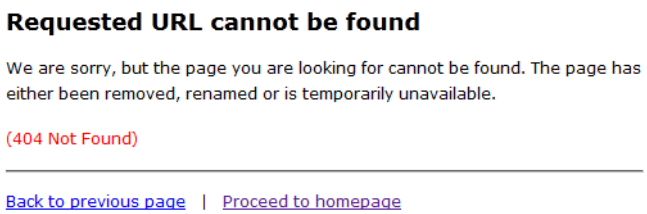


Figure 1.5. Default deny page

If the above is not displayed, please restart your browser and / or flush your DNS cache by running cmd.exe (on your PC) and enter **ipconfig /flushdns**. Then try the request again.

5. View the website deny log

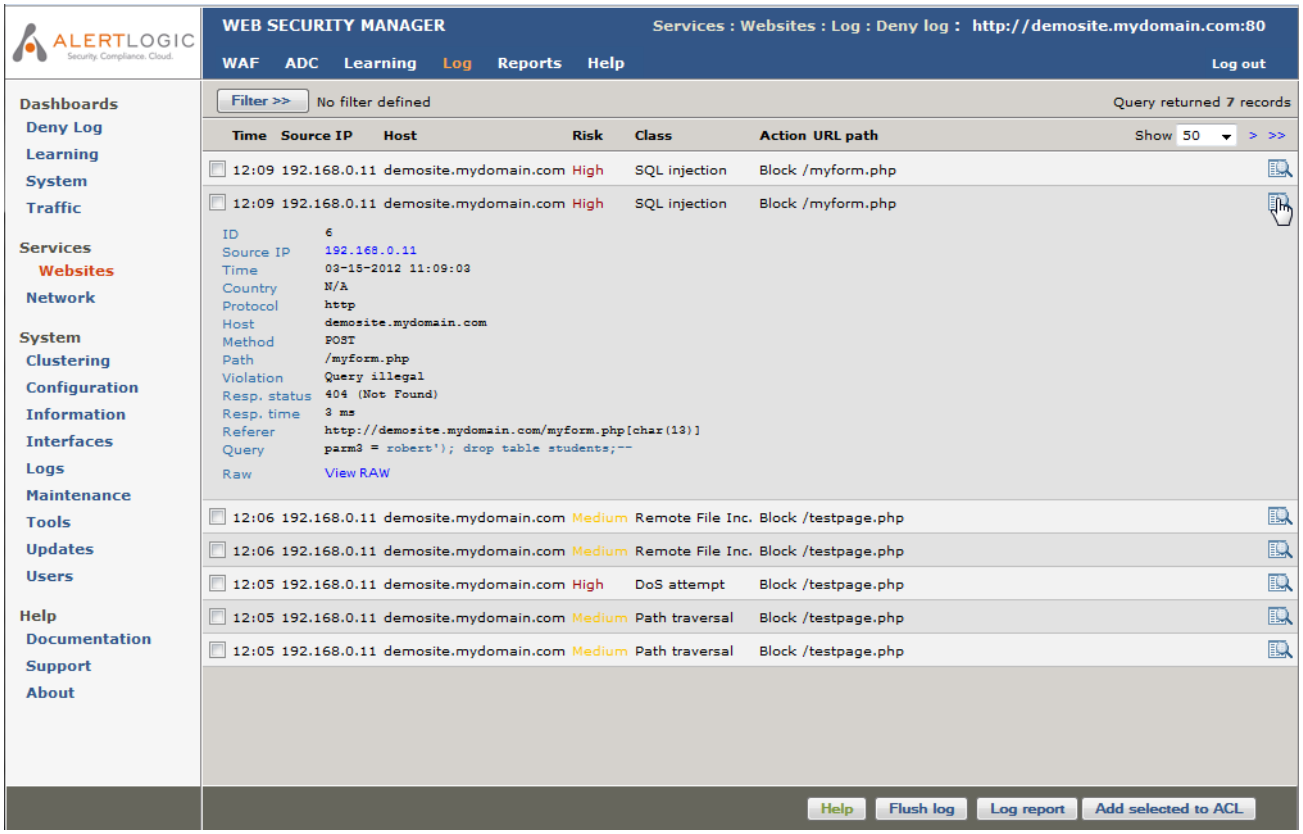


Figure 1.6. Deny log

In the Web Security Manager management interface select **Web Firewall** → **Websites** in the left vertical tool bar . The websites overview page will be displayed. Select the website by clicking on it.

When selecting a website the landing page is the Deny Log.

To view details of a log entry click the Inspect icon in the right most column of the list as in the example above.

6. Change default passwords

Now change the default passwords for the admin user (web based management interface) and the operator user (the system console) by completing the following:

6.1. admin user

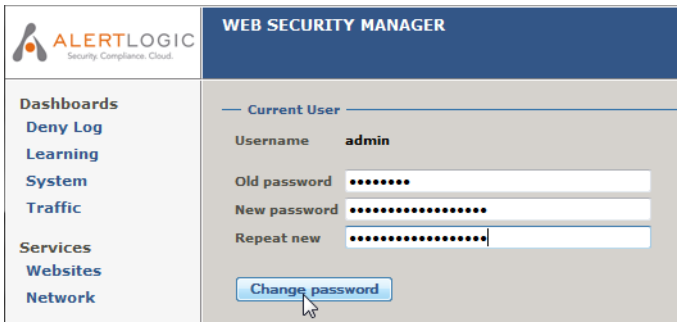


Figure 1.7. Password change page

Change the administrator password from the default value in:

System → **Users**

Change the password for the console user Operator in the console.

6.2. operator user

1. Log in to the console with

   user name: operator

   password: changeme

2. Enter the command **set password**

```
login: operator
Password: changeme

Web Security Manager command-line management interface

psh> set password
Changing local password for operator.
Old password: changeme
New password: R0dsQAVg
Retype new password: R0dsQAVg
psh> quit

Web Security Manager/amd64 (ttyC0)

login: _
```

7. Getting help

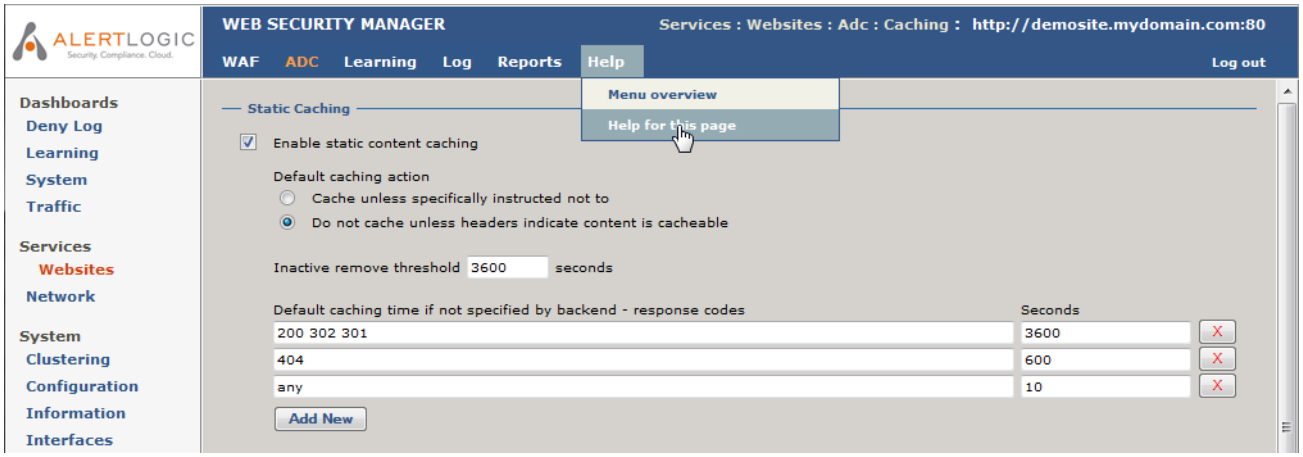


Figure 1.8. Context specific help

By clicking the green Help menu item in the horizontal menu the relevant section in the manual is opened in a new window.

Chapter 2

Dashboards

1. Deny Log

In Web Security Manager websites have separate security policies and deny logs. This allows for fine grained tuning of policies and makes it easy to provide detailed reporting to management and application/web site owners. For the security administrator it is necessary though to have the ability monitor the deny log for all websites. The deny summary window provides such functionality by summarizing log data for all configured websites. The window consists of two sections:

1. An interactive graph with drill down functionality which summarizes all deny log events in a column graph.
2. A more detailed interactive list with drill down functionality which shows deny log events for all websites above a configured risk level (default medium).

Both elements provide drill down functionality which will allow for narrowing in on events in the specific websites deny log.

1.1. Interactive graph

The interactive column allows for zooming in on log events through 3 levels.

For all three levels the date selector allows for scrolling through historic log events and Hovering the pointer over a column will display the exact number of requests for that category.

1. *By date and risk.*

   For each date in the selected period deny log events are shown divided into the 5 risk categories `critical` through `none`.

   Clicking one of the columns will zoom in on that date taking you to level 2.

2. *By website and risk.*

   For each website/application deny log events are shown divided into the 5 risk categories `critical` through `none`.

   Clicking one of the columns will zoom in on log events for that website for the specific date selected.

3. *Single website by attack class.*

   The lowest level of the interactive graph shows log events for a specific website by attack class, sql injection, XSS, etc. By default log entries are only shown for one day but the interval can be extended by selecting a different interval using the **Show** drop-down in the date selector.

   Clicking on an attack class column will take you to the deny log of the website creating a filter that shows only log entries satisfying the selection in the interactive graph.

1.2. Interactive list

The interactive list shows log entries above a configurable risk level for all websites.

Blue column headings indicate that the result can be sorted by that column. Clicking the same column will toggle sort direction (asc/desc).

The top level of the list shows attacks summarized by either source IP or country. Clicking on a row will display a list showing the number of attacks showed in the attacks column. When the list is summarized by IP the list will show log records from all websites from that specific source IP.

When the list is summarized by country the list will display log records from all websites summarized by source IP. Clicking on a row will show details from that specific IP.

When showing IP details, clicking the details icon in the rightmost column of the list will display details from that log event.

The description of the columns below apply to all detail levels of the list. Some columns are specific for a level and will not be visible in other.

By default the list shows all records for a maximum of 90 days. By checking **Limit to Graph interval** the list can be set to only display records for the interval specified in the graph above.

Source IP	Source IP the requests originated from.
Country	Country the requests originated from.
Attacks	Total number of attacks recorded from country/IP. Click row to zoom in on attacks.
Last seen	Date and time the last request from IP/Country was logged. By default results are sorted by date.
Risk	Risk classification of the log entry. Options are: • Critical • High • Medium • Low • None
Attack Class	Attack classification of the log entry. Options are: • SQL injection • XPath injection • SSI injection • OS commanding • XSS (Cross Site Scripting) • Path traversal • Enumeration • Format string • Buffer overflow • DoS attempt • Worm probe • Access violation • Malformed request • Session invalid • CSRF • Session expired

	• Broken robot • Broken int. link • Broken ext. link • Other • None • False positive • Friendly
Violation	Shows the general violation description as defined by Web Security Manager. Options are: • Generic violation • Header unknown • Header illegal • Path unknown • Query unknown - no policy rules match the name of the parameter. • Query illegal - a policy rule is matching name of the parameter but the parameter value does not match the corresponding regular expression for validating the input value. • Header length • Missing hostname • Invalid hostname • Header failed • Path denied • Upload attempt • Payload length • Session validation failed • Form validation failed • Session expired • Malformed XML • Content type not enabled - Content type is supported but not enabled. • Negative match
Action Showed only in IP details view.	Block action taken on the request. Options are: ***Allow*** The request was allowed, either because the current mode and whitelist configuration or because the requests was allowed according to policy. If the request was allowed by policy the reason for the

	request being logged in the deny log is typically that the backend server responded with an error. Expand the request to see details. ***Block*** The request was blocked by Web Security Manager. ***Block-IP*** The request was blocked by Web Security Manager and the source IP was blacklisted resulting in further requests from that source being blocked at the network level. ***Strip*** The offending part of the request was stripped before allowing the request. Used for instance to remove session cookies for expired sessions.
Time	Date and time the request was logged.
Method Detail - click details icon to view.	Offending method (if any)
Resp. status Detail - click details icon to view.	If applicable shows the response status from the backend server like `404 not found` or `200 (OK)`.
Resp. time Detail - click details icon to view.	The time from Web Security Manager received the request and forwarded it to the backend server until the response is sent to the client from Web Security Manager.
Referer Detail - click details icon to view.	The refering source, internal or external, from which the request originated.
Header Detail - click details icon to view.	Offending header fields and values (if any).
Query Detail - click details icon to view.	Offending parameter names and values (if any).
Raw Detail - click details icon to view.	Shows the original request as send by the client. To view it, click on the **View RAW request** button.

2. Learning

Key learning indicators for each website are displayed in an overview table.

Website	Website name as configured in Web Security Manager.
Samples	The total number of requests processed during the learning process.
URL paths	Total number of unique URL paths identified.
Parameters	Total number of unique parameter names identified. Uniqueness is determined by URL path. Two parameters with the same name but mapped as belonging to different URL paths are therefore identified as two unique parameters. When the policy is built Web Security Manager identifies parameters with similar names and input data as as global in scope and builds global patterns matching such parameters.
Sampling progress	An indicator bar showing the progress of the sampling process. Sampling is the process of collecting information about the website in terms of what paths/applications are used, what parameters do they take as input, what extensions are used for static content, etc.
Verification Progress	An indicator bar showing the progress of the verification process. The verification process 1) validates the data samples using statistical methods like analyzing spread in IP sources and time, number of requests, etc. and 2) verifies that the resulting policy covers the requests sampled. As the Web Security Manager Learner extracts characteristics like extensions, specific directories in paths and global parameters (parameter names a number of applications take as input - like print=1) and even patterns used in global parameters the verification process may start before the Data sampling progress has reached 100%. Verification is calculated as the number of sample runs in a row with no policy changes relative to the required number configured in learner settings. When Verification has reached 100% Web Security Manager will either build and commit a new policy or notify the administrator by email that verification has reached 100% and a new policy can be built and committed.

3. System

The Status Monitor page displays system monitoring information.

The monitor page can be viewed as a separate read-only page without the menu system. Also the information is available in XML-format. See velow for more information.

3.1. System status

Displays current system usage statistics.

The window is constantly updated.

CPU usage	CPU load.
Load	Average system load.
Memory (physical)	Free and total system memory in megabytes.
Memory (swap)	Maximum and used swap memory in megabytes/kilobytes.
Files/Sockets	Open files and sockets.
Processes	Number of running processes.
Network buffers	Currently used network buffers, peak usage and available.

3.2. Interfaces

Displays various interface information parameters.

Interface	Interface description/name.
Status	Physical interface status.
System IP	Current system IP address for the interface.
In data	Incoming data.
Out data	Outgoing data.
In pkts.	Incoming packets in packets per second.
Out pkts.	Outgoing packets in packets per second.

3.3. Modules

Displays status and memory usage for important system components.

In the graphical user interface in general daemons can have the values **OK** (in XML output 1) or **ERROR** (in XML output -1). The Sync Daemon and Proxy core have some extra status codes that are explained below.

Proxy core	Proxy core components. Status code can be STOPPED indicating that no physical interfaces are bound to the proxy core (XML output 2) or that Proxy core is stopped by the system since no proxies are defined (XML output 0).
ADM daemon	Admd - The administrative layer.
Stats daemon	The subsystem recording proxy statistics.
Log daemon	The subsystem handling logging.

Learner daemon	The automated learner.
Alert daemon	The subsystem issuing attack alerts via Syslog and email.
Sync daemon	The subsystem which handles cluster synchronization. Status code can be INACTIVE indicating that synchronization is not enabled and the Sync daemon therefore is not running.
Health check daemon	Daemon checking backend servers.

3.4. Disk I/O

The Disk I/O section shows disk activity information.

Read	Data read from disk per second.
Write	Data written to disk per second.
I/O ops.	I/O operations per second.

3.5. Disk

The disk section shows disk usage information per partition.

/log	Log partition.
/cache	Content cache partition.
/db	Access policy partition.
/wsm	Applications partition.

3.6. Read-only monitor access

To view the monitor page directly or using an XML client follow the instructions below.

3.6.1. As HTML

Click the monitor button in the lower button bar on the monitor page. This will open a new window.

3.6.2. XML format

Access the address https://address_of_management_interface:4849/monitor.html?xml using an XML client.

An XML data structure with the values above will be returned. Note however that the units can be different from the HTML output. The XML keys therefore the unit the value is returned in.

4. Traffic

The monitor window provides an overview of configured proxies. The overview includes real time traffic information.

4.1. Interfaces

Displays various interface information parameters.

Interface	Interface description/name.
Status	Physical interface status.
System IP	Current system IP address for the interface.
In data	Incoming data.
Out data	Outgoing data.
In pkts.	Incoming packets in packets per second.
Out pkts.	Outgoing packets in packets per second.

4.2. Traffic by website

Name	Total number of requests.
Services	Number of services configured.
Requests	Total number of HTTP requests received.
Responses	Total number of HTTP responses sent.
40x	Total number of responses af type 40x (404 - Not Found, 403 - Forbidden, etc). Unless denied requests are redirected 40x include denied requests.
50x	Total number of responses af type 50x (502 - Bad gateway, 500 - Internal Server Error, etc). These responses typically indicates that real servers are not responding withion the real server time out or that they are in error state.
Received	Total data received.
Sent	Total data sent.
Compression	Total compression ratio for the proxy. Eg. 60% means that the total original data was compressed to the 60% of it's original size.
Backend	Backend status OK or ERROR. When status is in parentheses the backend status is not being managed by the health checking daemon.
24hr Health %	Even when health checking is not enabled, WSM keeps track of backend availability and latency by querying the configured backend servers at 1-minute intervals and displays a calculated health score based on the last 24 hours of health monitoring. The score is calculated ratio of responses with a response code lower than 400. To display the result of the actual health health monitoring checks, click the magnifying glass symbol.

Mode	The mode the proxy is running in.
Details icon	Click to manage proxy settings.
Graph icon	Click to display traffic information graphs.

Chapter 3

Services

1. Websites

The Website menu gives access to all configuration options related to website security profile, management, ACL administration, security logging, and settings.

To manage website security profiles select **Services** → **Websites** in the left menu pane. This will take you to the website overview page.

1.1. Websites list

1.1.1. Defined websites

Displays the list of configured website security profiles in the system. The list shows the `id`, `virtual host`, `real host` and current `running mode` for each configured proxy.

1.1.1.1. Selecting a website proxy for management

To manage a configured proxy simply click on it in the defined proxies list.

1.1.1.2. Changing operating mode

In the list of configured website proxies select the new operating mode in the **Mode** drop-down box for the website proxy to be changed.

1.2. Adding a website

Path: **Services** → **Websites**+**Add Website**.

1.2.1. Virtual web server

Deployment Drop down list	The proxy deployment mode. ***Valid input*** Select option from list ***Default value*** Reverse Proxy For a description of the deployment options please refer to Deployment.
Web server protocol Drop down list	Select the web server protocol. ***HTTP*** Standard non-encrypted HTTP site. ***HTTPS*** SSL/TLS HTTPS website ***Both*** Create e website that responds to both HTTP and HTTPS requests. Note that depending on the deployment architecture "HTTPS" and "Both" may not be available in cloud environments.
Web server domain name	The public address of the web server you want to add a proxy for.

Input field	***Valid input*** A fully qualified domain name ***Input example*** www.mydomain.com ***Default value*** `none`
Listen IP Select combo	The IP address the virtual host is bound to. Click **Edit list** to change the IP address configuration. ***Valid input*** One or more IP addresses in the select list to the left. ***Default value*** The IP address(es) configured when creating the website proxy.
HTTP listen port Input field	The port number the virtual HTTP host is listening to. ***Valid input*** A valid TCP/IP Port number ***Input example*** `80` ***Default value*** The port number set for the server when created.
HTTPS listen port Input field	The port number the virtual HTTPS host is listening to. ***Valid input*** A valid TCP/IP Port number ***Input example*** `443` ***Default value*** The port number set for the server when created.

1.2.2. Real web servers

Real server protocol Drop down list	HTTP or HTTPS ***Valid input*** Options from the drop down list `HTTP` or `HTTPS` HTTPS is only available if website virtual host is SSL-enabled.

	Default value The protocol initially set when the website proxy was created.
Validate real servers and enable health checking Check box	When enabled Web Security Manager will 1) and 2) 1. Verify that the real servers entered respond to requests 2. Enable health checking with an initial simple configuration If one or more of the real servers are not reachable Web Security Manager will return an error. To disable real server validation uncheck this option. Default: `<disabled>`

Real server IP
Input field
	Hostname or IP address of the web-server(s) Web Security Manager is proxying requests for.
	Valid input
	Fully qualified hostname (FQDN) or IP address.
	Input example
	`web1.mycompany.com`
	`10.10.10.10`
	Default value
	`<none>`

Port Input	The port number the real server is listening to. ***Valid input*** A valid TCP/IP Port number ***Default value*** `80`
Role Drop down list	Define the servers role in the load balancing set. ***Active*** The server is operative and accepts requests. ***Backup*** The server is operative but should only be sent requests if none of the other servers in the load balancing set are available. ***Down*** The server is nor operative and will not respond to requests.

1.2.3. Default Proxy

When enabled the proxy will be used as the default host for requests for the IP address the proxy is configured to listen to. The default proxy will respond to all requests for virtual hosts that are

not configured as primary host name or as a virtual host for other proxies listening to the same IP address. This way it is possible to configure a single proxy that serves requests for several host-names that are served by the same backend web server without having to add all the virtual host names in WSM.

1.2.4. Initial operating mode

Set the initial operating mode for the website proxy.

Operating modes are sets of configurations defining what violations to block and what violations to just log. Two configurable and one non-configurable presets are available.

Protect

The Protect mode preset by default blocks and logs all violations according to the access policy.

Detect

In the default Detect mode preset only logging occurs and no blocking protection is activated. Blocking protection that would occur in Protect is logged and available for review in the deny log. Operating in the default Detect preset is comparable to an intrusion detection system - it detects and logs activities but does not protect or prevent policy violations.'

Pass

In Pass mode all requests are passed through the website proxy. No requests are blocked and no logging is performed. As no filters are active in Pass mode this mode is not configurable.

By default Detect mode is selected.

Note

Initial operating mode selection is only available in WAF licenses. For load balancer licenses the operating mode is Pass.

1.2.5. Removing a proxy

In the website overview, click on the trashcan symbol shown to the right of the website proxy you want to remove.

1.3. Global

Global HTTP settings that affect all websites.

1.3.1. Source based blocking

When deployed behind a layer 7 proxy (for instance a load balancer) the original client source IP is lost because the proxy (or proxies) that the request passes through before reaching WSM ter-minates the request and reconnects to WSM on behalf of the client. Therefore the connection source IP, as seen from WSM, will be the source IP of the proxy. Since all requests will appear as coming from the proxy before WSM, source IP based blocking controls would apply to the proxy and consequently block all traffic.

To be able to use source IP based blocking controls WSM can be configured to enforce the controls at the application layer instead of at the network layer. When an IP is blocked WSM will not send a response to a request. This is the equivalent of "silent drop" at the network level.

The controls are enabled/disabled by a global master switch. When enabled, blocking is either enabled for single websites or can be forced for all websites with a global switch.

Application layer source IP blocking enabled Check box	Master switch to enable/disable the application layer blocking feature. When enabled it will be enforced for websites that have it enabled. Default: `<disabled>`
Enable application layer source blocking for all websites Check box	Turn application layer source blocking on for all websites regardless of single website configuration. This is useful when all websites are being repeatedly and systematically attacked from one or more source IPs. Default: `<disabled>`
Enable application layer source blocking for all websites Input field	The global server ID. An empty string will completely remove the server ID (prevent sending the Server header). ***Valid input*** alphanumeric, space, dash, slash, underscore, period and parentheses ***Default value*** `<empty>`

1.3.2. Server ID

The server ID is the name of the server that will be sent in the response header "Server", also called the server banner. It is considered good practise to hide, mask or alter the server banner.

The server id can be set for each website proxy or globally for all websites.

Enforce server id for all website proxies Check box	Enable / disable to enforce the global server id for all websites. Default: `<disabled>`
Server ID Input field	The global server ID. An empty string will completely remove the server ID (prevent sending the Server header). ***Valid input*** alphanumeric, space, dash, slash, underscore, period and parentheses ***Default value*** `<empty>`

1.3.3. HTTP request throttling

HTTP request throttling tracks client request rate across all websites and enforces configured limits.

When WSM is configured to be aware of receiving the request from proxies at the application layer (Trusted proxy), these controls are transparently enforced at the application layer.

Enable client HTTP request throttling Check box	Enable / disable HTTP request throttling for all websites. Default: `<disabled>`

1.3.3.1. Max HTTP request rate throttling zones

Client request rate is tracked across website proxies using four global databases, throttling zones. To account for different usage patterns throttling limits are defined separately for each global throttling zone.

Zone T1, T2, T3 and T4 Input field	Each zone defines a maximum request rate in seconds. If for instance a website proxy is assigned Zone T3 client requests to that site will be throttled down to a maximum of 5 req/sec per IP. As the aim of throttling client requests typically is to prevent clients from consuming excessive system resources request throttling cannot is enabled on a global basis and client requests are tracked and throttled across all websites. This means that in the above example client requests are tracked across all website proxies and that the Zone T3 limits enforced for other websites using Zone T3. By default Zone T1 is selected for all sites. ***Unit*** Requests / second ***Valid input*** Number in the range 0 - 1000000 ***Default value*** `T1 = 50, T2 = 10, T3 = 5, T4 = 1`

1.3.3.2. Web site settings

Maximum burst rate Input field	How many requests the client is allowed to exceed the allowed request rate with. If for instance the maximum burst rate is set to 20 and the request rate is limited to 5 request per second then the client may issue 20 requests for one second but will then have to wait 4 seconds until the rate is balanced. When a client for instance loads an html page it typically results in a lot of sub-requests for graphic elements, style sheets, javascript, etc. Setting a reasonable burst rate will allow for fast page loads when the request rate is limited. ***Unit*** requests / second

	Valid input Number in the range 0 - 1000000 ***Default value*** `20`
Throttling action Drop down list	How to handle clients exceeding limits. ***Delay*** Slow down the client by delaying responses Default selection ***Error 503*** Return HTTP error 503
Throttling zone Drop down list	Client request rate is tracked across website proxies using four global databases, throttling zones. To account for different usage patterns throttling limits are defined separately for each global throttling zone.
Precedence Drop down list	The global website settings can either be default settings when a website is created or enforced settings for all websites. ***Default web site settings*** When creating a website the settings will default to the global website settings. ***Enforced for all websites*** The global website settings will be enforced for all websites overruling settings defined in website proxies.

1.3.4. HTTP connection limiting

HTTP connection limiting tracks client connection concurrency across all websites and enforces configured limits.

When WSM is configured to be aware of receiving the request from proxies at the application layer (Trusted proxy), these controls are transparently enforced at the application layer.

Enable client HTTP connection limiting Check box	Enable / disable connection limiting for all websites. Default: `<disabled>`

1.3.4.1. Max HTTP connections limiting zones

Client connection concurrency is tracked across website proxies using four global databases, connection limiting zones. To account for different usage patterns connection limits are defined separately for each global limiting zone.

Zone L1, L2, L3 and L4	Each zone defines maximum allowed concurrent connections per client IP.

Input field	If for instance a website proxy is assigned Zone L1 client IPs are not allowed to establish more than 4 concurrent connections to the website proxy. However as client connections are tracked across all website proxies the limits will also be tracked and enforced for other websites using Zone L1. Browsers will typically establish up to four concurrent connections when loading a web page, however many clients may access the website from behind the same gateway and this may result in a much higher concurrency from that IP. By default Zone L1 is selected for all sites. ***Unit*** Requests / second ***Valid input*** Number in the range 0 - 1000000 ***Default value*** `L1 = 100, L2 = 20, L3 = 10, L4 = 4`

1.3.4.2. Web site settings

HTTP connection throttling zone Drop down list	Client request rate is tracked across website proxies using four global databases, throttling zones. To account for different usage patterns throttling limits are defined separately for each global throttling zone.
Precedence Drop down list	The global website settings can either be default settings when a website is created or enforced settings for all websites. ***Default web site settings*** When creating a website the settings will default to the global website settings. ***Enforced for all websites*** The global website settings will be enforced for all websites overruling settings defined in website proxies.

1.3.5. SSL

SSL operations consume extra CPU resources. The most CPU-intensive operation is the SSL handshake.

There are two ways to minimize the number of these operations per client:

- Enabling keepalive connections to send several requests via one connection (this is done for single websites in the ADC : Acceleration page
- Reusing SSL session parameters to avoid SSL handshakes for parallel and subsequent connections

This section applies to optimizing SSL by configuring SSL session timeouts and the SSL session cache.

SSL Session Timeout Input field	When to time out sessions stored in the session cache. Sessions are stored in the SSL session cache shared between worker processes and configured by the ssl_session_cache directive. 1 megabyte of cache contains about 4000 sessions. The default cache timeout is 5 minutes. This timeout can be increased using the ssl_session_timeout directive. Below is a sample configuration optimized for a multi-core system with 10 megabyte shared session cache: ***Unit*** Minutes ***Valid input*** Number in the range 1 - 60 ***Default value*** `10`
SSL Session Timeout Drop down list	SSL session cache size in number of SSL sessions.
Name Based Virtual Hosts (SNI) Drop down list	Enable / Disable Server Name Indication. Allow several HTTPS sites using the same IP address. If enabled Web Security Manager will allow binding an HTTPS virtual host to an IP address that is already in use by another HTTPS host. Clients supporting TLS SNI (Server Name Indication) will include the requested hostname in the first message of the SSL handshake (connection setup). This allows the server to determine the correct named virtual host for the request and set the connection up accordingly using the correct vhost SSL certificate from the start. Clients not supporting SNI will not include the requested hostname and will be served the certificate from the first vhost using the shared IP. The most common browsers support of SNI is: • Mozilla Firefox 2.0 or later • Opera 8.0 or later (with TLS 1.1 enabled) • Internet Explorer 7.0 or later (not XP) • Google Chrome • Safari 3.2.1 on Mac OS X 10.5.6 Since there is still a lot of XP based IE users out there it is not recommended to rely on SNI if broad SSL support is required. Create some more virtual IP addresses instead (cluster or virtual IPs. Default `<disabled>`.

1.3.6. HTTP global request limit

Sets the maximum upload / request size that is allowed across all websites.

Maximum configurable value is 1073741824 bytes (1 GB).

1.3.7. HTTP error log level

Sets the log level for proxy core engine error logging to the System : Logs : Proxy log.

1.3.8. HTTP global access logging

Enable/Disable debug access logging.

When enabled every website configured on an appliance will keep an access log that gets rotated on a 3 day basis. These logs will be available in the wsm/log directory prefixed with "debug". When normal access logging is enabled for a website WSM will not log to the debug file.

2. Network

Web Security Manager can block hostile IP addresses at the network level. Addresses can be learned and automatically blocked in four different ways.

1. *DoS Mitigation*

   If DoS Mitigation is enabled source IPs exceeding configurable request limits are automatically blocked for a configurable number of seconds (i.e. 86400 - 24 hours).

2. *Attack source auto blocking*

   If Attack source auto blocking is enabled source IPs are tracked across all website deny logs. If a number requests above a certain risk level are recorded within a certain time span the source IP is automatically blocked for a configurable number of seconds.

3. Immediate source blocking.

   Each website can be configured to immediately block a source IP if a log event above a certain risk level is recorded.

4. Manual entry

   IP addresses can be added manually to the list of blocked source IPs.

Only traffic to inbound interfaces is blocked. Management interfaces are not blocked unless the management role has been bound to an interface which is also responding to inbound requests - typically the interface facing the Internet.

Blocking a source IP does not keep a determined attacker from accessing your website. Positive filtering at the application level, which is the core functionality of Web Security Manager is much better at stopping unauthorized intrusion attempts. It does however make it more difficult, especially if immediate source blocking is enabled as this will force the attacker to change IP every time he triggers an attack signature.

Note

Settings like blacklisting and DoS mitigation controls that work on the client IP are only effective when WSM is terminating the original request as received from the Internet. When WSM is deployed behind a Layer 7 device that hides the client IP at the network layer these settings should not be enabled.

2.1. Blacklisted Source IPs

The table shows which source IPs are currently blocked.

Source IP	Source IP
Violation	The reason for / type of blocking. Can be: ***DoS*** The source IP has triggered the DoS mitigation by issuing too many requests within a too short time span. ***Policy*** The source IP has either triggered the general attack source auto blocking or a website specific block-IP policy.

	Permanent The source IP has been added to the list manually.
Del Button	Remove IP from the list.

2.2. Network blocking bypass

The table shows IP addresses which are allowed to bypass network protection like blacklisting and DoS mitigation controls.

Trusted Client Source IP	The IP address which will bypass network controls.
In packets	Number of incoming packets from the source IP
In bytes	Number of incoming bytes from the source IP
Out packets	Number of outgoing packets to the source IP
Out bytes	Number of outgoing bytes to the source IP

2.2.1. Allowing an IP address to bypass network controls

The network blocking bypass white list is compiled of

1. the website trusted client lists,
2. the website trusted proxies,
3. the default gateway.

Website trusted client lists

IP addresses are added in **Services** → **Websites**+**Policy** → **Website global policy**+**Trusted clients** and network blocking bypass for trusted clients has to be checked in **Services** → **Websites**+**Policy** → **Website global policy**+**IP pass through**. In addition network blocking bypass has to be enabled in general (below).

Website trusted proxies

Trusted proxies are added in **Services** → **Websites**+**ADC** → **Virtual host**+**Trusted Proxy**.

The default gateway

This is enabled by default.

Note that this feature is only available on WAF licenses.

Allow website Trusted Client IPs to bypass network protection Check box	Enable / disable network blocking bypass for trusted clients. Default: `<disabled>`
Allow trusted proxy IPs to bypass network protection	Enable / disable network blocking bypass for trusted proxies.

Check box	Default: `<disabled>`
Allow gateway IP to bypass network protection Check box	Enable / disable network blocking bypass for the default gateway. Note that this will not allow requests passing through the default gateway but only requests with the default gateway as source. Default: `<enabled>`

2.3. DoS mitigation

When enabled the DoS mitigation system tracks source IP connections to inbound interfaces. If an IP exceeds the configurable limits it is added to the list of blocked IPs and further connection attempts are silently dropped at the network level.

Enable DoS mitigation Check box	Enable / disable DoS mitigation. Default: `<disabled>`
Max src conn rate Two input fields: number and seconds.	Limit the rate of new connections to a certain amount per time interval. ***Valid input*** Both fields take an integer as valid input. ***Input example*** 50 / 5 - 50 connections in 5 seconds ***Default value*** `<60 / 10>`
Blacklist IPs for	How long time IPs should be blacklisted in seconds. ***Valid input*** An integer ***Input example*** `<36000>` - 10 hours ***Default value*** `<86400>` - 24 hours IPs are automatically removed from the list when the blacklist period has ended.

2.4. Attack source Auto blocking

Attack source auto blocking tracks denied source IPs at the application level and blocks an IP at the network level if they reach configurable limits.

Enable Attack Source Auto Blocking Check box	Enable / disable Enable Attack Source Auto Blocking. Default: `<disabled>`

Attack threshold Input field	Sets the maximum number of denied requests across all websites within a certain time frame (below). Only websites with source tracking enabled contribute to the attack threshold number and for each website a risk threshold is configured above which denied requests are added to this global counter. ***Valid input*** Any integer ***Default value*** `<5>`
Time threshold Input field	Sets the time frame within `attack threshold` (above) is accepted. ***Valid input*** Any integer ***Default value*** `<86400>`
Blacklist IPs for	How long time IPs trigging the Attack source Auto blocking should be blacklisted in seconds. ***Valid input*** An integer ***Input example*** `<86400>` - 24 hours ***Default value*** `<604800>` - 1 week IPs are automatically removed from the list when the blacklist period has ended.

2.5. Network routing

In some network deployments it is desirable to have Web Security Manager perform routing functions by forwarding IP packets not destined for its own IP addresses and to allow these packets to pass between its interfaces. Enabling IP forwarding is a necessary prerequisite when websites are deployed in `routing proxy` mode.

A segmentation matrix allows for configuring policy rules for forwarding IP packets between network interfaces.

Enable IP forwarding Check box	Enable / disable IP forwarding. IP forwarding is required when websites are deployed in routing proxy mode. Default: `<disabled>`

Enforce network segmentation when routing Check box	Enable / disable network segmentation. When enabled network segmentation rules as specified in the segmentation policy matrix are enforced. Segmentation has no effect unless IP forwarding is enabled. Default: `<enabled>`
Network segmentation	The network segmentation matrix defines policy rules for traffic to travel across the Web Security Manager network interfaces. Policy rules are defined as *allow from* interfaces in the leftmost column *to* interfaces in the upper horizontal row. The segmentation matrix only shows physical interfaces. Cluster (VRRP) interfaces and virtual IP addresses inherit the policy rules applying to the physical interfaces to which they are bound. ***Example*** If a system has the interfaces em0, em1 and em2, to allow packets to travel from em0 to em1 check the cell em0,em1. ***Default value*** Traffic is not allowed to travel across interfaces.

Chapter 4

Application Delivery Controller (ADC)

1. Virtual host

The virtual host is the website proxy that is accepting requests on behalf of the web servers serving the website the ADC is proxying requests for.

1.1. Deployment

Web Security Manager is designed to easily fit into complex data centers without sacrificing the inherent protection advantages of the reverse proxy deployment mode. This is achieved through the deployment options *Reverse Proxy* and *Routing Proxy*. Both deployment options offer the full set of WAF features including inspection and rewriting/blocking of outgoing server responses, accelerating, caching and compression.

The two deployment options can be used in combination on the same appliance as the deployment option applies to single websites. In other words the same appliance can at the same time serve websites deployed in Routing Proxy and Reverse Proxy mode.

1.1.1. Reverse Proxy

In reverse proxy the appliance terminates all traffic destined to the website it protects. For HTTP(S) traffic requests are validated and forwarded to the backend web server on behalf of the client.

A number of IP addresses are assigned to the appliance. The number of IP addresses required depends on how many SSL websites are served and on which type of SSL certificates are used. As a rule of thumb one unique IP address is required for each certificate deployed on the appliance.

To direct traffic through the reverse proxy either NAT rules or DNS has to be altered to point to the appliance. If it is required that traffic to other (non-http) services can reach the web server from the Internet and separate NAT rules has to be created for the ports serving those services that bypass the appliance.

Reverse proxy completely shields the web server infrastructure and allows for inspection of both client requests and server responses as well as rewriting/insertion of cryptographic tokens allowing for protection against session hijacking, cross site request forgery and similar attacks.

Reverse proxy is easy to implement but a number of extra IP addresses are required and for more complex data centers it may also be undesirable because of the number of changes that are required to the network firewall NAT rules.

1.1.2. Routing Proxy

Routing proxy deployment has all the advantages of reverse proxy both in terms of protection, acceleration, caching and compression. In fact, there are no features that available for reverse proxy that are not also available in routing proxy deployment.

The major difference is that routing proxy deployment does not require more than one IP address for each of the Web Security Manager appliances network interfaces and the only change necessary on the network firewall (or router) is to configure it to route traffic to the protected web servers through Web Security Manager. Web traffic to the protected servers will be picked up and validated while traffic to other protocols like SSH, SMTP and FTP is routed through to the backend web servers.

The ability to route traffic to other services also means that it is only the HTTP services on the backend web servers that are protected by the appliance but small footprint in terms of IP addresses and network firewall policy rules makes it an attractive deployment option for complex data centers.

1.2. Virtual web server

Web server Read only	Protocol and Fully qualified domain name (FQDN) for the website the proxy is configured for.
Website status Drop down list	Controls if the website is served by the Web Security Manager node. ***Enabled*** The Web Security Manager node serves requests to the website. ***Disabled*** Requests to the website are served with a default 404 not found error message.
Proxy name Input field	The name of the website proxy when listed in overview tables and reports. ***Valid input*** An alphanumeric string. ***Default value*** The first part of the virtual host address - ie. if the host address is intranet.domain.tld, the proxy name defaults to "intranet".
Deployment Drop down list	The proxy deployment mode. ***Valid input*** Select deployment mode from list ***Default value*** Reverse Proxy For a description of the deployment options please refer to Deployment.
Listen IP Select combo	The IP address the virtual host is bound to. Click **Edit list** to change the IP address configuration. ***Valid input*** One or more IP addresses in the select list to the left. ***Default value*** The IP address(es) configured when creating the website proxy.
HTTP listen port Input field	The port number the virtual HTTP host is listening to. ***Valid input*** A valid TCP/IP Port number ***Input example*** `80`

	Default value The port number set for the server when created.
HTTPS listen port Input field	The port number the virtual HTTPS host is listening to. ***Valid input*** A valid TCP/IP Port number ***Input example*** 443 ***Default value*** The port number set for the server when created.
Update certificates Button	Click to update or add SSL server certificate. See Section 1.3, "SSL Certificate" for details.

1.3. SSL Certificate

In the SSL certificate section the current SSL certificate in use is displayed. To upload a new certificate click the **Manage certificates** button.

The SSL section is only shown for SSL enabled website proxies.

1.3.1. Importing the SSL certificate

To import a certificate go to **Web Firewall** → **Websites** → **Settings** → **Servers**.

In the section **Virtual web server** select **Update certificates**.

Depending on the format of the certificate select the appropriate action in the bullet list.

1.3.1.1. Importing the PKCS12 format

If the certicifate is in the PKCS12 format follow the guidelines below:

1. Enter the path to the certificate file in the **PKCS12 file** input field.
2. Enter Passphrase in the **Passphrase** input field.
3. Click **Save settings** in the lower button pane.

1.3.1.2. Importing the PEM format

If the certificate is in the PEM format follow the guidelines below:

1. Open the .PEM file in a text-editor. Copy the public certificate section of the certificate.

   The public key/certificate is the section of the certificate file between (and including) the certificate start and end tags. Example:

   ```
   -----BEGIN CERTIFICATE-----
    Certificate characters
   -----END CERTIFICATE-----
   ```

2. Select **Import SSL certificate** In the Web Security Manager management interface

   Paste the SSL public key/certificate into the **SSL-certificate** field.

3. Now copy the (SSL) private key section of the certificate. The (SSL) private key is the section of the certificate file between (and including) the private key start and end tags. Example:

```
-----BEGIN RSA PRIVATE KEY-----
 Private key characters
-----END RSA PRIVATE KEY-----
```

4. Enter the passphrase for the private key in the **passphrase field** (if the original private key was encrypted).
5. If a certificate authority chain is provided with your certificate enter the entire list of certificates (more than one certificate may be provided) in the SSL authority certificate(s) chain field

1.3.2. Exporting certificate from web server

When creating a proxy for an existing HTTPS web server you need to move the SSL-certificate from the web server to Web Security Manager. This is done by exporting the SSL-certificate from the web server and importing it into Web Security Manager.

Web Security Manager supports importing of PKCS12 and PEM encoded server certificates.

To export a certificate from the web server please refer to the vendors guidelines:

Microsoft

Microsoft guidelines can be found on these addresses:

IIS 5.0

How to back up a server certificate in Internet Information Services 5.0

IIS 6.0

Exporting a Client Certificate for One-to-One Mapping

Export the certificate to a .PFX file (default) which is PKCS12 encoded.

Apache

For web servers running Apache:

1. Obtain the SSL-certificate file from the web servers file system. By default the file is PEM-encoded.

1.4. Virtual host aliases

To configure Web Security Manager to handle requests for host aliases to the main configured domain name (e.g. www.mydomain.com) add a list of aliases in this section.

For instance if the web system answering requests to `www.mydomain.com` also serves requests to `mydomain.com`, `www.mydomain.net` and `mydomain.net` with the same content of `www.mydomain.com`, the alias domain names, when added in this section, will be handled and validated by Web Security Manager as aliases to the "main" virtual host.

Virtual host aliases Input area	A list of host names. ***Valid input*** Hostnames separated by new-line.

	Wildcard character * can be used to substitute the server name and sub domains.
	Input example
	`mydomain.com`
	`www.mydomain.net`
	`*.mydomain.net - matches www.mydomain.net, www.intra.mydomain.net, a.b.c.d.e.f.mydomain.net...`
	`10.10.10.20`
	Default value
	`<none>`

When WSM is deployed as a proxy requests for virtual host aliases are filtered and forwarded without modification to the host header.

1.4.1. Wildcards

The wildcard character `*` can be used to match the server name part of the domain name (e.g. www). If for instance the the domain names `www. domain.net`, `www2.domain.net`, `www3.domain.net` and `webserver.domain.net` all point to the same server with the same server the wildcard expression `*.domain.net` can be used to match all HTTP requests pointing to domain.net - provided, of course, that the DNS records of the respective hosts all point to Web Security Manager.

1.4.2. Default Proxy

When enabled the proxy will be used as the default host for requests for the IP address the proxy is configured to listen to. The default proxy will respond to all requests for virtual hosts that are not configured as primary host name or as a virtual host for other proxies listening to the same IP address. This way it is possible to configure a single proxy that serves requests for several hostnames that are served by the same backend web server without having to add all the virtual host names in WSM.

1.5. Timeouts

Client READ header timeout Input field	Max time to wait for the client request header. ***Unit*** Seconds ***Valid input*** Number in range 2 - 7200 ***Default value*** `60`
Client READ body timeout Input field	Max time to wait for the client request body. ***Unit*** Seconds

	Valid input Number in range 2 - 7200 ***Default value*** `60`
Client SEND timeout input field	Max time to wait for a client send to complete. ***Unit*** Seconds ***Valid input*** Number in range 2 - 7200 ***Default value*** `60`

1.6. HTTP Request and Connection Throttling

1.6.1. HTTP request throttling

HTTP request throttling status Info	Displays the global HTTP throttling status.
Maximum burst rate Input field	How many requests the client is allowed to exceed the allowed request rate with. If for instance the maximum burst rate is set to 20 and the request rate is limited to 5 request per second then the client may issue 20 requests for one second but will then have to wait 4 seconds until the rate is balanced. When a client for instance loads an html page it typically results in a lot of sub-requests for graphic elements, style sheets, javascript, etc. Setting a reasonable burst rate will allow for fast page loads when the request rate is limited. ***Unit*** requests / second ***Valid input*** Number in the range 0 - 1000000 ***Default value*** `20`
Throttling action Drop down list	How to handle clients exceeding limits. ***Delay*** Slow down the client by delaying responses

	Default selection ***Error 503*** Return HTTP error 503
Throttling zone Drop down list	Client request rate is tracked across website proxies using four global databases, throttling zones. To account for different usage patterns throttling limits are defined separately for each global throttling zone.

1.6.2. HTTP connection throttling

HTTP connection throttling status Info	Displays the global HTTP connection throttling status.
HTTP connection throttling zone Drop down list	Client request rate is tracked across website proxies using four global databases, throttling zones. To account for different usage patterns throttling limits are defined separately for each global throttling zone.

1.7. Client Source IP

HTTP requests often pass through one or more proxy servers before reaching the endpoint Web server. Examples include Web gateways in the client network, content delivery networks (CDN), caching servers, SSL accelerators, layer 7 load balancers and Web application firewalls. Each time the request passes through a proxy server, the source IP of the request is changed to the IP address of the proxy server. This means that endpoint Web servers cannot rely on the source IP from the network connection (socket) to be the IP address of the original request. To account for this it has become a de-facto standard that proxy servers insert the client source IP in a request header named X-Forwarded-For. This is the default behavior of WSM.

Another option is to configure WSM as a **Transparent Proxy**, which re-inserts the client source IP before forwarding the request to the backend Web environment.

1.7.1. X-Forwarded-For

De-facto standard for forwarding client source IP from layer 7 proxies. Header is always inserted.

X-Forwarded-For	• Client source IP will be inserted in X-Forwarded-For header • Client source IP will be present in the header even if the request has passed through other proxy servers • Standards compliant approach • Better performance because connections to backend Web servers can be kept alive • May require modifications to backend applications to read the X-Forwarded-For header The X-Forwarded header is always inserted.

The X-Forwarded-For (XFF) HTTP header field is a de facto standard for identifying the originating IP address of a client connecting to a Web server through an HTTP proxy or load balancer. This

is an HTTP request header that was introduced by the Squid caching proxy server's developers. An effort has been started at IETF for standardizing the Forwarded HTTP header.

When the request passes through multiple proxy servers, each server will add the respective source IP to the X-Forwarded-For header. The X-Forwarded-For header thus contains a list of IP addresses the request has passed through. The leftmost IP address is the original client IP. The rightmost IP address is the last proxy in the chain, and the source IP of the request is the address of the last proxy.

This allows the endpoint Web server to extract and log the original client IP address from the XFF header for applications that need this data rather than the IP address of the last proxy in the chain from which the endpoint Web server received the request.

As Web Security Manager is a proxy-based device, it terminates requests from clients and makes requests to the backend Webserver on behalf of the client. To make the original source IP available to the backend Web application, Web Security Manager forwards the source IP address to the backend server in the X- Forwarded-For header.

1.7.2. Other X-headers

In addition to the X-Forwarded-For header Layer 7 proxies also often insert X-headers with information about which protocol and port the request was received on. These headers are named

X-Forwarded-Proto

Contains the protocol the request was received on - HTTP or HTTPS

X-Forwarded-Port

Contains the port the request was received on

Note that contrary to the X-Forwarded-For header these headers are not lists. This means that the information in them will be overwritten by WSM if it receives the request from a proxy that has already inserted them.

If If requests through WSM passes through further layer 7 proxies on their way to the backend servers, to keep the value of these headers as received by WSM intact, it is necessary to copy the value of those headers into reserved custom headers like X-WSM-Forwarded-Proto. This can be accomplished by inserting request headers. See Section 2.4, “Health Checking” for details.

1.7.3. Trusted proxy

Since the X-headers are part of the client request WSM receives, and as such can be manipulated by the client, by definition they cannot be trusted. However, if the proxy that receives the request from the client, for instance a load balancer in front of WSM, follows the standard of appending the source IP it receives the request from to the X-Forwarded-For header along with overwriting values in X-Forwarded-Proto and X-Forwarded-Port headers with protocol and port information, the information inserted by that proxy **can** be trusted since the client does not control that information. Such a proxies, defined by one or more IPs, are referred to as **Trusted Proxies** throughout this manual.

When WSM receives a request from a trusted proxy it will extract the client source IP from the X-Forwarded-For header and use that IP address in place of the actual request (socket) IP for both HTTP Throttling and, if enabled, IP based blocking. For the IP based blocking controls the X-Forwarded-For IP will transparently replace the socket IP and the network blocking controls will work without other modification to the policy.

Use trusted proxy - extract client source IP from X-Forwarded-For header Check box	Enable Trusted Proxy functionality. When enabled, if request is received from a proxy in the list of trusted proxies (see below): • WSM will extract the client source IP from the X-Forwarded-For header • Use the extracted client source IP for HTTP Throttling (when enabled) • Use the extracted client source IP for IP whitelisting controls • Transparently enforce the source IP based request blocking at layer 7 based on the extracted IP instead of at the network level Default: `<disabled>`
Reset X-Forwarded-For header to last untrusted source in list Check box	Following the proposed standard, the X-Forwarded-For header may contain a comma separated list of IP addresses. Depending on the configuration options in the endpoint Web server and application, it may be logged as such which may not be desirable. To have WSM always forward the X-Forwarded-For a single IP address, enable this option. Default: `<disabled>`
Forward X-Forwarded-For and X-Forwarded-Proto headers from trusted proxy unaltered Check box	Leave X-Headers untouched from trusted proxy. If Web Security Manager is deployed behind another reverse proxy, by default, Web Security Manager will insert the source IP from that proxy in the X-Forwarded-For header sent to the backend web server. If the X-Forwarded-For header is already present the source IP will be appended to the header. While this behaviour conforms to standards it is not always desirable. It is therefore possible to configure trusted proxies from which Web Security Manager will simply forwarded the X-Forwarded-For header as it is received from the trusted proxy without modifying it. Default: `<disabled>`
List of trusted proxies Input field	List of trusted source IPs from which X-Forwarded-For header will be forwarded unmodified to the backend web server. ***Valid input*** IP addresses with net mask (IP/mask) in CIDR notation separated by newline ***Input example*** `192.168.0.8/32` - the IP address 192.168.0.8 `192.168.0.0/24` - IP addresses 192.168.0.0 - 255 `192.168.0.8/29` - IP addresses 192.168.0.8-15 ***Default value*** `<none>`

1.7.4. Transparent Proxy

Play tricks at the network level to inject the client IP as the source of the request to the backend server. Requires modification of default gateway at the backend server to make the response go back through WSM.

Transparent Proxy	• WSM will insert the client source IP as the source IP of the request it forwards to the backend Web server • Backend Web servers need to be configured to use WSM as the default gateway to ensure inspection of return Web traffic and a response to the client request • IP-Forwarding (routing) needs to be enabled in WSM • Performance drawbacks because connections from WSM to backend Web servers cannot be kept alive and because all traffic, including non-HTTP, from Web servers have to pass through WSM • Because WSM needs to be the default gateway Transparent Proxy is only recommended for smaller deployments where WSM is not deployed in combination with a load balancer Transparent proxy needs to be enabled and configured. See below.

Transparent Proxy is a configuration option that can be applied to both Routing Proxy and Reverse Proxy deployment modes.

When enabled WSM will preserve the client source IP by inserting it in the request to the backend web server. In practice it is spoofing the client source IP and for that reason this feature is sometimes also called Client Impersonation.

While transparent proxy does the job in terms of preserving the client source IP in the HTTP requests the backend web server receives it has a few drawbacks in terms of performance and availability risk. Because the original client IP is inserted as the source IP of the connection that is made to the backend a new connection has to be made for every request in order not to re-use connections from other client IPs. This will impact performance negatively. While WSM deployed as either Reverse or Routing is very easy to bypass if both nodes in a cluster fails because all that needs to be done is to change a NAT rule or a static route is gets a little more complicated when it is proxying transparently because the backend web servers are configured to use the WSM cluster as a gateway. This means that each backend web server has to be reconfigured to restore availability in the unlikely event that both WSM nodes in a cluster fails.

Note

Because the original client source IP appears to be the source IP for the backend web server receiving the request it needs to have WSM configured as the default gateway in order for the web server response to come back through WSM. This also means that it will be complicated and error prone at best to implement use Transparent Proxy in high performance deployments where WSM is deployed in combination with a load balancer.

1.7.4.1. Configuring Transparent Proxy

To minimize the availability impact on the web properties the configuration of Transparent Proxy should be performed in the following order:

Web Security Manager

- If WSM is deployed as a cluster, create a cluster interface with an VIP that is reachable by the backend web servers in System : Clustering.
- Enable IP Forwarding in Services : Network > Network routing.
- Make sure that routing does not violate network segmentation settings by editing the segmentation table in Services : Network > Network routing. By default routing cannot traverse network interfaces.

Backend web servers

- Configure backend web servers to use WSM as default gateway.

1.8. Redirects

Tell the client to get the requested resource somewhere else.

The Redirect feature is used to instruct clients to make a new request with a different URL. It is often used to redirect HTTP requests for resources requiring encryption to corresponding pages on an SSL encrypted connection - HTTPS.

1.8.1. Match types

Web Security Manager allows for either `prefix`, `regex` or `vhost regex` based matching of client requests.

Prefix

If prefix match is selected the requested URL is matched left to right beginning with a slash (/secret).

Regex

If Regex match is selected the requested URL is matched using a regular expression. Anything goes here so it is basically possible to match asp files in a specific directory and instruct the client to request a php file in another directory on another server using HTTPS instead of HTTP.

Do not select Regex match type unless you really need it. Prefix is cheaper CPU wise.

Vhost regex

The vhost regex type allows for matching on elements in the virtual host name and redirecting to a different virtual host optionally with some of the matched elements in the target url - like redirecting foo.alertlogic.com to http://www.alertlogic.com/foo or foo.alertlogic.net to http://www.alertlogic.com/net/foo.

The syntax is dependent on the match type selected.

1.8.2. Prefix match

Enable external redirects Check box	When checked Web Security Manager will redirect client requests based on redirect rules configured.
Proto Drop down list	For website proxies serving both HTTP and HTTPS select the protocol to match.

	If for instance you only want to serve a specific page using the HTTPS protocol match the corresponding HTTP page and redirect to HTTPS on the same site.
Match type Drop down list	See above.
Match Input field	The client request to match. If prefix match is selected the requested URL is matched left to right beginning with a slash (/secret). Only complete path segments are matched so prefix match type is basically matching on a "directory" basis. ***Valid input*** A case-sensitive (%-decoded) path beginning with a slash ***Input example*** `/secret` - will match requests for /secret, /secret/, /secret/secret_file1.php, etc.. Will NOT match /secret_file.php. `/` - will match requests for any resource, useful for setting up an HTTP proxy which redirects all requests to the same "location" on an HTTPS proxied website. ***Default value*** `<none>`
Redirect externally to Input field	The new URL path the client is redirected to. If prefix match is selected the new URL path corresponds to the prefix matched. If /secret is entered in the match field (above) then the part of the request following the prefix (/secret) is sent to the new URL path. ***Valid input*** An absolute URL beginning with a scheme and hostname, but a URL-path beginning with a slash may also be used, in which case the scheme and hostname of the current server will be added. ***Input example*** `https://ssl.somename.tld/moresecret` - In combination with the prefix match example above /secret requests for /secret will be re-directed to https://ssl.somenane.tld/moresecret, /secret/secret_file1.php will be redirected to https://ssl.somen-ane.tld/moresecret/secret_file1.php, etc. `https://ssl.somename.tld/` - In combination with the prefix match example / above will redirect any request to https://ssl.somename.tld. ***Default value*** `<none>`

1.8.3. Regex match

Enable external redirects Check box	When checked Web Security Manager will redirect client requests based on redirect rules configured.
Proto Drop down list	For website proxies serving both HTTP and HTTPS select the protocol to match. If for instance you only want to serve a specific page using the HTTPS protocol match the corresponding HTTP page and redirect to HTTPS on the same site.
Match type Drop down list	See above.
Match Input field	The client request to match. If Regex match is selected the requested URL is matched using a regular expression. The supplied regular expression is matched against the requested URL-path, and if it matches, the server will substitute any parenthesized matches into the redirect URL path sent in the redirect response to the client. ***Valid input*** A valid regular expression ***Input example*** `(.+)\.jsp` - will match requests for any url path ending in .jsp. The path and filename but not the extension will be in the substitute variable $1 (for instance a request for /secret/secret_java1.jsp will result in $1 containing /secret/secret_java1 making it possible to redirect to https://ssl.somename.tld$1.php resulting in the client being redirected to https://ssl.somename.tld/secret/secret_java1.php). ***Default value*** `<none>`
Redirect externally to Input field	The new URL path the client is redirected to. If Regex match is selected the parenthesized matches in $1, $2, etc. is substituted into the new URL path allowing fine grained and complex redirect rules. ***Valid input*** An absolute URL beginning with a scheme and hostname optionally with $1, $2, $3, etc. as placeholders to substitute matches into. ***Input example*** `https://ssl.somename.tld$1.php` - In combination with the regex match example (.+)\.jsp requests for any url path ending in .jsp will be redirected to https://ssl.somename.tld/ but the extension jsp will

	be php. For example /secret/secret_java1.jsp will be redirected to https://ssl.somename.tld/secret/secret_java1.php. ***Default value*** `<none>`

1.8.4. Vhost regex match

Enable external redirects Check box	When checked Web Security Manager will redirect client requests based on redirect rules configured.
Proto Drop down list	For website proxies serving both HTTP and HTTPS select the protocol to match. If for instance you only want to serve a specific page using the HTTPS protocol match the corresponding HTTP page and redirect to HTTPS on the same site.
Match type Drop down list	See above.
Match Input field	The vhost part of client request to match. If vhost regex match is selected the vhost part of the client request is matched using a regular expression. If it matches, the server will substitute any parenthesized matches into the redirect URL path sent in the redirect response to the client. ***Valid input*** A valid regular expression ***Input example*** foo\.alertlogic\.com www\.alertlogic\.(\w){1,5} ***Default value*** `<none>`
Redirect externally to Input field	The new URL path the client is redirected to. If the match expression contains parentheses the parenthesized matches are placed in the variables $c1, $c2, $c9. These variables can be used in the redirect URL to allow for fine grained and flexible redirects. ***Valid input*** An absolute URL beginning with a scheme and hostname optionally with $c1, $c2, $c9 as placeholders to substitute matches into. NOTE that placeholder variable names are different from the regex type above.

	Input example
	`http://www.alertlogic.com/foo`
	In combination with the match example foo.alertlogic.com redirects requests for the hostname foo to a corresponding subdir.
	`http://www.alertlogic.com/$c1`
	in combination with the match www\.alertlogic\.(\w{1,5} will redirect www.alertlogic.net/somepath?somequery to www.alertlogic.com/dk/somepath?somequery
	Default value
	`<none>`

1.8.5. Examples summary

The examples from the table above are summarized below. Substitute "ssl.somename.tld" with correct address.

On an HTTP proxy redirect all requests to the corresponding location on an HTTPS proxy

Match type: prefix

Match: /

Redirect externally to: https://ssl.somename.tld/

On an HTTP proxy redirect all requests for resources in /secret to /moresecret on an HTTPS proxy

Match type: prefix

Match: /secret

Redirect externally to: https://ssl.somename.tld/moresecret

On an HTTP proxy redirect all requests for .jsp to a .php script with the same name and location on an HTTPS proxy

Match type: regex

Match: (.+)\.jsp

Redirect externally to: https://ssl.somename.tld$1.php

Virtual host redirect - redirect requests to somehost.somename.cc to www.somename.tld/cc/somehost/

Match type: vhost regex

Match: (\w+)\.somename\.(\w){1,5}

Redirect externally to: http://www.somename.tld/$c2/$c1

1.9. Lower button bar

Save settings	Click **Save settings** to save settings.

2. Load balancing

2.1. Real web servers

Real server protocol Drop down list	HTTP or HTTPS ***Valid input*** Options from the drop down list `HTTP` or `HTTPS` HTTPS is only available if website virtual host is SSL-enabled. ***Default value*** The protocol initially set when the website proxy was created.

Real host
Input field
	Hostname or IP address of the web-server(s) Web Security Manager is proxying requests for.
	Valid input
	Fully qualified hostname (FQDN) or IP address.
	Input example
	`web1.mycompany.com`
	`10.10.10.10`
	Default value
	`<none>`

Port Input	The port number the real server is listening to. ***Valid input*** A valid TCP/IP Port number ***Default value*** `80`
Role Drop down list	Define the servers role in the load balancing set. ***Active*** The server is operative and accepts requests. ***Backup*** The server is operative but should only be sent requests if none of the other servers in the load balancing set are available. ***Down*** The server is nor operative and will not respond to requests.
Status Read only	The status of the real server. *enabled* or *disabled*.

X (delete) Button	Mark the real server for deletion. The server will not be deleted until the button **Save settings** in the lower button bar is activated.

2.2. Timeouts

Real server CONNECT timeout (seconds) Input field	Max time to wait for connection to the backend web server to succeed. ***Unit*** Seconds ***Valid input*** Number in range 2 - 75 ***Default value*** `10`
Real server SEND timeout Input field	Max time to wait for sending request to backend web server to complete. ***Unit*** Seconds ***Valid input*** Number in range 2 - 7200 ***Default value*** `60`
Real server READ timeout input field	Max time to wait for reading response from backend web server. ***Unit*** Seconds ***Valid input*** Number in range 2 - 7200 ***Default value*** `60`

2.3. Load balancing settings

The load balancing settings control the behaviour of the *Load Balancer*.

Web Security Manager COOKIE based session persistence Check box	When checked Web Security Manager will issue a cookie when a user first connects to the virtual host being proxied / load balanced. The cookie binds the session to the real server selected by the load balancer ensuring that the users session with the real server is not broken. This method requires that visitors have support for cookies enabled in their browser.

	Despite the name this feature works equally well with HTTPS and HTTP.
HEADER based session persistence Check box	When checked Web Security Manager will bind the user session to the real server based on a hash of a selected client request header.
Header Input field	The header to use for calculating load balancing hash when header based session persistence is selected. ***Valid input*** An HTTP-header sent by the client. ***Default value*** User-Agent
SOURCE IP based session persistence Check box	When checked Web Security Manager will bind the user session to the real server based on the visitors source IP. This method ensures that requests from visitors with cookie support disabled will be sent to the same server every time. To compensate for visitors changing IP address during the session (for instance because their requests are sent through different forward proxies) a mask is applied to the users source address (below). Applying a mask ensures that even if the users IP address changes the same server is selected.
IP mask: Drop down list	The mask to be applied to the visitors source ip address when calculating destination real server based on source hashing. The mask and resulting number of IP addresses within each "load balancing address chunk" is displayed in the drop down. ***Valid input*** Options from the drop down list ***Default value*** 255.255.240.000 (4,096 hosts)
Enable real server failover Check box	When checked Web Security Manager will attempt to redirect a request to another real server in case the real server to which the session is bound fails. Disabling real server failover only is effective when session persistence is enabled. If real server failover is disabled the user will receive an error message and the session will have to be restarted (usually by closing and restarting the browser).
Max real server failover attempts Input field	Maximum failover attempts in case a real server fails. This value defines how many times Web Security Manager should try other real servers in case a real server fails.

	Valid input Number in range 1 - (number of real servers -1) ***Input example*** `1` ***Default value*** `1`

Real server retry interval (seconds)
Input field
	Specifies for how long a failed real server should be kept in error state before trying to connect again.
	Valid input
	Number in range 1 -
	Input example
	`20`
	Default value
	`60`

2.4. Health Checking

Health checking checks the real (backend) servers for errors and availability. If a server is not responding correctly (as configured) it is disabled until it responds correctly again.

Enable real server health checking Check box	Enable / disable health checking Default: <disabled>
Request interval Input field	How often the health check daemon should check the server. ***Valid input*** Number in range 10 - 60 ***Default value*** `10`
Request timeout input field	Max time to wait for real server to respond before marking the attempt as failed. ***Valid input*** Number in range 1 - 30 ***Default value*** `2`
Error threshold Input field	Specifies how many failed health checks should be recorded before the server is disabled.

	Valid input Number in range 1 - 10 ***Default value*** `3`

Request method
drop-down list
	What method should be used for health checking.
	Valid input
	`HEAD` or `GET`
	Default value
	`HEAD`
	The HEAD method only checks the server response code. If the server returns 200 OK within the configured timeout the request is a success.
	The GET method validates the page the server returns using a checksum. If the content of the page has changed (compared to the stored checksum) the request is marked as failed.
	Note
	If the request method is GET and the content of the requested resource is changed on all servers, all servers will be disabled as they will fail the checksum check. be sure to run a checksum re-generation immediately after such an update.

Request
Input field
	The resource to request when health checking.
	Valid input
	A string starting with / specifying an application, static page, graphic or other content on the web server.
	Input example
	`/testpage.php`
	`/index.aspx?showpage=999999`
	`/graphics/1x1.gif`
	Default value
	`<none>`

Force checksum re-generation Check box	When request method GET is selected, when settings are saved Web Security Manager request the configured resource on all real servers to calculate a checksum which will be stored for further health checking. If the checksum is not the same on all servers Web Security Manager will return an error and the new settings will not be saved. The checksum is only generated when things change, like when a new Request is configured or method is changed to GET.

	There can be situations though where it is desirable to have the checksum re-generated, for instance if the content of the request page has changed. If this option is checked the checksum will be re-generated. Default: <disabled>

2.5. Insert request headers

These settings allows for inserting new headers with either a static value or with the value of different request variables.

Enable insert request headers Check box	Enable/disable request headers insert. Default: <disabled>

Header
Input field
	The name of the request header to insert
	Valid input
	Alphanumeric and -
	Input example
	`Foo-Bar`
	`Client-IP`
	Default value
	`<none>`

Value type Drop down	Specify the type of input entered in the value field. ***variable*** Specifies that the string entered in the value value field is to be interpreted as the name of a request variable, the value of which will be inserted in the request with the header name specified. ***literal*** Specifies that the string entered in the value field is to be inserted "as is". So if "foo" is entered the value "foo" will be inserted.
Value Input field	The value to insert in the new header field. Can be literal or variable. ***Valid input*** If literal selected: alphanumeric, space, dash, underscore, space and comma. If variable selected: See list below. ***Input example*** `remote_addr` - the IP address of the client ***Default value*** `<none>`

2.5.1. Request header variables

The variables below can be inserted in a header and forwarded to the backend server.

args

This variable is the GET parameters in request line, e.g. foo=123&bar=blahblah.

cookie_COOKIE

The value of the cookie COOKIE, e.g. to forward the value of the cookie SESSID enter cookie_SESSID or cookie_sessid as match is case insensitive.

hostname

Set to the hostname of the Web Security Manager node.

http_HEADER

The value of the HTTP header HEADER when converted to lowercase and with dashes converted to underscores, e.g. User-Agent = http_user_agent, Referer = http_referer...

remote_addr

The IP address of the client.

remote_port

The port of the client.

request_method

The method of request, usually GET or POST.

request_uri

The original request URI as received from the client including the args.

scheme

The HTTP scheme (i.e. http, https).

server_name

The virtual host server name of the website proxy handling the request(i.e. www.alertlogic.com).

server_port

The port of the server, to which the request arrived.

server_protocol

The protocol of the request, e.g. HTTP/1.0 or HTTP/1.1.

uri

The URI in the request without arguments, those are in the variable args.

2.6. Advanced settings

These settings specify request time out, keep alive behavior. Also web application behavior is specified here.

Enable real server keepalive	Enable/disable support for keepalive to backend web servers.

Check box	If enabled WSM will keep connections to the backend web servers open and reuse then for new requests thereby reducing the overhead of establishing the connection. Default: `<enabled>`
Add HTTP/1.1 VIA header information Check box	Enable/disable support for HTTP/1.1 VIA header sending information. Íf enabled, Web Security Manager will append the Via header in each forwarded request indicating to the backend server that the request if coming through a proxy server. Default: `<disabled>`
Proxy buffering enable Check box	Proxy buffering. By default Web Security Manager buffers the response from the backend web server in order for the web server to be able to deliver the request as fast as possible no matter how slow the connection to the client is. This ensures that server resources will not be consumed by clients on slow connections. However, some applications are known to have problems with this behaviour, Comet applications for instance. To account for such problems disable proxy buffering. Default: `<enabled>`
Upstream SSL session reuse enable Check box	SSL session reuse to backend servers. When Web Security Manager connects to a backend server over SSL, the server creates a session for that connection. This session ID is sent as a part of the backend `Server Hello` message. To make things efficient Web Security Manager can behave as a normal HTTP client (a browser) and reuse that session ID next time it connects to the backend server. Thus the time spent in verifying the certificates and negotiating the keys is saved. If the backend web server is configured to not support SSL session reuse pages will not load or not load correctly - typically stylesheets, images, javascript files, etc will not load. To account for such problems disable upstream SSL session reuse Default: `<enabled>`

2.7. Lower button panel

Save settings	Click **Save settings** to save settings.

3. Caching

3.1. Static Caching

Documents that can be cached, are locally stored by Web Security Manager. Any further requests for documents found in the cache, are automatically delivered to clients directly from Web Security Manager. Therefore, the back-end web servers can focus on delivering dynamic content with improved response times to clients, without the overhead of delivering static content like images, PDF documents, static HTML documents, style-sheets and others.

Enable static content caching Check box	If enabled, Web Security Manager will store static content from the web-server locally on Web Security Manager. This dramatically accelerates response times and off-loads the web-server. Default: `<enabled>`
Default caching action Input field	Web Security Manager can either cache all responses unless explicit no-cache instructions are set by the backend server or only cache content with response headers that indicates that the response is cacheable. ***Cache unless specifically instructed not to*** Cache all responses but honor Expires, Cache-Control: no-cache, private and no-store headers. ***Do not cache unless headers indicate content is cacheable*** Only cache responses if headers Expires, Cache-Control or Last-Modified indicates that the content is cache-able.
Inactive remove threshold Input field	Defines how long to keep data that is not requested in the cache. ***Valid input*** Number (seconds) ***Default value*** `3600`

Default caching time
Input field
	Define by response code how long to store cached responses in cache if expiration is not set by backend server.
	Valid input
	List of error codes and seconds.
	Default value
	`200 302 301 = 3600`
	`404 = 600`
	`any = 10` (any error code not specified directly).

Warning

Caching of content should only be activated where appropriate as caching, in particular of dynamic content, involves the risk of losing confidentiality to private data.

3.2. Dynamic caching

Using this option is only appropriate where the dynamically served content has the characteristics of static content. An example of such data could be news articles generated from a database.

If appropriate the effect of dynamic content caching can be dramatic. If for instance an article on a news site is requested at a rate of 100 requests/sec enabling caching with a content expiry of 20 seconds will result in 1 in every 2000 requests reaching the web server. The remaining 1999 will be served from the Web Security Manager cache.

Enable dynamic content caching Check box	If enabled, Web Security Manager will store dynamic content from the web-server locally on Web Security Manager for a configurable number of seconds. Default: `<disabled>`
Dynamic content expiry Input field	Store dynamically cached documents for the specified period. ***Valid input*** Number (seconds) ***Default value*** `60`
Caching locations	Cache response from requests matching regular expressions. Enter regular expressions matching part of or the entire path part you want to be cached. The expressions are matching from left to right. Full match is not implied but matching always start at start of line. This implies that for instance the expression /news will match any URI starting with /news. ***Valid input*** A valid regular expression ***Input example*** /news/.+\.php cache responses from php scripts served from locations in and below the directory news. ***Default value*** `none`

Warning

Caching of content should only be activated where appropriate as caching, in particular of dynamic content, involves the risk of losing confidentiality to private data.

3.3. Lower button bar

Flush DYNAMIC cache	Delete the contents of the website proxy static document cache.
Flush STATIC cache	Delete the contents of the website proxy static document cache.
Default values	Revert to default values.
Save settings	Click **Save settings** to save settings.

4. Acceleration

Web Security Manager can accelerate web server performance by caching static content and by compressing traffic to clients.

Depending on the mix of static and dynamic content caching of static documents can potentially improve performance by 300 - 500%.

Dynamic compression of transmission data potentially reduces bandwidth usage by 30 - 60% and thereby increases transfer rate by 50 - 100%.

4.1. Compression

HTTP compression reduces the transfer volume of static and dynamically generated web pages to approximately 1/3 of their original size proportionally speeds up the load time performance. This results in reduced traffic costs and in a better experience for the web site visitors.

The performance gain depends on the ratio to which content from the web server can be compressed, the size of the pages and the clients bandwidth.

4.1.1. Compression level

Set the gzip compression level. Compression level is a trade off between CPU resources and Bandwidth consumption. High compression levels saves more bandwidth but consumes more CPU and vice versa.

The default compression level is set at 3 which is moderate but fast.

4.1.2. Compress response content-types

Compression of server responses is based on the response content type.

Text Check box	If enabled, Web Security Manager will compress HTTP documents matching content-type `text/*`. Default: `<enabled>`
Images Check box	If enabled, Web Security Manager will compress HTTP documents matching content-type `image/*` . Normally compression should not be enabled for images because in most cases they are already optimized/compressed for the web in which case compressing will be a waste of processing power. Default: `<disabled>`
Application data Check box	If enabled, Web Security Manager will compress HTTP documents matching content-type `application/*`. Default: `<disabled>`

4.1.3. Exceptions

In some cases it may be necessary to specify exceptions from the compression by content type policy. Exceptions are defined using regular expressions matching the path segment of the requested URL (the URI).

Enable Compression Exception by Regular Expression Check box	If enabled Web Security Manager will match the URI using the regular expressions in the list. If there is a match compression will be disabled for the server response.
Regular expression	Enter regular expressions matching part of or the entire path part you want to be excluded from compression. Note that unlike filter policy regular expressions the expression does not have to match the entire path from start to end. For example 'a' would be a valid regular expression matching all paths containing an 'a' while '\.class' would match all paths containing '.class' ***Valid input*** A valid regular expression ***Input example*** ^/forms/ (do not compress responses from paths starting with /forms/) ^.+\.jar (do not compress responses from files with the extension ".jar") ***Default value*** `none`

4.2. TCP connection reuse

TCP connection reuse dramatically improves response times for clients that have support for keep-alive by reusing socket connections already established.

Enable keep-alive requests Check box	Enable / disable support for HTTP/1.1 keep-alive requests. If enabled, Web Security Manager will support keep-alive protocol specification as defined by HTTP/1.1 standard. Default: `<enabled>`

Max Keep-Alive requests
Input field
	Maximum number of requests on a kept alive connection.
	This setting limits the number of requests allowed per connection when Keep-Alive requests is enabled. If it is set to 0, unlimited requests will be allowed.
	Valid input
	Number in range 0 - 10000
	Input example
	`100`
	Default value
	`10`

Max Keep-Alive timeout	Max idle time on a keep alive connection.

Input field	This value defines the number of seconds Web Security Manager will wait for a subsequent request before closing the connection. ***Valid input*** Number (seconds) in range 1 - 300 ***Default value*** 5 **Note** Keep-Alive timeout sets the timeout on *idle* connections. As long as the connection is active (that is: the client is requesting content with a maximum of "Keep-Alive timeout" between each request) the Max Keep-Alive requests value determines when the connection is closed and the client has to re-establish the connection.

5. Statistics

The **Web Firewall** → **Websites** → **reports** → **Statictics**

The *Statistics section* contains various proxy specific statistics information.

5.1. Interval selection

This section is used for selection of the interval used for generating the statistics. The interval is always calculated from the current time.

Show last

Shows the statistics from current date and time and back the selected interval. Eg. 8 hours.

Show stats

Refresh the statistics based on the current selection.

5.2. Summary section

This section shows the statistics for the currently selected proxy.

Requests total	Total number of requests.
Requests/sec (avg.)	Average number of requests for the selected period.
Compression ratio	Compression ratio for the selected period. Eg. 60% means that the original data was compressed to the 60% of it's original size.
Cache hits	Percentage of responses served from the cache
Original data	Amount of original data before compression (in megabytes).
Data transferred	Amount of data transferred.
Data received	Amount of data received.
Response codes	Click **Show details** link to toggle display of web server response codes: ***Normal*** Response code: `200` Number of requests processed normally. ***Redirect*** Response codes: `300-399` Number of requests redirected. ***Access denied*** Response codes `400-499` except `404` Number of requests that was denied for some reason. ***Not found*** Response code: `404` Number of requests for unavailable content or resources.

	Internal error Response codes: `500-599` except `502` Number of requests resulting in an internal server error. ***Bad gateway*** Response code: `502` Number of requests resulting in the `bad gateway` error. ***Other errors*** Number of requests generating other error codes.
Interval Drop down	Selection of the interval used for generating the statistics. The interval is always calculated from the current time. ***8 hours*** Displays an interval of 8 hours counting backwards from the current time. ***24 hours*** Displays an interval of 24 hours counting backwards from the current time. ***Week*** Displays an interval of one week counting backwards from the current time. ***Month*** Displays an interval of one month counting backwards from the current time.
Period start	Starting date and time for the generated statistics.
Period end	Ending date and time for the generated statistics.

5.3. Compression and served from cache graph

This graph shows the compression ratio and served from cache ratio for the selected proxy and interval.

Compression ratio	Compression ratio spanning the selected interval.
Requests/sec (avg.)	Shows the served from cache ratio spanning the selected interval.

5.4. Requests total and served from cache graph

This graph shows the total number of requests and served from cache number for the selected proxy and interval.

Requests total	Shows the total requests spanning the selected interval.

Requests served from cache	Shows the served from cache requests spanning the selected interval.

5.5. Original data and data sent graph

This section shows the original data and data sent (megabytes) for the selected proxy and interval.

Original data	Shows the size of original data for the selected interval.
Data sent	Shows the size of data transferred for the selected interval.

5.6. Lower button bar

Clear stats	Clear all stat data and start from scratch.

Chapter 5

Web application firewall (WAF)

1. Policy

1.1. Validation order and scope

The Policy define a list of allowed requests and parameters to a given web system to which access is filtered by Web Security Manager.

The policy is defined by a collection of *proxy global policies* and *application specific policies*. This mix provides the ability to specify short yet fine grained access control policies:

Global policy

These are general rules which specify criteria for allowing requests on a proxy global basis. Rules are specified by extension and by specifying a grammar (using regular expressions) for valid URLs and parameters.

Global patterns include *Static content* policies, *Global URL* policies and *Global parameters* policies.

Web applications

In access policy terms a web application is defined as an URL path which takes one or more parameters as input.

The web application policy list consists of one or more URL paths each with a specific policy, a web application policy entry.

The web application policy entry is defined by its URL path and valid input for one or more of the URLs parameters are defined using either a list of allowed values, grammar (a regular expression) or a class which is a predefined regular expression.

Web application policy entries always take precedence over global rules. It is perfectly possible though to utilize a mix of global and specific rules - even for a single application.

Incoming requests are validated in the following order:

1. **Static content policy**

   If the extension and path of the requested filename matches the policy defined in Static content policy and the request has no parameters, the request is allowed.

2. **Global URL path policy**

   If the request has no parameters and one of the global URL policy patterns matches it it is allowed unless the URI matches one of the `Denied paths` policy rules in which case the request is denied.

3. **Web applications policy**

   If the request (including possible parameters) matches an entry in the detailed web application policy it is allowed.

4. **Web applications policy + global parameters policy:**

   If a request matches an entry in the web applications policy but one or more parameters are offending, these parameters are checked against the global parameters policy.

   If there is a combined match the request is allowed.

5. **Global URL policy + global parameters policy:**

If a requested URL with parameters matches a global URL policy pattern and all supplied parameters match global parameter patterns the request is allowed.

6. **No match:**

   The request is denied.

1.2. Basic operation

1.2.1. WAF operating mode definitions

Operating modes are sets of configurations defining what violations to block and what violations to just log. Two configurable and one non-configurable presets are available.

Protect

The Protect mode preset by default blocks and logs all violations according to the access policy.

Detect

In the default Detect mode preset only logging occurs and no blocking protection is activated. Blocking protection that would occur in Protect is logged and available for review in the deny log. Operating in the default Detect preset is comparable to an intrusion detection system - it detects and logs activities but does not protect or prevent policy violations.'

Pass

In Pass mode all requests are passed through the website proxy. No requests are blocked and no logging is performed. As no filters are active in Pass mode this mode is not configurable.

For each violation Web Security Manager can be configured to either block and log or just log.

1.2.1.1. Violations

Content violations

Path unknown	No policy rules allow the path segment of the URL, either because it does not match a positive policy rule or because it matches a negative policy rule - a signature.
Path denied	The path is explicitly denied by an URL blocking policy rule.
Query unknown	No positive policy rules match the name of the request parameter.
Query illegal	No policy rules allow the value of the request parameter, either because it does not match a positive policy rule or because it matches a negative policy rule - a signature.
Session validation failed	The request session ID is not valid, either because the session token has been tampered with or hijacked.
Form validation failed	The form submitted cannot be verified as having been issued by the web application in a response to a request from the current user session. This is an indication of a CSRF attack.
Session expired	The request session has exceeded the idle expiration threshold configured in Web Security Manager for the web application.

Malformed XML	Submitted XML request is malformed and hence cannot be parsed and validated.
Multiple or %u encoded request	The request contains elements that are encoded more than twice or it contains elements that are encoded using %u-encoding.
Authorization failed	User is not authorized to access requested resource.
Header unknown	Request header not RFC 2616 compliant.
Header illegal	Header value failed strict validation.
Header validation failed	Header value failed pragmatic validation.
Output illegal	Server response contains illegal string.

Protocol violations

Generic protocol violation	Protocol violations like missing content length or content type headers for POST requests.
HTTP Protocol version	HTTP protocol version not allowed.
Method illegal	HTTP method not allowed.
Missing hostname	Request does not specify host name.
Invalid hostname	Not website proxy is configured for the requested host name.
Request line maximum length	Entire request line (URI?query) exceeds allowed maximum length.
Request path maximum length	Request path exceeds allowed maximum length.
Query string maximum length	Request query exceeds allowed maximum length.
Content type not enabled	Request content type is supported but not enabled.
Header name length	Header name exceeds allowed maximum length.
Header value length	Header value exceeds allowed maximum length.
Maximum number of headers	Header number exceeds allowed maximum.
Upload attempt	Upload attempted but upload not allowed.
Payload length exceeded	POST payload exceeds allowed maximum size.
Maximum number of upload files	Number of files to upload in a request exceeds allowed maximum.
Total upload size	Total size of upload files in request exceeds allowed maximum.
Maximum file size	Size of a single upload file exceeds allowed maximum.
Cookie version not allowed	Request cookie version not allowed.

Maximum number of cookies	Number of cookies in request exceeds allowed maximum.
Cookie name length	Name of a cookie exceeds allowed maximum length.
Cookie value length	Value of a cookie exceeds allowed maximum length.
Maximum number of GET parameters	GET parameter number exceeds allowed maximum.
GET parameter name length	GET parameter name exceeds allowed maximum length.
GET parameter value length	GET parameter value exceeds allowed maximum length.
GET parameter combined length	Combined length of GET parameter name and value exceeds allowed maximum length.
Maximum number of POST parameters	POST parameter number exceeds allowed maximum.
POST parameter name length	POST parameter name exceeds allowed maximum length.
POST parameter value length	POST parameter value exceeds allowed maximum length.
POST parameter combined length	Combined length of POST parameter name and value exceeds allowed maximum length.
General request violation	Other generic violations.

1.2.2. Request parsing

In order for Web Security Manager to parse requests as close to the way the target application/web server technology does it is important to configure web application behaviour.

1.2.2.1. Delimiters

According to the official RFC, query part of a URL is delimited by ? and parameter part by &. Some web applications don't honor this and use different delimiters. Possible delimiters are: ";?:@&+$,". Several delimiters are separated by a space.

Re-use of delimiter characters across the three delimiter categories is not allowed.

Query delimiter(s) Input field	Characters used for delimiting query part of the URL. ***Valid input*** Characters: `;?:@&+$,` Several delimiters are separated by a space. ***Input example*** ? - /somepage.jsp**?**par1=val1&par2=val2 ***Default value*** ?

Parameter delimiter(s) Input field	Characters used for delimiting parameters in the URL. ***Valid input*** Characters: `;?:@&+$,` Several delimiters are separated by a space. ***Input example*** `&` - /somepage.jsp?par1=val1**&**par2=val2 ***Default value*** `&`
URL session id delimiter(s) Input field	Characters used for delimiting URL based session identfiiers from the rest of the query. ***Valid input*** Characters: `;?:@&+$,` Several delimiters are separated by a space. ***Input example*** `;` - /somepage.jsp**;**jsessionid=longidstring?par1=val1&par2=val2 ***Default value*** `;`

Note

Don't change these delimiters unless you are absolutely certain that you know the consequences.

1.2.2.2. Response encoding

When output rewriting or CSRF protection is enabled it is necessary for Web Security Manager to know the character set for the pages served by the web application/server in order to rewrite pages correctly. Web Security Manager will try to read the character set in use from the Content-Type header in the web server response. However, if the header does not specify a character set Web Security Manager will default to the configured charset.

Response charset default Input field	Default character set used for encoding served pages if none specified by backend server. ***Valid input*** Character set as defined in the response server header Content-Type or in the META tag content-type in the response body of pages served by the backend web server. Examples: Meta tag: <meta http-equiv="content-type" content="text/html; charset=UTF-8"> - UTF-8 Header: Content-Type: text/html; charset=iso-8859-1 - iso-8859-1

	Input example
	`utf8`
	`iso-8859-1`
	`shift_jis`
	Default value
	`utf8`

1.2.2.3. Content type - POST requests

These options are all related to parsing and validation of POST requests.

Guess Content-Type Check box	Inspect payload of POST requests to guess content type if Content-Type header not present. Default: `<disabled>`
Validate multi-part/form-data request format Check box	When parsing multipart form data require that the payload is formatted correctly. If enabled requests that does not validate correctly will be denied. Default: `<disabled>`
Block on multi-part/form-data request parsing errors Check box	When parsing multipart form data block on recoverable request parsing errors like missing data caused by content-length being too small. If enabled requests that that does not parse correctly are blocked. It is highly recommended that this setting is enabled as disabling it introduces the risk of attacks bypassing the WAF filter. Default: `<enabled>`

1.2.2.4. Case sensitivity

Some web systems match requests case sensitive and some do not. When web systems are not case sensitive it is not uncommon that samples of requests are presented in different case combinations.

To avoid requests to resources with different case being learned as different requests, case sensitivity can be disabled.

Enable case sensitivity Check box	Enable / disable case sensitivity matching. Some web systems match requests case sensitive and some do not. When web systems are not case sensitive it is not uncommon that samples of requests are presented in different case combinations. If enabled, Web Security Manager will match case sensitive. Default: `<disabled>`

1.2.2.5. Request header re-writing

Web Security Manager allows for re-writing arbitrary request header values using regular expressions for matching the value to re-write.

Enable header re-writing	Check or uncheck the checkbox **Enable request header re-writing** to enable this feature.
Rewriting rules	In the input area enter one or more rules for header re-writiting. ***Valid input*** A triplet in the format: `Header_field::match_regular_expression::subst_value`. `match_regular_expression` is a regular expression matching the substring in the header value to replace with `subst_value`. Escape meta characters (`.*+?()[]-|^$\`) with `\` to match literally. ***Input examples*** ***Rewriting Referer field value*** substitute https with http Referer::^https::http ***Rewriting X-Forwarded-For ip address*** X-Forwarded-For::10\.10\.10\.10::192.168.0.11 ***Default value*** `none`

1.2.3. Attack class criticality

For each attack class in the list define the criticality level.

Attack class Drop down lists (SQL injection, XPath injection, SSI injection, OS commanding, XSS, Path traversal, Enumeration, Format string, Buffer overflow, DoS attempt, Worm probe, etc.)	Select a criticality level for the attack class. ***Valid input*** Options from the drop down list: • `Critical` • `High` • `Medium` • `Low` • `None` ***Default value*** Attack class dependent.

1.2.4. Source IP tracking and blocking

Source IP tracking and blocking adds IP sources exceeding a certain risk level to summary database. This allows for tracking attacker activity across the websites configured in Web Security Manager. The Dashboard Deny Log Interactive List and the global Attack Source Auto Blocking are both based on the information collected when this feature is enabled.

If the

1.2.4.1. Track violating IPs across websites

Enable IP source tracking Check box	Enable / disable IP source tracking. Tracked IPs will feed into the blacklisting controls and, if enabled, IPs exceeding limits will be blocked. Default: `<enabled>`

Risk level
Drop down list
	Sets the risk level above which the source IP is tracked and added to the global database.
	Valid input
	Options from the drop down list
	`Critical`, `High`, `Medium`, `Low`, `None`
	Default value
	`<High>`

1.2.4.2. Immediate blacklisting

When a request is denied at the application level, instead of just stopping the request the source IP can be blacklisted forcing the attacker to change IP address or to find another target.

Enable IP source immediate blocking Check box	Enable / disable IP source immediate blocking. Default: `<disabled>`

Risk level
Drop down list
	Sets the risk level above which the source IP is immediately blocked.
	When the IP is blocked the attacker will not be able to access the website from that source IP for a duration configured in Attack source auto blocking.
	Valid input
	Options from the drop down list
	`Critical`, `High`, `Medium`, `Low`, `None`
	Default value
	`<High>`

1.2.4.3. Layer 7 source IP blocking

If application layer source IP blocking is enabled, when running behind a layer 7 proxy that otherwise would hide the client source IP, client source IPs are extracted from the X-Forwarded-For header and source IP based blocking controls are enforced at layer 7 instead of at the network layer.

Layer 7 source IP blocking Check box	Enable / disable Layer 7 source IP blocking. Note that this feature has to be enabled at the global level. If this is not the case this will be indicated in the field label. Default: `<disabled>`

1.2.5. External notification

1.2.5.1. Alerts and summary

Syslog alerts

Enable sending of alerts to syslog server. Only alerts with priority above or equal to the configured threshold are sent. Alerts are sent to `Local3` facility.

To have attack alerts sent to an external syslog server configure threshold level and server address in **System** → **Configuration**.

Enable alerts to syslog Check box + Drop down list	Enable or disable sending of alerts to syslog server. When enabled the drop down menu **Syslog criticality threshold** specifies the lowest informational level (priority) for which alerts will be sent to the syslog server. ***Valid input*** Options from the drop down list: • `LOG_CRIT` • `LOG_ERR` • `LOG_WARNING` • `LOG_NOTICE` • `LOG_INFO` • `LOG_DEBUG` ***Default value*** `Disabled - LOG_WARNING`

Email alerts

Enable sending of alerts by email. Only alerts with priority above or equal to the configured threshold are sent.

Enable email alerts Check box + Drop down list	Enable or disable sending of email alerts. When enabled the drop down menu **Instant email criticality threshold** specifies the lowest informational level (priority) for which alerts will be sent. ***Valid input*** Options from the drop down list: • `LOG_CRIT` • `LOG_ERR` • `LOG_WARNING` • `LOG_NOTICE` • `LOG_INFO` • `LOG_DEBUG`

	Default value `Enabled - LOG_ERR` The email address is the contact email specified in **System Configuration**.

1.2.5.2. Attack class criticality to log priority mapping

For each criticality level set the corresponding log priority (informational level).

Criticality level Drop down lists (Critical, High, Medium, Low, None)	Select a log priority level for each criticality level. ***Valid input*** Options from the drop down list: • `LOG_ALERT` • `LOG_CRIT` • `LOG_ERR` • `LOG_WARNING` • `LOG_NOTICE` • `LOG_INFO` • `LOG_DEBUG` ***Default value*** • `Critical -> LOG_CRIT` • `High -> LOG_ERROR` • `Medium -> LOG_WARNING` • `Low -> LOG_NOTICE` • `None -> LOG_INFO`

1.2.6. Deny log settings

1.2.6.1. Policy violations

Enable/disabled support for logging of blocked requests.

When a request fails the defined access policy for a given proxy, Web Security Manager will block the request.

Enable logging for normal/filtered requests Check box	Enable / disable logging for normal/filtered requests. If enabled, Web Security Manager will log blocked requests for normal/filtered end-user traffic. Default: `<enabled>`
Enable logging for pass-through requests (IP whitelisted)	Enable / disable logging for pass-through requests (Pass through mode) If enabled, Web Security Manager will log blocked requests from client matching the pass-through white-list.

Check box	Default: `<disabled>`
Do not log bypassed requests from trusted clients Check box	Disable logging of bypassed requests (that would have been blocked) from trusted clients. If checked, Web Security Manager will not log requests that are violating the policy but are bypassed because because the source IP is in the trusted clients list of the website proxy and HTTP blocking bypass is enabled for trusted clients. Default: `<disabled>`
Do not log blocked requests from trusted clients Check box	Disable logging of blocked requests from trusted clients. If checked, Web Security Manager will not log requests that get blocked if the source IP is in the trusted clients list of the website proxy. It is not recommended to disable logging of blocked requests unless there is a good reason for it. If for example some kind of monitoring software is used to regularly verify that specific requests are being blocked it could be desirable not to have these requests logged in order to prevent the log from being filled with these (known) requests. Default: `<disabled>`

1.2.6.2. Broken requests

Enable/disable logging of broken requests.

Broken requests are either requests resulting from broken internal or external links. Broken bot requests are requests originating from bots not adhering to standards.

Enable logging of broken links Check box	Enable / disable logging of referred requests (requests with a referrer header) allowed by the policy but resulting in a `404 not found` error from the web server. Default: `<disabled>`
Enable logging of webserver 404 not found Check box	Enable / disable logging of requests allowed by the policy but resulting in a `404 not found` error from the web server. Default: `<disabled>`
Enable logging of broken bot requests Check box	Enable / disable logging of requests classified as broken bot requests. Default: `<disabled>`

1.2.6.3. Log data masking

In order to avoid compromising confidential data like for instance payment card numbers (along with other payment card data like control code and expiration date) ending up in the deny log it is possible to configure log data masking policies based on regular expressions.

Enable rewriting of logged querys	Check or uncheck the checkbox **Enable rewriting of logged querys** to enable this feature.
Name	This is an informal field allowing for assigning a human readable name to the rewrite policy rule.

	Valid input Any text string ***Input example*** • Payment Card • SSN ***Default value*** `Payment card`
Search for	A regular expression matching the string to replace. ***Valid input*** A regular expression. ***Input example*** • `(?:\d{4}[\-\x20]?){2}\d{4,5}[\-\x20]?(?:\d{2,4})?` - matches most payment card numbers • `\d{3}-\d{2}-\d{4}` - matches US Social Security Number strings, no validation of the value. ***Default value*** `(?:\d{4}[\-\x20]?){2}\d{4,5}[\-\x20]?(?:\d{2,4})?` Notice the use of backslash ("\") in the examples above to escape the metacharacter ".". Without escaping the "." it will be interpreted as a metacharacter matching any character resulting in the regular expression also matching strings like xxxyhost2xxx4tld and xxxhost_xxx_tld (a.o.). The regular expressions matches case insensitive in a repetitive fashion meaning that if more than one instance of the search pattern is present in the string they will all be replaced.
Replace with	A string to replace with ***Valid input*** Any text string gramatically equal to the data being matched but with no semantic meaning (in order to mask the information in the string being matched). ***Input example*** • `9999-9999-9999-9999` • `999-99-9999` ***Default value*** `9999-9999-9999-9999` **Note** The log data input policy rules rewrites all data being handled by the website proxy log subsystem. This includes data for the

	Learner. Therefore **do not** rewrite a payment card number to something like MASKED_PAN as this will result in the Learner wrongfully selecting the class alphanum for payment card input which will not match payment card numbers with "-" (dash) or " " (space) in. Instead rewrite to something similar like in the examples above.

1.2.7. Access log settings

When *access logging* is enabled all requests to the website is logged.

The access log is generated on a per day basis and closed logs are made available for download.

Enable access logging Check box	Enable / disable access logging. If enabled, Web Security Manager will log all proxied requests to the web site. Default: `<disabled>`

Access log format
Drop down list
	Select the format for the access log.
	Valid input
	Options from the drop down list
	`Web Security Manager Common`, `Common Log Format`, `Common Log Format with Virtual Host`, `NCSA extended/combined`, `Custom`
	Default value
	`<Web Security Manager Common>`
	When the `Custom` is selected the input field below the drop-down becomes active and allows for specifying a custom log format.

Custom fortmat
Input field
	Define a custom log format.
	Valid input
	A sequence of the input fields below separated by space.
	Input example
	`remote_addr time_local request status body_bytes_sent referer`
	Default value
	`remote_addr remote_logname remote_user time_local request status body_bytes_sent referer user_agent cookie roundtrip`

Add roundtrip time and cache info to access log format Check box	Enable / disable additional proxy specific log fields. If enabled, Web Security Manager will add roundtrip time and "served from cache flag" to the selected log format. Default: `<disabled>`

1.2.7.1. Getting/viewing the access log files

When log files are available for download the filename is an active link. To download an access log file click on the filename.

When remote backup is enabled, the latest access log file made available for download will be compressed (using gzip) and copied to the remote backup destination along with the backup of the system configuration.

1.2.7.2. Access log formats

Web Security Manager supports a number of standard access log formats suitable for importing into log-analysis tools. For plain access logging the Web Security Manager format is the most condensed.

Web Security Manager Common

The Web Security Manager Common log format is a condensed version of the Common log format (below). It contains only basic HTTP access information and the time field is kept in Unix epoch format to save time and space.

Source IP	The client source IP address.
Time	Time the request was received (UNIX timestamp)
Request	First line of request
Server response code	Web server server response code - i.e. `200`
Response size	Size of response in bytes, excluding HTTP headers
Response time	The time taken to serve the request, in microseconds
Cached response	Is response served from cache or not (1=yes, 0=no)

Common Log Format

The Common log format contains only basic HTTP access information.

Source IP	The client source IP address.
Remote logname	N/A - will contain a dash, included for compatibility.
Remote user	N/A - will contain a dash, included for compatibility.
Time	Time the request was received (standard english format)
Request	First line of request
Server response code	Web server server response code - i.e. `200`
Response size	Size of response in bytes, excluding HTTP headers. In CLF format, i.e. a '-' rather than a 0 when no bytes are sent.

Common Log Format with Virtual Host

The Common log format contains only basic HTTP access information with the addition of canonical name of the Virtual Host serving the request.

Virtual Host	The canonical ServerName of the server serving the request.
Source IP	The client source IP address.
Remote logname	N/A - will contain a dash, included for compatibility.
Remote user	N/A - will contain a dash, included for compatibility.

Time	Time the request was received (standard english format)
Request	First line of request
Server response code	Web server server response code - i.e. `200`
Response size	Size of response in bytes, excluding HTTP headers. In CLF format, i.e. a '-' rather than a 0 when no bytes are sent.

NCSA extended/combined

The NCSA extended/combined format contains the same information as the Common log format plus two additional fields: the referral field and the user_agent field. This log format is also called Apache Combined log format.

Source IP	The client source IP address.
Remote logname	N/A - will contain a dash, included for compatibility.
Remote user	N/A - will contain a dash, included for compatibility.
Time	Time the request was received (standard english format)
Request	First line of request
Server response code	Web server server response code - i.e. `200`
Response size	Size of response in bytes, excluding HTTP headers. In CLF format, i.e. a '-' rather than a 0 when no bytes are sent.
Referrer	The content of the request header "Referer".
User-Agent	The content of the request header "User-Agent".

Custom format

The custom log format allows for specifying custom log formats by entering log format field names separated by space. The field names below are available.

Source IP	remote_addr	The client source IP address.
Remote log-name	remote_logname	N/A - will contain a dash, included for compatibility.
Remote user	remote_user	N/A - will contain a dash, included for compatibility.
Time	time_local	Time the request was received (standard english format)
Request	request	First line of request
Server re-sponse code	status	Web server server response code - i.e. `200`
Response size	body_bytes_sent	Size of response in bytes, excluding HTTP headers. In CLF format, i.e. a '-' rather than a 0 when no bytes are sent.
Referrer	referer	The content of the request header "Referer".
User-Agent	user_agent	The content of the request header "User-Agent".
Cookie	cookie	The content of the request header "Cookie".
Response time	roundtrip	The time taken to serve the request, in microseconds.

UNIX timestamp	timestamp	Time the request was received (UNIX timestamp).
Cached response	cache	Is response served from cache or not (1=yes, 0=no).

Additional fields

If "Add roundtrip time and cache info to access log format" is enabled the fields below will be added to the selected log format.

Response time	The time taken to serve the request, in microseconds
Cached response	Is response served from cache or not (1=yes, 0=no)

1.2.8. Mirror proxy policy from master

This feature allows for configuring the proxy to dynamically mirror the policy of another website proxy.

To mirror a proxy select it in the drop down list and enable the Mirror proxy policy from master module.

When mirroring is enabled it will be indicated in the top of the page with the text (MIRROR OF PROXY xx). Also in the websites overview the information in the Virtual Web server column will contain the text (M:X) where X is the website proxy ID.

Note that the selected mode is not mirrored so if the mirrored proxy (the master) is running in Protect mode and the mirror (the proxy for which mirroring is enabled) is running in Detect mode it will log/block according to the Detect mode preset while the mirrored proxy will use the Protect mode preset.

1.3. Protocol restrictions

1.3.1. Allowed HTTP methods, protocol versions and web services

1.3.1.1. Protocol version allowed

Restrict which HTTP protocol versions are allowed.

Corresponding violation: `HTTP Protocol version`

HTTP 1.0 Check box	Allow / disallow HTTP 1.0 requests . Default: `<allow>`
HTTP 1.1 Check box	Allow / disallow HTTP 1.1 requests . Default: `<allow>`

1.3.1.2. Methods allowed

Restrict which HTTP methods are allowed.

Corresponding violation: `Method illegal`

HEAD Check box	Allow / disallow HTTP method HEAD. Default: `<allow>`

GET Check box	Allow / disallow HTTP method GET. Default: `<allow>`
POST Check box	Allow / disallow HTTP method POST. Default: `<allow>`
OPTIONS Check box	Allow / disallow HTTP method OPTIONS. Default: `<allow>`

1.3.1.3. Web services

Web Security Manager supports inspection of XML and JSON based web services requests, including SOAP and XML RPC.

XML based requests are learned like other queries and positive and negative policies and combinations thereof can be enforced.

Corresponding violation: `Content type not enabled`

Enable XML web services support Check box	Enable / disable support for XML web services support . If enabled, Web Security Manager will parse requests with Content-Type = text/xml and treat the XML as a query. Default: `<enabled>`
Enable JSON web services support Check box	Enable / disable support for JSON web services support. If enabled, Web Security Manager will parse requests with Content-Type = application/json, text/x-json or text/json and treat the JSON request payload as a query. Default: `<enabled>`
Parse text/plain content type requests Check box	Enable / disable support for POST requests with Content Type text/plain. If enabled, Web Security Manager will accept requests with the text/plain Content Type and parse the payload of the request. As there is no standard for how the payload is composed the parser is configurable. The default configuration parses the payload as a carriage return / newline separated list of parameter name / value pairs in the form name=value. This is the format used by the Direct Web request (or DWR) Java library. To change the way the payload is parsed click the **advanced** button. This will display the regular expression that extracts the name / value pairs. If you want to change it you may want to contact Alert Logic support to get help doing it. It not complicated if you are comfortable with regular expressions though. `([^\r\n\=]+)=?([^\n\r]*)` The values are captured in the two parentheses. The first parenthesis `([^\r\n\=]+)` matches the parameter name. Note the '+' after the bracketed list of negated (^) characters. This means one or more occurrences of the characters matched by the bracketed list (anything but carriage return (\r), newline (\n) or equals (=).

	The `=?` part matches an optional equals sign. The last parenthesis ([^\n\r]*) the value but is optional as set by the asterisk (*) after the bracketed list. When changing the regular expression it is a requirement that there is at least one pair of parentheses matching something. The simplest allowed regular expression would be (.+) which will match the entire payload. When composing regular expressions note that the expression is run with the /gsi options meaning that the expression is iterated over until there are no more matches (/g), the payload is treated as one string (including \r and \n) (/s) which redefines the meaning of the meaning of the "anything" meta character (.) to include \r and \n and finally that matching is case insensitive (\i). Default: `<disabled>`

1.3.1.4. HTTP Tunneling and bypass

Web Security Manager allows for encapsulating other protocols in the HTTP protocol, so called HTTP tunneling.

Corresponding violation: `Content type not enabled`

Allow HTTP tunneling Check box	Enable / disable HTTP tunneling (Content-Type = application/octet-stream). When HTTP tunneling is enabled requests with content type application/octet-stream are passed through without parsing the payload. Default: `<disabled>`
Bypass Flash Remoting Check box	Enable / disable Flash Remoting (Content-Type = application/x-amf). When Flash Remoting is enabled requests with content type application/octet-stream are passed through without parsing the payload. Default: `<disabled>`
Bypass ActiveSync WBXML (binary XML) and message/rfc822 Check box	Enable / disable WBXML (binary xml) and message/rfc822 content types. When enabled binary XML and content type message/rfc822 will be bypassed. This is necessary for Activesync synchronization with mobile devices and outlook web access to work. Default: `<disabled>`

1.3.2. Headers, restrict length and number

Restrict length and number for HTTP request headers.

If a header fails this check, the entire request is blocked and handled accordingly.

Header name maximum length Input field	Maximum length for each inbound HTTP header name. Corresponding violation: `Header name length` ***Valid input*** An integer in the interval 1 to 8192

	Unit Bytes ***Default value*** `64`
Header value maximum length Input field	Maximum length for each inbound HTTP header value. Corresponding violation: `Header value length` ***Valid input*** An integer in the interval 1 to 8192 ***Unit*** Bytes ***Default value*** `4096`
Maximum number of headers Input field	Maximum number of HTTP headers in request. Corresponding violation: `Maximum number of headers` ***Valid input*** An integer ***Default value*** `50`

1.3.3. Cookies, restrict length and number

Restrict type, length, number and type for HTTP request cookies.

If a cookie fails this check, the entire request is blocked and handled accordingly.

Accept Version0 Check box	Allow / disallow version 0 cookies. Version 0 is most widely used on the internet today. Corresponding violation: `Cookie version not allowed` Default: `<allow>`
Accept Version1 Check box	Allow / disallow version 1 cookies. Corresponding violation: `Cookie version not allowed` Default: `<allow>`
Cookie name maximum length Input field	Maximum length for each cookie name. Corresponding violation: `Cookie name length` ***Valid input*** An integer in the interval 1 to 8192

	Unit Bytes ***Default value*** `64`
Cookie value maximum length Input field	Maximum length for each cookie value. Corresponding violation: `Cookie value length` ***Valid input*** An integer in the interval 1 to 8192 ***Unit*** Bytes ***Default value*** `1024`
Maximum number of cookies Input field	Maximum number of cookies in request. Corresponding violation: `Maximum number of cookies` ***Valid input*** An integer ***Default value*** `20`

1.3.4. Request, restrict length and number

Restrict length and number for HTTP request in general.

If the request fails this check, the entire request is blocked and handled accordingly.

Request line maximum length Input field	Maximum allowed length of the request line. When the request is displayed in the browser address bar the request line is everything following the protocol://domain.name.tld part of the request. The request line is the emphasized part of http://domain.name.tld**/path/to/resource?query=1&string=1** Corresponding violation: `Request line maximum length` ***Valid input*** An integer in the interval 1 to 8192 ***Unit*** Bytes ***Default value*** `2048`

Request path maximum length Input field	Maximum allowed length of the path part of the request line. The path part is the emphasized part of http://domain.name.tld**/path/to/resource**?query=1&string=1 Corresponding violation: `Request path maximum length` ***Valid input*** An integer in the interval 1 to 8192 ***Unit*** Bytes ***Default value*** `512`
Query string maximum length Input field	Maximum allowed length of the query part of the request line. The query part is the emphasized part of http://domain.name.tld/path/to/resource?**query=1&string=1** Corresponding violation: `Query string maximum length` ***Valid input*** An integer in the interval 1 to 8192 ***Unit*** Bytes ***Default value*** `1536`
POST form payload limit Input field	Defines the maximum allowed POST content length. If a given POST request length fails the check, the entire request is blocked and handled accordingly. Corresponding violation: `Payload length exceeded` ***Valid input*** An integer in the interval 1 to 2048000 ***Unit*** Bytes ***Default value*** `524288`

1.3.5. File uploads, restrict size and number

Maximum number of files Input field	Maximum number of allowed files to upload in request. Corresponding violation: `Maximum number of upload files`

	Valid input An integer in the interval 1 to 100 ***Default value*** `1`
Individual file size Input field	Maximum allowed size for each individual file in upload request. Corresponding violation: `Maximum filesize` ***Valid input*** An integer in the interval 1 to 1048576000 ***Unit*** Bytes ***Default value*** `2097152` (2 mb)
POST upload payload limit Input field	Maximum allowed size for entire upload request, i.e. total size of all files in upload request. Corresponding violation: `Total upload size` ***Valid input*** An integer in the interval 1 to 1048576000 ***Unit*** Bytes ***Default value*** `2097152` (2 mb)

1.3.6. Request parameters, restrict size and number

GET Parameter name maximum length Input field	Maximum length for each GET parameter name. Corresponding violation: `GET parameter name length` ***Valid input*** An integer in the interval 1 to 8192 ***Unit*** Bytes ***Default value*** `64`
GET Parameter value maximum length Input field	Maximum length for each GET parameter value. Corresponding violation: `GET parameter value length`

	Valid input An integer in the interval 1 to 8192 ***Unit*** Bytes ***Default value*** `512`
GET Parameter combined length Input field	Maximum length for each GET parameter name + value pair. Corresponding violation: `GET parameter combined length` ***Valid input*** An integer in the interval 1 to 8192 ***Unit*** Bytes ***Default value*** `576`
GET Maximum number of parameters Input field	Maximum number of GET parameters in request. Corresponding violation: `Maximum number of GET parameters` ***Valid input*** An integer in the interval 1 to 1000 ***Default value*** `100`
POST Parameter name maximum length Input field	Maximum length for each POST parameter name. Corresponding violation: `POST parameter name length` ***Valid input*** An integer in the interval 1 to 524288 ***Unit*** Bytes ***Default value*** `64`
POST Parameter value maximum length Input field	Maximum length for each POST parameter value. Corresponding violation: `POST parameter value length` ***Valid input*** An integer in the interval 1 to 524288

	Unit Bytes ***Default value*** `65536`
POST Parameter combined length Input field	Maximum length for each POST parameter name + value pair. Corresponding violation: `POST parameter combined length` ***Valid input*** An integer in the interval 1 to 524288 ***Unit*** Bytes ***Default value*** `65600`
POST Maximum number of parameters Input field	Maximum number of POST parameters in request. Corresponding violation: `Maximum number of POST parameters` ***Valid input*** An integer in the interval 1 to 8192 ***Default value*** `200`

1.4. Website global policy

1.4.1. Validate static requests separately

The Static content policy allows requests without parameters based on *file extension* (i.e. .gif) and *allowed path characters*.

To define a static content policy enter or edit **file extensions** and **allowed path characters**.

File extension

The file extension is defined as a list of comma separated values.

Allowed path characters

Allowed path characters are defined by selecting them on a list.

The letter A denotes all international alphanumeric characters and other characters are represented by their glyph, their UTF-8 number and a description.

As static content is not supposed to have any parameters (hence the denotation "static") only requests without parameters and with the method `GET` are validated against this rule.

It is possible to allow static requests in general.

Allow all static requests Radio button	If selected, requests without parameters like requests for graphic elements, stylesheets, javascript, etc. are allowed in general. Allowing all static requests is faster but less secure as only input to web applications will be inspected when this option is enabled.
Validate static requests path and extension Radio button	If selected, requests without parameters like requests for graphic elements, stylesheets, javascript, etc. are validated using allowed path extension and allowed path characters. Default: `<selected>`
Allowed static file extensions Input field	The file extension is defined as a list of comma separated values. ***Valid input*** A list of comma separated file extensions without a trailing period. ***Input example*** css,png,ico,jpg,js,jpeg,gif,swf ***Default value*** css,png,ico,js,jpg,jpeg,gif,swf
Allowed path characters List of check boxes	Allowed path characters are defined by selecting them on a the list which appears when activating the button **Edit**. In the list the letter A denotes all international alphanumeric characters and other characters are represented by their glyph, their UTF-8 number and a description. ***Valid input*** All characters in the list ***Input example*** • Hyphen-minus ("-", UTF-8: 2d) • All international alphanumeric • Space (" ", UTF-8: 20) ***Default value*** The path characters in the input example above.
Validate cookies for static requests Check box	Enable / disable validation of cookies for requests for static content. Default: `<disabled>`

1.4.2. URL path validation

The URL regular expressions filter matches URLs without parameters on a proxy global basis. If a request matches any of the defined regular expressions, it will be marked as valid by Web Security Manager and forwarded to the back-end server.

For examples of global URL regular expressions, please refer to Table 5.6, “Examples of global URL regular expressions”

Note

Full match is implied for each regular expression, meaning that each will match from the start to the end of the request (a caret ^ and dollar $ will be appended if not already present).

Negative validation Check box	Check or uncheck to enable validation of the path element of the URL against negative signatures. Paths not matching attack signatures will be allowed.
Positive validation Check box	Check or uncheck to to enable positive validation of the path element of the request URL. Paths matching one of the regular expressions in the list will be allowed.
Allowed path	In the list enter one or more regular expressions defining the global path policy. ***Valid input*** Valid regular expressions. ***Input example*** (/[\w\-]+)+\.(htm|html|shtml|pdf|asp|aspx|php|jsp) ***Default value*** `None`

1.4.3. Denied URL paths

The URL regular expressions block filter matches URLs without parameters on a proxy global basis. If a request matches any of the defined regular expressions it will instantly be blocked.

Suppose for instance that a global paths policy rule allows all URL paths's with the extension ".php" but that you want to block access to all resources in the /admin directory - including subdirectories. To do that simply add the policy rule "/admin/".

Note

The expressions are matching from left to right. Full match is not implied but matching always start at start of line. This implies that for instance the expression /admin will match any URI starting with /admin.

Denied path Input fields	In the list enter one or more regular expressions defining the global denied path policy. ***Valid input*** A valid regular expressions ***Input example*** /admin (any request starting with "/admin") /testarea (any request starting with "/testarea") .+\.php (any request for files with the extension ".php")

	.+\.htm([^l]|$) (block .htm but allow .html) ***Default value*** `none`

1.4.4. Query and Cookie validation

Depending on the web server and web application technology and design of the web applications on the back end web server cookie names and values may in some cases be parsed as part of a general request object with the risk that client request cookies may be used to bypass validation controls. It is therefore recommended that cookies are parsed and validated as an integral part of the client query. That is as request parameters.

Web Security Manager parses cookies and when learning is enabled the Learner maps cookie values as global parameters.

Cookie validation enabled Check box	If enabled client request cookies will be parsed and validated as request parameters. Default: `enabled`

1.4.4.1. Validation

In the global parameters section, parameters which all or many URLs have in common can be added. For instance in many CMS systems an URL can be viewed in a printer friendly version by adding a specific parameter to the URL.

When adding parameters to the list the name of the parameter is interpreted by Web Security Manager as regular expressions. Like with the global URL-regular expressions full match from start to end is implied. The value can either be a regular expression or a predefined input validation class.

Enable global parameter signature based negative matching Check box	Check or uncheck the checkbox **Enable global parameter signature based negative matching** to enable signature bases matching of parameter names and corresponding values. When learning is enabled for the website this option should be enabled as it ensures that parameters not being validated by positive policy rules are validated negatively and thus not rejected by default.
Enable global parameter regexp matching	Check or uncheck the checkbox **Enable global parameter regexp matching** to enable global parameter regexp matching.
Name Input fields	In the list enter a regular expression matching the parameter name or names you want to match. ***Valid input*** A valid regular expression. ***Input example*** • `\w{1,32}_btn` - matches all parameter names which start with a string of up to 32 characters and ends with the specific string '_btn'.

	• `print` - matches the specific name print. ***Default value*** `None`
Type Drop down list	Input validation type. ***Valid input*** Options from the drop down list ***Class*** A predefined named regular expression like *numeric* or *alpha-numeric*. Editing the class definition will affect all policy components that uses it. ***Regexp*** Regular expression. Please refer to Table 5.8, "Examples of global parameters regular expressions" for examples. ***Bypass*** The parameter will be completely bypassed. ***Default value*** `Class`
Update Drop down list	Controls how the Learner handles the parameter. When update is set to `manual` the parameter entry will not be maintained and updated by the Learner. When set to `auto` the entry will be maintained by the Learner.
Value Depends on `type`	Value for input validation. ***Valid input*** • A class selected from the class drop down list. • A regular expression ***Default value*** • When `type` = `class`: `num` • When `type` = `regex`: `empty` When type is `class` the corresponding regular expression of the input validation class is displayed to the right of the class selector.

For examples of specifying global parameters using regular expressions please refer to Table 5.8, "Examples of global parameters regular expressions".

For more general examples using regular expressions for input validation please refer to Table 5.7, "Examples of regular expressions for input validation".

Note

Full match is implied for each regular expression, meaning that each will match from the start to the end of the request (a caret ^ and dollar $ will be appended if not already present).

1.4.5. Headers validation

Allow only RFC defined headers Check box	Enable / disable enforcement of strict HTTP compliant headers. If enabled, Web Security Manager will enforce strict HTTP header compliance according the RFC standards and deny any custom HTTP header sent in the request. Default: `<disabled>`
Input headers validation rules Check box	The header validation policy rules allow for enforcing a combination of positive and negative validation rules on either specific named headers or all headers. "All" header rules also applies to specific named headers. For each header policy entry the options are: ***Status*** `On` or `Off` - enabling or disabling the policy rule. ***Rule type*** `Negative` or `Positive` - Negative will look for the presence of strings matching the specified regular expression (like searching for Carriage Return in a header) and Positive will require the entire header value to match the specified regular expression. ***Match*** `General` or `Named` - General will apply the policy rule to all headers and Named will only apply the policy rule to a named header. ***Header*** Only applies to `Named` header rules. The name of the header to validate using the specified regular expression. ***Regex*** The regular expression specifying the validation rule for the header. ***Description*** A description of the policy rule - like "XSS match tags".

1.4.5.1. Denied headers

Requests can be blocked/logged based on the value of a header.

Enable headers blocking Check box	Check or uncheck to enable blocking of requests based on header content. Requests with headers matching the blocking rules will denied.
Header	Enter name of header to match.

Input fields	***Valid input*** The name of an HTTP header ***Input example*** User-Agent ***Default value*** `None`
Allowed path Input fields	Regular expression to match header value. Note that full match is not implied so the regex `string` will match any value containing "string". ***Valid input*** Valid regular expressions. ***Input example*** rogue-spam-bot ***Default value*** `None`

1.4.6. Attack signatures usage

The use of attack signatures can be enabled or disabled for each request method supported.

1.4.6.1. Negative filtering column

The checkboxes in the negative filtering column enable or disable the use of attack signatures for validating input. The settings only applies to requests or or request parts for which negative filtering is enabled.

Attack Class	The name of the signature attack class
HEAD Check box	Check or uncheck to enable signature for method `HEAD`. Default: Signature dependent.
GET Check box	Check or uncheck to enable signature for method `GET`. Default: Signature dependent.
POST Check box	Check or uncheck to enable signature for method `POST`. Default: Signature dependent.
OPTIONS Check box	Check or uncheck to enable signature for method `OPTIONS`. Default: Signature dependent.

1.4.6.2. Classification column

The checkboxes in the classification column enable or disable the use of attack signatures for classifying log records and learning samples. If for instance the website takes HTML as in input like some CMS'es and bulletin board systems does this is likely to trick the Cross Site Scripting (XSS) signature. If it is not possible to white-list the IP address(es) from which the input originates

or to only allow access to the CMS via VPN it might be neccessary to disable the XSS signature in order to ensure that the Learner gets the data samples.

Attack Class	The name of the signature attack class
HEAD Check box	Check or uncheck to enable signature for method `HEAD`. Default: Signature dependent.
GET Check box	Check or uncheck to enable signature for method `GET`. Default: Signature dependent.
POST Check box	Check or uncheck to enable signature for method `POST`. Default: Signature dependent.
OPTIONS Check box	Check or uncheck to enable signature for method `OPTIONS`. Default: Signature dependent.

1.4.7. Session and CSRF protection

Web Security Manager has the ability to protect against session hijacking and CSRF (Cross Site Request Forgery) by:

1. Binding client IPs to session cookies by issuing a validation cookie containing a cryptographic token (a checksum) which validates session id + client IP + a secret for each client request.
2. By binding forms to sessions and verifying the origin of the form through insertion of a form validation parameter containing a cryptographic token which proves that the action formulator (the system issuing the page containing a form with an action) knows a session specific secret.
3. Additionally idle sessions are timed out in order to prevent users from staying logged in making them vulnerable to CSRF attacks.

When the web system issues a session cookie Web Security Manager detects it and issues a corresponding session validation cookie. In order to be able to identify the session cookie it is necessary to enter the name of the cookie containing the session id - i.e. PHPSESSID, JSESSIONID, ASPSESSIONID, SID.

An easy way to identify the session cookie name for the site you are configuring protection is to establish a session with the site (logging in, visiting the site or whatever actions are necessary to make the site issue a session cookie) and then view the cookies issued for that specific site in your browser.

Finding session cookie name in Firefox

When a session is established view the cookie in **Tools** → **Options**+**Privacy** → **Cookies** → **Show Cookies**

Enter the domain name of the site in the search field.

Session ID name Input field	The name of the cookie containing the session identifier. This field value is required to enable session and form (CSRF) protection. ***Valid input*** Any regular expression matching the name of the session id cookie.

	Input example
	`PHPSESSID`
	`JSESSIONID`
	`ASPSESSIONID`
	`ASPSESSIONID\w+ (matching asp session id's with random strings appended to the name like ASPSESSIONIDAAQTDQRT)`
	`SID`
	Default value
	`<none>`

Secret for signing checksums Input field	A hard to guess string used to generate session cookie validation tokens. ***Valid input*** Any string ***Input example*** didnqdndnwqdnqdagdiddbuqh3shjethdnssbvsunjn ***Default value*** `<random value>`
Idle session timeout Input field	Idle session timeout specifies the maximum duration of an idle session before it is dropped resulting in the user being logged out from the web site. ***Valid input*** A number (integer) in the interval 10 - 86400 (24 hours). ***Input example*** `900` - 15 minutes ***Default value*** `600`

1.4.7.1. Cookie flags

Add Secure flag to session cookie Check box	Add secure flag to session cookie to instruct users browser to only send the cookie over an SSL connection. Default: `<disabled>`
Make session cookie HttpOnly Check box	Add HttpOnly flag to session cookie to instruct users browser to make the cookie inaccessible to client side script. Default: `<disabled>`

1.4.7.2. HSTS - HTTP Strict Transport Security

HSTS is a mechanism enabling web sites to declare themselves accessible only via secure connections - HTTPS. The policy is declared by web sites via the Strict-Transport-Security HTTP re-

sponse header field. When enabling HSTS in WSM the Strict-Transport-Security header will be injected in server responses if it is not already present.

Enable HSTS Check box	Add Strict-Transport-Security header to backend server responses if not already present. Default: `<disabled>`
Max age Check box	Max age corresponds to the required "max-age" directive in the HSTS directive and specifies the number of days, after the reception of the STS header field, during which the User Agent (browser) regards the web server (from which the HSTS header was received) as a Known HSTS Host.. Default: `<365>`

1.4.7.3. Session protection configuration

Enable session protection Check box	Enable / disable validation of session identifiers. If enabled, Web Security Manager will issue a validation cookie containing a cryptographic token (a checksum) which validates session id + client IP + secret for signing checksums (above) for each client request. The validation cookie is named __PFV__ and is issued whenever Web Security Manager detects a set_cookie with a cookie name matching the value configured (above) from the web site to protect. Default: `<disabled>`
Session violation action Check box	What Web Security Manager should do when an invalid session id is detected. Session violation actions ***Block request*** The request is blocked and a session cookie with max-age=0 is sent back to the client resulting in the clients browser to expire the session cookie. ***Drop session, allow request*** The session cookie is removed from the request before the request is allowed to reach the web system. In the deny log the request will be listed with action = strip. Default: `<Drop session, allow request>`

1.4.7.4. CSRF protection configuration

Generate request form validation tokens (CSRF protection) Check box	Enable / disable generation of request form validation tokens (CSRF protection) If enabled, Web Security Manager will parse web system responses of type text/* searching for form tags. When forms tags are detected a session specific checksum validating the form action is inserted as a hidden parameter (named ___pffv___) to the form.

	Default: `<disabled>` Now go to **Policy** **Web applications** to enable request validation for specific applications (see Section 1.5.1, "Web application settings"). If configured the Learner will learn and configure CSRF protection for applications.
Form violation action Check box	What Web Security Manager should do when an invalid request is detected. Form violation actions ***Block request*** The request is blocked and a session cookie with max-age=0 is sent back to the client resulting in the clients browser to expire the session cookie. ***Drop session, allow request*** The session cookie is removed from the request before the request is allowed to reach the web system. In the deny log the request will be listed with action = strip. Default: `<Drop session, allow request>`

1.4.7.5. Request authorization configuration

Enable request authorization Check box	Enable / disable request authorization for configured web applications. If enabled, Web Security Manager will authorize access to resources based on session validity. Request authorization is only enforced for resources for which this feature is enabled. Default: `<disabled>` Now go to **Policy** **Web applications** to enable request authorization for specific applications and other resources incl. static files (see Section 1.5.1, "Web application settings").

1.4.8. Trusted clients - IP whitelisting

List if IP addresses which are trusted / whitelisted. The in- and output filters can be configured to be bypassed for the whitelisted addresses.

Whitelist Input field	Per default, requests originating from any IP address (0.0.0.0/0) is affected when Pass Through Mode is enabled. The white list allows for the definition of specific IP address(es) or networks for which Pass Through Mode is enabled. ***Valid input*** IP address with net mask (IP/mask) in CIDR notation ***Input example*** `192.168.0.8/32` - the IP address 192.168.0.8

	`192.168.0.0/24` - IP addresses 192.168.0.0 - 255 `192.168.0.8/29` - IP addresses 192.168.0.8-15 ***Default value*** `<none>`

1.4.8.1. IP pass through

IP pass through allows for configuring overriding of filter actions based on the source of the request.

Enable HTTP request blocking bypass for trusted clients Check box	Enable / disable HTTP pass through With Pass Through for trusted clients enabled, all requests will be forwarded to the real server, but will be otherwise handled the usual way (ie. Web Security Manager will learn about the site and log any would be blocked requests not matching the applied access control list). Default: `<disabled>`
Enable IP network blocking bypass for trusted clients Check box	Enable / disable network blocking pass through When enabled, IP addresses listed as trusted clients will be included in the global list of IP addresses that are allowed to bypass network blocking and DoS mitigation controls. Note that the address will not be bypassed unless network blocking bypass is allowed in **Services** **Network** Default: `<disabled>`

1.4.9. Trusted domains

The trusted domains is a whitelist of domains which is composed of 1) the domain of the website proxy virtual host and the domains of the host names in Virtual host aliases and 2) a list of other trusted domains which can be entered manually.

The effective list of trusted domains is used in Remote File Inclusion signatures to leave out URLs targeting hosts within the list and when validating redirects to allow redirects to hosts within the list.

Effective trusted domains	This is the effective list of trusted domains, i.e. the automatically generated list of the domain of the website proxy virtual host, the domains of the host names in Virtual host aliases and the manually entered domains (if any).
Other trusted domains	Enter additional domains to the list of trusted domains. Domains are separated by newline.
Include other trusted domains in domains list	When enabled the manually entered domains will be added to the effective trusted domains list.

1.4.9.1. IP pass through

IP pass through allows for configuring overriding of filter actions based on the source of the request.

Enable HTTP request blocking bypass for trusted clients Check box	Enable / disable HTTP pass through With Pass Through for trusted clients enabled, all requests will be forwarded to the real server, but will be otherwise handled the usual way (ie. Web Security Manager will learn about the site and log any would be blocked requests not matching the applied access control list). Default: `<disabled>`
Enable IP network blocking bypass for trusted clients Check box	Enable / disable network blocking pass through When enabled, IP addresses listed as trusted clients will be included in the global list of IP addresses that are allowed to bypass network blocking and DoS mitigation controls. Note that the address will not be bypassed unless network blocking bypass is allowed in **Services** **Network** Default: `<disabled>`

1.4.10. Evasion protection

Block multiple and %u encoded requests Check box	Enable / disable blocking of multiple (or %u) encoded requests. In an attempt to evade detection attackers often try to encode requests multiple times. If enabled, Web Security Manager will block requests which after being decoded still contains encoded characters. Default: `<enabled>`

1.4.10.1. Duplicate parameter names

If duplicate parameter names are allowed, wrongly configured web application behaviour may result in Web Security Manager not learning the web site correctly and may also lead to WAF bypassing vulnerabilities depending on the target application/web server technology.

An attacker may submit a request to the web application with several parameters with the same name depending on the technology the web application may react in one of the following ways:

1. It may only take the data from the first or the last occurrence of the duplicate parameter
2. It may take the data from all the occurrences and concatenate them in a list or put them in an array

In the case of concatenation it will allow an attacker to distribute the payload of for instance an SQL injection attack across several duplicate parameters.

As an example ASP.NET concatenates duplicate parameters using ',' so `/index.aspx?page=22&page=42` would result in the backend web application parsing the value of the 'page' parameter as page=22,42 while Web Security Manager may see it as two parameters with values 22 and 42.

This behaviour allows the attacker to distribute an SQL injection attack across the three parameters.

`/index.aspx?page='select data&page=1 from table` would result in the backend web application parsing the value of the 'page' parameter as 'select data, 1 from table while Web Security Manager may see it as two parameters with values `'select data` and `1 from table`.

By default when Web Security Manager validates parameters negatively it automatically concatenates the payload of duplicate parameters. It is mostly in the case where a positive application or global rule allows a specific parameter with an input validation rule that makes room for attacks like the above the parameter duplication problem exists. In the page example above the attack would be stopped because the page parameter would be learned as numeric input (an integer). This would not allow text input like in the example above. Nevertheless it is important to configure Web Security Manager to mimic the target web applications parsing of requests as closely as possible.

Block duplicate parameter names Check box	Enable / disable blocking of duplicate parameter names. If enabled, Web Security Manager blocks requests containing duplicate parameter names. Default: `<disabled>`
Join duplicate parameter names Check box	Enable / disable concatenation duplicate parameters. If enabled, Web Security Manager will concatenate the values of the duplicate parameters using the configured join separator (below). Default: `<enabled>`
Join separator Input field	Character(s) used for separating concatenated parameter values. ***Valid input*** A string of 0 to 5 characters ***Input example (in quotes)*** `','` - comma, ASP and ASP.NET ***Default value*** `''` - empty string

The best option is to disallow duplicate parameter names. It may not be practical though as the use of duplicate parameters may be intended in some applications - the most prominent example being PHP which parses parameter names suffixed with `[]` as an array - like par1[]=22&par1[]=42 becoming array(22,42). If this feature is not in use block it.

If the application technology is ASP/IIS or ASP.NET/IIS and it is not possible to disallow duplicate parameters the recommended setting is to join duplicate parameters using ',' as in the join separator example above.

1.4.11. Time restricted access

Access to a website can be restricted on a time basis.

1.4.11.1. Opening hours

For each weekday enter opening hours.

Opens Input field	Time the website opens on the weekday. ***Valid input*** 24h time string in the format hh:mm.

	Input example 08:00 ***Default value*** `00:00`
Closes Input field	Time the website closes on the weekday. ***Valid input*** 24h time string in the format hh:mm. ***Input example*** 18:00 ***Default value*** `24:00`

1.4.11.2. Website is closed

To specify dates where the website is closed enter a list of dates in the format mm/dd separated by whitespace, comma or semicolon.

1.4.11.3. When website is closed redirect to

URL to redirect the visitor to when website is closed.

This field is required.

1.4.12. Input validation classes

Characters classes are useful when you want to use a predeclared set of criteria used by Web Security Manager for input request validation. Eg. if you have lots of HTML forms that use an input field "email", you can define a class and a regular expression which defines what a valid e-mail address is. This class can then be used throughout the entire policy.

When a `class` is changed, all affected policy elements are automatically updated to reflect the change.

Rank Read only	The class rank when used by the Learner. To change the rank, place the cursor in one of the classes input fields. The rank number will be indented. Use the buttons **Move up** and **Move down** in the lower button panel to change the class' rank.

Name
Input field
	The class' name.
	Valid input
	A text string. No spaces or special characters.
	Input example
	`my_class`
	Default value
	`<none>`

Value Input field	The class regular expression. ***Valid input*** A valid regular expression. Full match is implied for each regular expression, meaning that each will match from the start to the end of the request (a caret ^ and dollar $ will be appended if not already present). ***Input example*** `[A-Za-z]{1,32}` - a string of max. 32 7-bit letters. ***Default value*** `<none>`
X Button	Mark class for deletion. When classes are saved the marked classes will be deleted. When deleting classes that are in use in the policy you will be prompted to accept replacement of the deleted classes with existing classes. Learner data samples using deleted classes will be deleted.

For more information about classes and their corresponding regular expressions refer to Section 1.8, “Regular expressions”.

1.4.12.1. Negative signatures policy

Some user input is so complex and unpredictable that, to avoid false positives, positive validation of input ends up being very general and loose. An example of this is free text input fields which often get mapped to the input validation class "printable" which basically allows all printable characters. It is often better validate such input negatively - which Web Security Manager does by default.

Web Security Manager determines if an input should be validated negatively based on the input validation class rank. By default the threshold is the class `Printable`. If a parameters input is learned/configured to be the class configured as threshold the signatures policy will be used instead of the class regular expression.

Move up	Change the rank of the selected class. To move the class upwards. Select the class by clicking anywhere in the class row. When selected the class rank number is highlighted and indented. Click **Move up** to move the class one step upwards.
Move down	Change the rank of the selected class. Works as described above.
Add new	Add new class. When clicked an empty row will appear at the bottom of the class list. Fill out the blanks and place the class in the class hierarchy with the move buttons.
Use negative checking above and including class rank	The class rank above and including which input will be validated negatively.

Drop down	***Valid input***
	Values in the drop down list.
	Input example
	`standard`
	Default value
	`printable`
	To disable negative class checking select `disabled` in the list.

1.5. Web applications

The *Web applications section* allows for defining policy rules with a scope that is limited to specific web applications.

Web applications are either added manually or they are automatically created created by the Learner.

1.5.1. Web application settings

Requests
Drop down list
	Configure URL request status.
	When set to deny all requests for the web application will be denied.
	Valid input
	Options from the drop down list
	`allow` or `deny`
	Default value
	`allow`

Update Drop down list	Configure URL update setting. ***Valid input*** Options from the drop down list `auto` or `manual` When update is set to `manual` the ACL entry will not be maintained and updated by the Learner. When set to `auto` the entry will be maintained by the Learner. ***Default value*** `auto` - `manual` when URL added manually
Violation action Drop down list	Action to take when a request for the web application is denied. When set to block or detect this setting will override the global setting for the violation at hand. ***Valid input*** Options from the drop down list

	Use global Global settings will be used. ***Protect*** Block settings (as defined in global violation action) will be used no matter if the website is running in Protect or Detect. ***Detect*** Detect settings (as defined in global violation action) will be used no matter if the website is running in Protect or Detect. ***Pass*** Bypass violations for this specific application. Violations will neither be blocked nor logged. ***Default value*** `Use global`

1.5.2. Global violation action override

Global violation action override allows for an even more fine grained violation action exception handling than simply specifying violation action for the web application. This override feature allows for specifying exceptions from the global violation action on a per violation type basis.

If, for instance, you have an application that generates "Malformed XML" because of some custom built client application sending XML requests that does not conform to standards it is possible to specify a policy exception for that specific violation for that specific application. This way you do not have to bypass XML validation globally or put the entire application in Pass or Detect mode.

To add a violation exception:

1. Select the violation type from the drop down list **`Global violation action override`**
2. The selected violation type is listed above the drop down with two action types: One for global Protect mode and one for global Detect mode.
3. For each mode select the desired action which can be either of Protect, Detect or Pass.

1.5.3. Methods allowed

Restrict which HTTP methods are allowed.

Corresponding violation: `Method illegal`

HEAD Check box	Allow / disallow HTTP method HEAD. Default: `<allow>`
GET Check box	Allow / disallow HTTP method GET. Default: `<allow>`
POST Check box	Allow / disallow HTTP method POST. Default: `<allow>`
OPTIONS	Allow / disallow HTTP method OPTIONS.

Check box	Default: `<allow>`

1.5.4. Session protection

Require a valid session to access this resource Check box	Enable / disable authorization of access to this resource based on session validity. If enabled, whenever this resource is requested, Web Security Manager will only allow the request if it originates from a valid user session. Note that session protection and request authorization have to be enabled for resource request authorization to be effective - see Section 1.4.7, “Session and CSRF protection” Default: `<disabled>`
Enable request origin validation for this application Check box	Enable / disable validation of requests resulting from forms with this application as action. If enabled, whenever a request for this application contains a specific parameter (see below) it is verified that the request origins from a form on a web page / application belonging to the web system and that the form has been issued on a page belonging to the current users session. Note that for the validation token to be generated Generate request form validation tokens (CSRF protection) has to be enabled - see Section 1.4.7, “Session and CSRF protection” Default: `<disabled>`

Validate parameter name
 	
Input field
	String specifying the name of a specific parameter to be present for Web Security Manager to perform request origin validation.
	
	Valid input
	
	A string definining a parameter name.
	
	Input example
	
	`amount`
	
	Suppose for instance that you want to validate a money transfer request from a logged in user. With request CSRF protection configured (Session protection / generation of request form validation tokens / origin validation) enabled should a legitimate logged in user be tricked into issuing a forged request like the example on wikipedia <img src="http://bank.example/withdraw?account=bob&amount=1000000&for=mallory"> then the origin of the request does not validate. In this case because the validation parameter (__pffv__) is not present.
	
	Default value
	
	`<none>`

1.5.5. Parameters

This section contains a list of current defined parameters with corresponding input validation type and value and other settings.

To update a parameter simply change the values and click on the **Save** button.

Select parameter Check box	Check or uncheck to mark for deletion. Default: <unchecked> To mark an entry for deletion, check the box. When the parameter list is saved the parameter will be deleted.
Name Input field	String specifying the parameters name. ***Valid input*** A string defining a name. No regular expressions. ***Input example*** `print` - the parameter print ***Default value*** The parameter name
Type Drop down list	Input validation type. ***Valid input*** Options from the drop down list ***Class*** A predeclared regular expression used by Web Security Manager for input request validation. Classes are defined on a proxy global basis. When a class is modified all parameters using that class is affected. ***Static*** Legitimate values for the parameter can only have fixed values defined. Values are separated by a newline. ***Regexp*** Legitimate input values for the parameter are based on the regular expression defined. Only one regular expression is allowed. ***Default value*** Class
Value(s) Depends on `type`	Value for input validation.

	Valid input ***Class name*** When type `Class` is selected a drop down menu is available in the **values** field. Input validation for the parameter is based on the regular expression corresponding to the selected class name. ***Static values*** Input values are validated against the static list of legitimate values for the parameter. If they match, the request is allowed by Web Security Manager. otherwise, the entire request is blocked. ***Regular expression*** input values are validated against the defined regular expression. if they match, the request is allowed by Web Security Manager. otherwise, the entire request is blocked. For examples of using regular expressions for input validation please refer to Table 5.7, "Examples of regular expressions for input validation" **Note** full match is implied for each regular expression, meaning that each will match from the start to the end of the request (a caret ^ and dollar $ will be appended if not present) ***Default value*** Class `Numeric`
Negative Check Drop down	Use negative checking if validation class is above configured threshold. If set to Auto the policy configured in classes negative signatures policy will be applied when validating input. ***Valid input*** Drop down options ***Default value*** `Auto`
Update Drop down	Configure how the parameter should be handled by the Learner. If set to `Upgrade only` the Learner will only change the parameter if the class needs to set to a higher rank (relaxed). When set to `Auto` the Learner will make all changes to the parameter, including removing it if it later is learned to be a global parameter. ***Valid input*** Drop down options

	Default value `Upgrade only`

1.6. Output filter

The *Output filter section* allows for configuring policies for rewriting headers and body of server responses sent to the client.

1.6.1. Backend server cloaking

A typical web server gives out a lot of information about it's version, installed software, operating system, etc.

This information is completely irrelevant for normal HTTP/HTTPS communication between clients and web server. However, attackers and worms can misuse this information to craft more targeted attacks on a vulnerable web application or server.

Server ID Input field	The server name string that is sent in respones to clients in the Server header. When the website proxy is created the value is extracted from the backend server response in a short form. Leave the field empty to omit the Server header from responses. ***Valid input*** Alphanumeric, space, dash, underscore and period. ***Input example*** Apache/2.2 ***Default value*** Backend server banner without details
Enable Web server cloaking mode Check box	Enable / disable Web server cloaking mode. If enabled, Web Security Manager removes web server information from the response sent back from the back-end server before forwarding it to the client thus protecting the web application and server from leaking potentially sensitive information. This includes stripping of all HTTP headers that start with "X-". Eg. header "X-Powered-By: PHP/4.4.0" will be removed. Default: `<enabled>`
Intercept backend error pages Check box	Intercept error pages with error code 400 or higher sent by the backend web server and replace with a general error page. Configure error pages in Section 2.2, “Error messages” Default: <checked> When enabled the original error code can be replaced with a general one (ie. 405 > 404) or the original error can be sent to the client.

	Show original backend error code The original error code is sent and the error code and name is displayed in the error message. ***Generalize backend error codes*** A general error code is sent and displayed.
Exclude status codes Input field	Exclude specific error codes from error interception (if enabled). ***Valid input*** list of error codes separated by space ***Input example*** 401 403 ***Default value*** <empty>

1.6.2. Output headers validation and rewriting

1.6.2.1. Redirects validation

Redirects validation protects against attacks that redirect victims to phishing or malware sites through target applications that use untrusted data to determine the destination pages.

Block redirects to non trusted domains	When enabled Web Security Manager will validate redirects from the protected web applications and only allow redirects to domains in the **trusted domains** whitelist.
Whitelist	The whitelist is the effective list of trusted domains. Redirects are allowed to hosts in the domains in this list. The list can be edited in Trusted domains (Section 1.4.9, "Trusted domains") in the global policy section.

1.6.2.2. Response headers rewriting

Web Security Manager allows for re-writing arbitrary response header values using regular expressions for matching the value to re-write.

Enable response header re-writing	Check or uncheck the checkbox **Enable response header re-writing** to enable this feature.
Header	In the list enter a the name of the header to match. ***Valid input*** Any header field. ***Input example*** • `Location` - matches a redirect response header. • FooBar - matches the custom header field FooBar.

	Default value `none`
Search for	A regular expression matching the string to replace. ***Valid input*** A regular expression. ***Input example*** • `xxxhost\.xxx\.tld` - matches `xxxhost.xxx.tld` • `[a-z]{1,32}\.xxx\.tld` - matches any host name in the xxx.tld domain consisting of characters a-z (case insensitive) with length 1 - 32 characters. • `http://` - matches http:// ***Default value*** `none` Notice the use of backslash ("\") in the examples above to escape the metacharacter ".". Without escaping the "." it will be interpreted as a metacharacter matching any character resulting in the regular expression also matching strings like xxxyhost2xxx4tld and xxxhost_xxx_tld (a.o.). The regular expressions matches case insensitive in a repetitive fashion meaning that if more than one instance of the search pattern is present in the string they will all be replaced.
Replace with	A string to replace with ***Valid input*** Any text string ***Input example*** • `yyyhost.yyy.tld` • `newhost.yyy.tld` • `https://` ***Default value*** `none`

1.6.3. Output body validation and rewriting

Web Security Manager allows for parsing and rewriting the body of server responses. This is useful for screening (and replacing) output for confidential data like credit card numbers. Note however that rewriting server responses involves parsing the complete document and therefore will introduce added latency.

It is important that the correct response content type is configured in Web application behaviour.

Search for	A regular expression matching the string to replace.

	Valid input A regular expression. ***Input example*** • `(?:\d{4}[\-\x20]?){2}\d{4,5}[\-\x20]?(?:\d{2,4})? -` matches a payment card number ***Default value*** `none` As with the response header rewrite function the the regular expressions matches case insensitive in a repetitive fashion meaning that if more than one instance of the search pattern is present in the string they will all be replaced. Also meta characters should be escaped if they are to be interpreted literally.
Action Drop down	Action to take if there is a search match. Replace: replace matched string with replace string. Block: block the rest of the response and log the violation. ***Valid input*** Drop down options ***Default value*** `Replace`
Replace with	A string to replace with ***Valid input*** Any text string ***Input example*** • `masked_payment_card` ***Default value*** `none`

1.6.3.1. Exceptions

Web Security Manager can be configured to not rewrite the response body if the request is originating from trusted clients or if the requested path matches a regular expression.

Do not re-write from whitelisted IP's (trusted clients)	Check or uncheck the checkbox to exclude requests from trusted clients / whitelisted IP from being rewritten. The list of trusted clients is edited in the global policy section
Do not re-write from URIs matching regex	Enter a regular expressions matching the path part of the requests to be excluded. Only responses with content types text/*`[sometype]`* will be rewritten.

	Valid input A valid regular expression ***Input example*** ^/forms/ (do not rewrite requests starting with /forms/) ^.+\.js (do not rewrite files with the extension ".js") ***Default value*** none

1.7. Authentication

1.7.1. SSL client authentication

1.7.1.1. Client certificate authentication

Verify Client Drop down list	This directive enables the verification of the client identity. When set to deny all requests for the web application will be denied. ***Require*** Ask for the client certificate and only allow client access if a valid certificate is presented. ***Optional*** Ask for the client certificate and checks the client identity if the certificate is presented by the client.
Verify Depth Input field	Sets how deep Web Security Manager should go in the client provided certificate chain in order to verify the client identity. ***Valid input*** An integer in the interval 1 to 50. ***Default value*** 10

1.7.1.2. Certificate Authority certificates

One or more Certificate Authority certificates are required for authenticating clients.

To upload an authority certificate click the **Add Certificate Authority certificate** button. This will expand an area in which you paste the certificate authority certificates. Be sure to include the -----BEGIN CERTIFICATE----- and -----END CERTIFICATE----- lines.

To view detailed certificate information click the + in the left column.

1.7.1.3. Certificate forwarding

The entire client certificate or specific certificate information can be forwarded to the backend web server in HTTP request headers.

The information selected will be forwarded in HTTP headers with the name of the selected info, ie. SSL_CLIENT_CERT will be forwarded in the header SSL_CLIENT_CERT.

1.7.2. SSL client Certificate Revocation Lists (CRLs)

Web Security Manager uses Certificate Revocation Lists (CRL) to support certificate revocation.

To use CRL configure a location where Web Security Manager can retrieve CRLs. When configured CRLs are downloaded and compiled at regular intervals to make sure that CRL updates are included.

Downloaded CRL files are displayed in the table below the CRL location rules.

Enable Check box	Check to enable CRL checking for certificate. Default: <unchecked>
CA Certificate Drop down list	Select the CA certificate to enable CRL checking for.
Location URL Input field	A URL that points to the directory the where the CRL files are served from or directly to a CRL-file if CRLs for the CA are served as one (big) file. ***Valid input*** A URL ***Input example*** http://crl.eid.belgium.be/ - the CRL repository for the Belgian eID. ***Default value*** none
Type Drop down list	The type of the Location URL - index or file. ***Index*** The URL Location points to a directory that contains a number of CRLs that has to be downloaded. ***CRL File*** The URL Location points directly to a single file that contains all CRLs for the CA.
Wildcard Input field	If the URL Location type is Index it is necessary to specify a wildcard that matches the CRL files in the location. ***Valid input*** A simple wildcard. Use the following characters to specify wildcards: * = any string any length. ? = one occurrence of any character.

	Input example *.crl - matches all files with extension .crl ***Default value*** *
Test Button	Before saving the CRL rule click the Test button. This will display all CRL files in the location matching the CRL location rule.

1.7.3. SSL client authorization

In addition to generally restricting access to the website based on validity of client certificate it is possible to specify requirements which has to be fulfilled in order to allow access to website resources defined as paths.

Suppose you have an organisation, Snake Oil, that have a website to which they only want employees with valid client certificates issued by Snake Oil to be able to access in general. In the website there is an area, /management/, that they only want management to be able to access.

The first requirement can be achieved by requiring a valid Snake Oil certificate to access the website.

The second requirement can be met by creating an authorization rule that matches the Organisational Unit of the Subject DN, like: Location=/management, SSL_CLIENT_S_DN regex match OU=management.

Location	In the list enter one or more regular expressions defining locations. Note that the expressions are matching left to right so the expression can be a simple path like `/management` which match any path in the website tree starting with /management but not /employees/management. ***Valid input*** A valid regular expressions ***Input example*** /admin (any request starting with "/admin") /management (any request starting with "/management") .+\.php (any request for files with the extension ".php") ***Default value*** `none`
Requirements	One or more requirements to be met for the client to be allowed access to the specified location. Using the operators AND and OR complex requirements can be specified. The AND operator precedes the OR operator so O=Snake Oil AND OU=management OR O=Rattle Snake Oil AND OU=admin will be evaluated as (O=Snake Oil AND OU=management) OR (O=Rattle Snake Oil AND OU=admin).

	Client cert field Select the cert field to match ***SSL_CLIENT_SERIAL*** The serial of the client certificate ***SSL_CLIENT_S_DN*** Subject DN in client's certificate ***SSL_CLIENT_I_DN*** Issuer DN of client's certificate ***SSL_CIPHER*** Cipher for established SSL-connection ***SSL_CLIENT_CERT*** PEM-encoded client certificate ***SSL_CLIENT_VERIFY*** Verification status - SUCCESS, FAILED or NONE if no cert granted by client ***SSL_PROTOCOL*** Protocol for the established SSL connection ***SSL_SESSION_ID*** SSL session ID for the established SSL connection ***Match type*** Select `Regex Match` to require a match or `Regex No Match` to negate match. ***Match criteria*** A regular expression to match data in the client cert field selected. Example: If SSL_CLIENT_S_DN is selected `OU=management` would match certificates where the client cert has Organisational Unit = management.

1.8. Regular expressions

Web Security Manager has full support for standard PCRE (Perl Compatible Regular Expressions).

Following below is a brief regular expression "survival guide". For a more thorough explanation of the subject some links and books are recommended at the end of the section.

1.8.1. What are regular expressions

A regular expression is a formula for matching strings that follow some pattern.

Regular expressions are made up of normal characters and special characters. Normal characters include upper and lower case letters and digits. The characters with special meanings and are described in detail below.

In the simplest case, a regular expression looks like a standard text string. For example, the regular expression "john" contains no special characters. It will match "john" and "john doe" but it will not match "John".

In an input validation context we always want the expression to match the whole string. The expression above would now be expressed as ^john$, where the characters ^ and $ means starting of line and end of line. Now john will only match "john" but not "john doe". To obtain match of "john doe" as well as "john smith" etc. we employ a few more simple special characters. In its simplest form "john *lastname*" could be expressed as "^john.*$" meaning: A string starting with the characters "john" followed by zero or more (the "*") occurrences of any character (the "."). For those familiar with the simple wild-card character "*" in (a.o.) DOS and Unix, ".*" equals "*" - that is: *anything*.

Specifying *anything* is not very useful in an input validation context. With regular expressions much more fine grained input validation masks can be defined with the rich set of meta characters, character classes, repetition quantifiers, etc.

A brief explanation with some examples follows below.

1.8.2. Metacharacters

`^`	Beginning of string (implied in Web Security Manager)
`$`	End of string (implied in Web Security Manager)
`.`	Any character except newline
`*`	Match 0 or more times
`+`	Match 1 or more times
`?`	Match 0 or 1 times; or: shortest match quantifier (i.e. *?)
`|`	Alternative (like logical OR)
`()`	Grouping
`[]`	Set of characters (a list of characters)
`{}`	Repetition modifier
`\`	Quote or special

Table 5.1. Metacharacters in regular expressions

To present a metacharacter as a data character standing for itself, precede it with \ (e.g. \. matches the full stop character "." only).

Note

In Web Security Manager all regular expressions are forced to match the entire string (URL path or parameter value) by automatically prefixing an expression with "^" and suffixing it with "$".

1.8.3. Repetition

`a*`	Zero or more a's
`a+`	One or more a's
`a?`	Zero or one a's (i.e., optional a)
`a{m}`	Exactly m a's

`a{m,}`	At least m a's
`a{m,n}`	At least m but at most n a's
`repetition?`	Same as repetition but the shortest match is taken

Table 5.2. Repetition in regular expressions

Read "a's" as "occurrences of strings, each of which matches the pattern a".

Read *repetition* as any of the repetition expressions listed above it.

Shortest match means that the shortest string matching the pattern is taken. The default is "greedy matching", which finds the longest match.

1.8.4. Special notations with \

`\t`	tab
`\n`	newline
`\r`	return (CR)
`\xhh`	character with hex. code hh
`\b`	"word" boundary (zero space assertion)
`\B`	not a "word" boundary
`\w`	matches any single international character classified as a "word" character (alphanumeric or _). Examples: A, z, 1, 9, Æ, â
`\W`	matches any non-"word" character
`\s`	matches any whitespace character (space, tab, newline)
`\S`	matches any non-whitespace character
`\d`	matches any digit character, equiv. to [0-9]
`\D`	matches any non-digit character
`\pN`	Matches any UNICODE character classified as numeric

Table 5.3. Notations with \ in Web Security Manager regular expressions

1.8.5. Character sets [...]

A character set is denoted by [...]. Different meanings apply inside a character set ("character class") so that, instead of the normal rules given here, the following apply:

`[characters]`	matches any of the characters in the list (c,h,a,r,a,c,t,e,r,s)
`[x-y]`	matches any of the characters from x to y (inclusively) in the ASCII code
`[\-]`	matches the hyphen character -
`[\n]`	matches the newline; other single character denotations with \ apply normally, too
`[^something]`	Negation. Matches any character except those that [something] denotes; that is, immediately after the leading [the circumflex ^ means "not" applied to all of the rest

Table 5.4. Character sets in regular expressions

1.8.6. Lookaround

The lookaround construct allows for the creation of regular expressions matching *something* but only when it is followed/preceded or *not* followed/preceded by *something else*. Note that the lookaround construct is a zero-width assertion. It is testing for a match of something else but it will not actually match it - that is why it is called an assertion. The lookaround constructs allows for the creation of otherwise impossible or too complex expressions.

In an input validation context look ahead could be used for specifying an expression allowing angle brackets <> but only when they are not closed.

a(?!*expression*)	Negative lookahead. Matches "a" when not followed by *expression*, where expression is any regular expression.
a(?=*expression*)	Positive lookahead. Matches "a", when followed by *expression*.
(?<!*fixed-expression*)a	Negative lookbehind. Matches "a" when not preceded by *fixed-expression*, where fixed-expression is any regular expression specifying a fixed number of characters. That is "aaa" wil work but a+ will not work.
(?<=*fixed-expression*)a	Positive lookbehind. Matches "a" when preceded by *fixed-expression*.

Table 5.5. Lookaround in regular expressions

1.8.7. Examples

1.8.7.1. Global URL regular expressions

The URL regular expressions filter matches URLs without parameters on a proxy global basis. If a request matches any of the defined regular expressions, it will be marked as valid by Web Security Manager and forwarded to the back-end server.

Expression	Matches
`(/[\w\-]+)+\.html`	URL with the extension html containing any international word characters, digits, _ and -. (\w matches upper and lower case alphanumeric characters plus _).
`/abc(?:/[\w\-]+)*\.html`	Same URL starting with `/abc`, including the URL `/abc.html`.
`(/[\w\-]+)+\.html?`	Same URL matching extensions `html` and `htm`
`(/[\w\-]+)+\.(html|pdf)`	Same URL matching extensions `html` and pdf.
`(/[abcdefgh]+)+\.html`	URL with the extension `html` containing any of the lower case letters `abcdefgh`.
`/index\.html`	Exact match of `/index.html`
`(/[\w\-]+)+/?`	"Natural" URL containing international alphanumeric characters, digits, _ and `-`.
`/sw[0-9]{0,12}\.asp`	URL with the extension `asp` starting with `/sw` followed by 0-12 digits.
`/(login|logout)`	Only URLs `/login` or `/logout`

Expression	Matches
`(/[\w\-]+)+\.(htm|html|shtml|pdf)`	Any international characters URL with one of the extensions htm, html shtml or pdf.

Table 5.6. Examples of global URL regular expressions

1.8.7.2. Validating input parameters

regular expression	matches
`^[\w \.@()\-]+$`	International alphanumeric characters, underscore, a space, dot, @, parentheses and a dash.
`^[0-9a-za-z. ]+$`	digits, ASCII characters a-z, a dot and a space.
`^[0-9]+$`	only digits. `[0-9]` can also be expressed as `\d`
`^[\d]{1,5}$`	one to five digits.
`^[a-z]+$`	only lower case ASCII characters from a-z.
`^[a-z]{0,32}$`	matches only lower case ASCII characters from a-z and limits the total length to maximum 32 characters.

Table 5.7. Examples of regular expressions for input validation

1.8.7.3. Global parameters

When specifying global parameters both the name and the value are defined using regular expressions.

Name	Value	Matches
`usepf`	`true`	The specific parameter `usepf` with the static value `true`
`parm\d{3}`	`[a-zA-Z\d]{3,32}`	All parameters with name starting with `parm` followed by three digits with the value any combination of letters `a-Z` (upper and lowercase) or digits with a minimum length of 3 and a maximum length of 32 characters.
`\w{1,25}`	`[\w\s_,/:()@$*\.\-]*`	Any parameter with name consisting of international word characters and with values containing zero or more"friendly characters".

Table 5.8. Examples of global parameters regular expressions

1.8.7.4. Predefined standard classes in Web Security Manager

The following classes are predefined in Web Security Manager. The classes are presented in the order the `Automatic Policy Generator` evaluates them when automatically mapping classes to input parameters.

Class	Regular expression	Description
empty		No values allowed

Class	Regular expression	Description
num	`\d{1,32}`	Digits - a maximum of 32 digits
payment_card	`(?:\d{4}[\-\x20]?){2}\d{4,5}[\-\x20]?(?:\d{2,4})?`	Payment card numbers, allows for spaces and hyphens between number groups.
ms_ident	`{?[A-Za-z0-9]{8}-[A-Za-z0-9]{4}-[A-Za-z0-9]{4}-[A-Za-z0-9]{4}-[A-Za-z0-9]{12}}?`	Microsoft identifier with optional preceding and trailing curly brackets.
alphanum	`\w{1,256}`	International alphanumeric characters. No spaces. max. 256 chars.
text	`(?!.*(\.\.|//).*)[\w\x20+.,\-:@=/]+`	Simple international text.
url	`(?:https?://)?(?!.*(\.\.|//).*)[\w\x20@,.(){}/\-=?&]+`	Simple international URL match. With parameters. Consecutive "/" or "." not allowed (negative look ahead)
standard	`[\w\s@,.(){}/\-=?&_:]+`	Text input, international, several special characters allowed including newline.
printable	`[^\x00-\x08\x0b\x0c\x0e-\x1f\x7f]+`	Any number of printable characters. Defined by negating character class containing non-printable characters.
anything	`.+`	Anything but newline.
Anything_multiline	`(?:.|\n)*`	Anything including newline.

Table 5.9. Predefined standard classes in Web Security Manager

1.8.8. Further reading

A number of web sites and books are describing regular expressions in more detail.

1.8.8.1. Web sites

Wikipedia

A general description

http://en.wikipedia.org/wiki/Regular_expression

The 30 Minute Regex Tutorial

The code project

.NET specific tutorial, includes a software tool for testing.

http://www.codeproject.com/dotnet/RegexTutorial.asp

Regular-Expressions.info

Excellent web site dealing extensively with the subject.

http://www.regular-expressions.info

1.8.8.2. Books

There are many good books covering regular expressions. Here we mention a few.

Regular Expression Pocket Reference

Introduction and quick reference from O'Reilly.

http://www.oreilly.com/catalog/regexppr/index.html

Regular Expression Pocket Reference

Introduction and quick reference from O'Reilly.

http://www.oreilly.com/catalog/regexppr/index.html

Mastering Regular Expressions, Second Edition

Learning to use regular expressions efficiently. Does not pretend to be introductory in any way. Also from O'Reilly.

http://www.oreilly.com/catalog/regex2/index.html

Sams Teach Yourself Regular Expressions in 10 Minutes

Sounds appealing. If you are new to regular expressions this is probably a good place to start. From Sams Publishing.

http://www.samspublishing.com/title/0672325667

2. Deny and error handling

When a request is blocked at the application level Web Security Manager can either just close the connection and not respond at all, send an HTTP error code along with an error message or redirect the client to a URL.

2.1. Deny action

Web Security Manager distinguishes between violations that are Query and Authentication. () and Parameter (given value for a known parameter failed the access policy)

URL Policy Violation

Violations related generally to the URL like HTTP method and headers, path and parameter names.

Parameter Policy Violation

Violations related to the content of query parameters.

Authentication Required

Violations related to authentication and authorization.

For each type a Deny Action can be configured.

Deny with [deny type] Radio button	Display *404 not found* or *403 authentication required* error message. When a request is denied the corresponding error page (403 or 404) is displayed. Default: `<selected>`
Close connection Radio button	Close the connection. When a request is denied Web Security Manager simply closes the connection. No response is sent to the offending client. Default: `<not selected>`
Redirect Radio button	Redirect the request. When a request is denied Web Security Manager sends HTTP/302 and a Location redirect HTTP header which redirects the offending client to the URL configured. Default: `<not selected>`

2.2. Error messages

Web Security Manager intercepts error messages from the backend and replaces them with a generic customizable error page. These are also the pages that are displayed If Web Security Manager is configured to display an error message when a request is denied.

The error pages are customizable and timed redirects can be inserted.

2.2.1. Document not found (error 40x)

When a request is denied with an error message or if the backend server returns an HTTP error 40x (400 401 402 404 405 406 407 408 409 410 411 412 413 414 415 416 417) the `Document not found` page is displayed.

Heading Input field	The heading of the message page. ***Valid input*** Any string ***Default value*** `Requested URL cannot be found`
Message Input field	The message displayed. ***Valid input*** Any string not containing html tags. Newlines are transformed into . Use [b]text[/b] to put some text in bold typeface. Use [p]paragraph text[/p] to insert paragraphs. ***Default value*** `We are sorry, but the page you are looking for cannot be found. The page has either been removed, renamed or is temporarily unavailable.`
Error Input field	The error message displayed. ***Valid input*** Any string ***Default value*** `HTTP 404 Not Found`
Nav. back Input field	The error page contains two navigation buttons. The **nav. back** button will take the user to the page the user came from. ***Valid input*** Any string ***Default value*** `Back to previous page`
Nav. forward Input field	The error page contains two navigation buttons. The **nav. forward** button will take the user to the web site homepage. ***Valid input*** Any string ***Default value*** `Proceed to homepage`
Include redirect text and script	Enable / disable insertion of timed redirect javascript with corresponding text.

Check box	If enabled a redirect text and a piece of javascript displaying a configurable countdown is displayed with the error text configured (above). Default: `<disabled>`
Redirect text Input field	The redirect message displayed. ***Valid input*** Any string not containing html tags. Newlines are transformed into . Use [b]text[/b] to put some text in bold typeface. Use [p]paragraph text[/p] to insert paragraphs. Use [countdown] to display countdown. Use [link]link text[/link] to insert link to configured redirect target server. ***Default value*** `You will be redirected to a an error page in [countdown] seconds. [link]Click here[/link] to be redirected immediately.`
Redirect delay Input field	Idle session timeout specifies tha maximum duration of an idle session before it is dropped resulting in the user being logged out from the web site. ***Valid input*** A number (integer) in the interval 2 - 3600 (one hour). ***Input example*** 60 - (one minute) ***Default value*** `10`

Redirect URL
Input field
	The URL to redirect to.
	Valid input
	A valid URL
	Input example
	`http://sorryserver.mydomain.tld`
	Default value
	`none`

Web Security Manager text Read only Trial license only.	In Web Security Manager Trial error messages contains the message Web Security Manager web application firewall - TRIAL VERSION

2.2.2. Authentication required (error 403)

When a client request fails authentication or resource authorization and the request is denied with an error message or if the backend server returns an HTTP error 403 the `Authentication required` page is displayed.

Heading Input field	The heading of the message page. ***Valid input*** Any string ***Default value*** `Not allowed`
Message Input field	The message displayed. ***Valid input*** Any string ***Default value*** `Access to the page you are trying to access is restricted to authorized clients. Please contact the site administrator if this is an error.`
Error Input field	The error message displayed. ***Valid input*** Any string ***Default value*** `HTTP 403 Forbidden`
Nav. back Input field	The error page contains two navigation buttons. The **nav. back** button will take the user to the page the user came from. ***Valid input*** Any string ***Default value*** `Back to previous page`
Nav. forward Input field	The error page contains two navigation buttons. The **nav. forward** button will take the user to the web site homepage. ***Valid input*** Any string ***Default value*** `Proceed to homepage`

Include redirect text and script Check box	Enable / disable insertion of timed redirect javascript with corresponding text. If enabled a redirect text and a piece of javascript displaying a configurable countdown is displayed with the error text configured (above). Default: `<disabled>`
Redirect text Input field	The redirect message displayed. ***Valid input*** Any string not containing html tags. Newlines are transformed into . Use [b]text[/b] to put some text in bold typeface. Use [p]paragraph text[/p] to insert paragraphs. Use [countdown] to display countdown. Use [link]link text[/link] to insert link to configured redirect target server. ***Default value*** `You will be redirected to a an error page in [countdown] seconds. [link]Click here[/link] to be redirected immediately.`
Redirect delay Input field	Idle session timeout specifies tha maximum duration of an idle session before it is dropped resulting in the user being logged out from the web site. ***Valid input*** A number (integer) in the interval 2 - 3600 (one hour). ***Input example*** 60 - (one minute) ***Default value*** `10`

Redirect URL
 	
Input field
	The URL to redirect to.
	
	Valid input
	
	A valid URL
	
	Input example
	
	`http://sorryserver.mydomain.tld`
	
	Default value
	
	`none`

Web Security Manager text	In Web Security Manager Trial error messages contains the message Web Security Manager web application firewall - TRIAL VERSION

Read only Trial license only.	

2.2.3. Server error (error 50x)

When the the backend server returns an HTTP error 50x (500 501 502 503 504 505 506 507) the the `Server error` page is displayed.

Heading Input field	The heading of the message page. ***Valid input*** Any string ***Default value*** `Requested URL cannot be found`
Message Input field	The message displayed. ***Valid input*** Any string ***Default value*** `We are sorry, but the page you are looking for cannot be found. The page has either been removed, renamed or is temporarily unavailable.`
Error Input field	The error message displayed. ***Valid input*** Any string ***Default value*** `HTTP 502 Bad Gateway`
Nav. back Input field	The error page contains two navigation buttons. The **nav. back** button will take the user to the page the user came from. ***Valid input*** Any string ***Default value*** `Back to previous page`
Nav. forward Input field	The error page contains two navigation buttons. The **nav. forward** button will take the user to the web site homepage. ***Valid input*** Any string ***Default value*** `Proceed to homepage`

Include redirect text and script Check box	Enable / disable insertion of timed redirect javascript with corresponding text. If enabled a redirect text and a piece of javascript displaying a configurable countdown is displayed with the error text configured (above). Default: `<disabled>`
Redirect text Input field	The redirect message displayed. ***Valid input*** Any string not containing html tags. Newlines are transformed into . Use [b]text[/b] to put some text in bold typeface. Use [p]paragraph text[/p] to insert paragraphs. Use [countdown] to display countdown. Use [link]link text[/link] to insert link to configured redirect target server. ***Default value*** `You will be redirected to a an error page in [countdown] seconds. [link]Click here[/link] to be redirected immediately.`
Redirect delay Input field	Idle session timeout specifies tha maximum duration of an idle session before it is dropped resulting in the user being logged out from the web site. ***Valid input*** A number (integer) in the interval 2 - 3600 (one hour). ***Input example*** 60 - (one minute) ***Default value*** `10`

Redirect URL
 	
Input field
	The URL to redirect to.
	
	Valid input
	
	A valid URL
	
	Input example
	
	`http://sorryserver.mydomain.tld`
	
	Default value
	
	`none`

Web Security Manager text	In Web Security Manager Trial error messages contains the message Web Security Manager web application firewall - TRIAL VERSION

Read only Trial license only.	

2.3. Lower button bar

Default values	Revert to default values.
Save settings	Click **Save settings** to save settings.

3. Learning

The Learner builds a complete profile of the web site including static requests, web applications and input parameters bu analyzing incoming requests.

To avoid learning from worms, attacks and other unauthorized access the Learner employs a combination of heuristic attack classification, statistics and server responses.

When learning is enabled for the website the Learner keeps analyzing requests until no changes to the resulting policy are recorded. That is, for every 10,000 requests the Learner builds a trial policy, compares it to the former trial policy and records the number of changes. When a configurable number of trial policies in a row (default 30) has not resulted in a number of changes between each trial build exceeding a configurable threshold (default 0) a policy is built.

By default the Learner is configured to generate a short yet fine grained policy. This is achieved by identifying global characteristics of the web site and generating global patterns matching those characteristics. The global patterns typically account for the majority of the web systems content and applications leaving only the "real" web applications to be accounted for by specific web application policy entries.

3.1. Learning data

3.1.1. Applications learned

Applications learned is shown as a 3-level expandable table.

Applications learned Expandable: Click **+** to expand. Expands 2 levels.	Group, URL path and details. ***Application group (level 1)*** Applications are divided into groups based on path characteristics. The group name reflects the characteristics of the group. The most common grouping criteria is the file extension. But also the appearance of special characters like '$' or '.' in the path is used as grouping criteria. ***Applications URL paths (level 2)*** When a group is expanded the URL paths in that group is listed. Each URL path is an application learned. Note that this list also contains "simple" applications, applications that only takes global parameters as input, and therefore potentially can be very long. ***Application details (level 3)*** When an application URL path is expanded the details learned about that specific application is shown.
Paths	Number of unique URL Paths in the group. Applies to: Group level (1).
Param	Number of parameters the application takes as input.

	If a blue number in parentheses is shown at the left of the number this number indicates how many of the parameters learned that are approved based on the Learner thresholds which are configurable. Parameters that does not exceed one or more threshold values are colored blue while trusted parameters name are black. Applies to: URL path level (2).
Class	Name of input validation class mapped to a parameter. If the parameter is not trusted yet, the class name is blue. Applies to: Detail level (3).
Source	Number of unique IP-addresses requesting the resource. Applies to: Group (1), URL path (2) and Detail level (3).
Time	Number of unique timestamps in requests for the resource. Applies to: Group (1), URL path (2) and Detail level (3).
Time (delta time)	Time difference between the first and last observed request for the resource. Applies to: Group (1), URL path (2) and Detail level (3).

3.1.1.1. Deleting applications or corresponding parameters

To delete a learned application or a corresponding parameter expand to the level desired and click the red X.

3.1.2. Global parameters learned

The Global parameters learned section shows all parameters observed on a number of paths that exceeds the Learner setting Global parameters *Path duplication threshold*.

Note that the list also includes observed parameter names which are still pending approval based on the Learner threshold settings. The number of approved, or trusted, observations is indicated with black number while a blue number shows the number of non-approved observations.

Global parameters Expandable: Click **+** to expand. Expands 1 level.	Group, URL path and details. ***Parameter name (level 1)*** Name of the parameter ***Applications URL paths (level 2)*** When Global parameter is expanded a list of URL paths which are observed taking the parameter as input is shown.
Class	Name of input validation class mapped to a parameter. Applies to: Parameter name level (1).
Paths	Number of unique URL Paths observed using the parameter. Applies to: Parameter name level (1).
Pending	Number of unique URl Paths using the parameter but where the parameter name is not approved yet - where threshold values is not reached yet.

	Applies to: Parameter name level (1).
Trusted	Number of unique URl Paths using the parameter where the parameter name *is approved* - where threshold values is reached. Applies to: Parameter name level (1).

3.1.3. Static content learned

This section shows all URL Paths to static resources learned. URL Paths are grouped by their extension.

Static content learned Expandable: Click **+** to expand. Expands 1 level.	Extension and URL Paths learned. ***Extension (level 1)*** The static content policy is based on allowing extension and URL Path based on characters in the URl path. To be included in the static content policy, static resources must therefore have a file extension. A case where natural URLs are pointing to static content is handled by the Learner by building Global URL policies. ***Static content URL paths (level 2)*** When an extension is expanded the URL paths in that extension group is listed.
Paths	Number of unique URL Paths in the extension group. Applies to: Extension level (1).
Source	Number of unique IP-addresses requesting the resource. Applies to: Extension (1) and URL path level (2).
Time	Number of unique timestamps in requests for the resource. Applies to: Extension (1) and URL path level (2).
Time (delta time)	Time difference between the first and last observed request for the resource. Applies to: Extension (1) and URL path level (2).

3.1.3.1. Deleting static content extensions

To delete a static content extension (a group) click the red X in the list.

3.1.4. Tools

This contains tools for tidying the learning data set.

Delete querys by name wildcard Input field	Delete learned parameter names using simple wildcard matching. ***Valid input*** A string or a simple wildcard. Use the following characters to specify wildcards: * = any string any length.

	`?` = one ocurrence of any character. ***Input example*** `http://*` - matches all querys (parameter names) beginning with http:// ***Default value*** `<none>` **Preview** displays parameter names matching the wildcard below the input field. **Delete** performs deletion of parameters matching wildcard.
Delete querys by data Input field	Delete learned parameter names using matching occurrence data. **Source** Number of IP addresses requesting the resource. ***Valid input*** `number` in range 0 - ***Input example*** `10` - Querys requested by 10 or less IP addresses. ***Default value*** `<none>` **Time** Number of unique timestamps in requests for the resource. ***Valid input*** `number` in range 0 - ***Input example*** `10` - Querys requested in a maximum of 10 intervals of 1 second. ***Default value*** `<none>` **Time (delta time)** Time difference between the first and last recorded request for the resource. ***Valid input*** Time interval specified in seconds. `number` in range 0 - ***Input example*** `86400` - Querys with a recorded difference between first and last request of maximum 24 hours (24 * 60 * 60).

	Default value `<none>` **Preview** displays parameter names matching search criteria below the input fields. **Delete** performs deletion of parameters matching search criteria.

3.1.5. Lower button bar

The lower button bar contains the following buttons.

Re-analyze data Button	To see the effect of deleting selected learning data in the resulting policy section click this button. Wait a few seconds and reload the page.
Reset learn data Button	Use with caution! When clicking this button and accepting the confirm pop-up window. *All learning data for that proxy will be deleted!* If learning is enabled the learning and data sampling process will start from scratch.

3.2. Learning status

3.2.1. Learning progress indicators

The two bars in the top of the page indicates the current state of sampling and verification.

The Learner works in two stages when profiling the website.

1. **Data sampling**

   This is the process of collecting information about the website in terms of what paths/applications are used, what parameters do they take as input, what extensions are used for static content, etc.

2. **Verification**

   The verification process 1) validates the data samples using statistical methods like analyzing spread in IP sources and time, number of requests, etc. and 2) verifies that the resulting policy covers the requests sampled.

   As the Web Security Manager Learner extracts characteristics like extensions, specific directories in paths and global parameters (parameter names a number of applications take as input - like print=1) and even patterns used in global parameters the verification process may start before the Data sampling progress has reached 100%.

   Verification is calculated as the number of sample runs in a row with no policy changes relative to the required number configured in learner settings.

   When Verification has reached 100% Web Security Manager will either build and commit a new policy or notify the administrator by email that verification has reached 100% and a new policy can be built and committed.

3.2.2. Policy history

When a new policy is generated and committed, either automatically or manually, it is added to the Policy history list.

Policy history	The policy number.
Type	Automatic or manual (requested by administrator user)
Changes	Click link to see resulting policy and changes compared to the former (if any).
Sample run	The sample run number at which the policy was generated.
Web Apps	The number of entries in the `Web Application Policy`.
Global URLs	The number of entries in the `Global URL Policy`.
Global Parms	The number of entries in `Global Parameter Policy`.
Static	The number of entries in static file types `Static Content Policy`.

3.2.3. Resulting policy

This section shows a sample of the policy resulting from the *Learner* settings effective.

When the settings are changed the resulting policy sample is rebuilt using the new threshold values. This is done as a background job and depending on the load on the Web Security Manager node and the complexity of the sample data it may take anywhere from e few seconds to a minute or two to build the policy. If the new policy is not visible yet, wait a while and refresh the window.

Commit to WAF Button	Builds a policy which is accessible and editable in the Global Patterns and Web applications windows. When clicked the policy displayed in the table will be committed to the WAF engine, that is: made active for filtering requests. If policy verification has not reached a warning message (two actually) will be displayed asking to confirm the action. Remember: Web Security Manager is a white-list based web application firewall. If the policy put into production does not match real life requests building the policy prematurely (not fully verified) is likely to result in false positives. If verification has not reached 100% it means that it not verified that the policy does not generate false posivites. Have patience and wait for verification to reach 100%.
Web applications Expandable: Click **+** to expand.	Learned web applications. Expand the item to get a list of applications learned. For each application is shown: • URL path • Methods learned • Parameters. Parameters are shown as name, value pairs where the value is the name of the input validation class learned for that parameter.

	Note that only the applications private parameters are shown here. Parameters which the application have in common with other applications are included in the Global parameters list.
Global URL patterns Expandable: Click **+** to expand.	Global URL Path Policy built from learned applications. For each application group (see Section 3.1.1, “Applications learned” below) a regular expression is built which matches all samples in that specific group. Most CMS based web systems have a number of global parameters, like for instance `print=1`, which can be appended to most requests. Without the combination of *Global URL Path Policy* and *Global Parameters Policy* pages with static content that take global parameters, like `index.php?print=1`, would be learned as web applications and the URL paths would have to be added to the policy as web applications. This can potentially result in a huge policy which is never up to date because new content is added all the time. By making global policies that account for all the static content which is served dynamically only "real" web applications with a number of private parameters have to be mapped in detail. Thus the global patterns allows for building a condensed, yet fine grained, policy which also account for future standard content added to the web site.
Global parameters Expandable: Click **+** to expand.	Global Parameters Policy built from learned applications. Displayed in the format: `name = value` Depending on the Name grouping threshold value the name can either be a literal string or a regular expression matching a number of parameter names with name and value similarities. The value is displayed as a class name. When the policy is built the corresponding regular expression will be used.
Static content allowed extensions Expandable: Click **+** to expand.	Learned static path extensions which will be allowed.
Static content path allowed characters Expandable: Click **+** to expand.	Unique characters and character classes (like 'A' - all international word characters) learned from static path samples. Also the regular expression built to match requests for static content is shown. Note the last set of parantheses preceded by an escaped period `\.(\w+)`. This part will be matched with the list of allowed extensions to determine if the extension is alowed.

3.2.4. Sample run information

The Learner analyzes request samples in chunks of approximately 10,000 requests (or more if the system is very busy) . For each sample run an entry is added to the Sample run information table which shows total and delta values of summarizing the learning process.

Sample run	The sample run number.
Hits total	The total number of hits processed during the learning process.
URL paths	Total number of unique URL paths identified.
Parameters	Total number of unique parameter names identified. Uniqueness is determined by URL path. Two parameters with the same name but mapped as belonging to different URL paths are therefore identified as two unique parameters. When the policy is built Web Security Manager identifies parameters with similar names and input data as as global in scope and builds global patterns matching such parameters.
Changes	When the chunk of raw sample data has been processed Web Security Manager builds a policy based on the total sample population. This policy is compared to the policy built in the last sample run and changes are recorded. The number shown is the sum changes recorded to the Web Application Policy (`ACL`), Global URL Policy (`GURL`), Global Parameter Policy (`GParm`) and the Static Content Policy (`EXT`). Click on the number shown to get a change report detailing the changes.
ACL	The number of changes to the `Web Application Policy` compared to the sample run before.
GURL	The number of changes to the `Global URL Policy` compared to the sample run before.
Gparm	The number of changes to `Global Parameter Policy` compared to the sample run before.
Ext	The number of changes to the `Static Content Policy` compared to the sample run before.

Note

The number of policy changes recorded is calculated with the *Learner* settings effective when the sample data is analyzed. Whereas the resulting policy (below) is recalculated when the *Learner* settings are changed this is not the case with the sample run policy builds. It is therefore possible that the two sections show different results. The next sample run is run using the new settings.

3.2.5. Lower button bar

The lower button bar contains the following buttons.

Re-analyze data Button	To see the effect of deleting selected learning data in the resulting policy section click this button. Wait a few seconds and reload the page.
Reset learn data	Use with caution!

Button	When clicking this button and accepting the confirm pop-up window. *All learning data for that proxy will be deleted!* If learning is enabled the learning and data sampling process will start from scratch.

3.3. Learning settings

3.3.1. Policy generation options

Learning Drop down list	Enable/disable learning
Develop static extensions list Check box	Enable / disable static extensions learning. If enabled, Web Security Manager will treat static content separately and develop a static content policy (from learned static content. Default: `<enabled>` See Section 1.4.1, "Validate static requests separately" for more information
Enable global parameters generation Check box	Enable / disable global parameters generation If enabled, Web Security Manager will identify parameters which many or all learned applications have in common. If a (configurable) number of applications takes a specific parameter as input the parameter will be learned as a global parameter and added to the Global Parameters Policy (Section 1.4.4, "Query and Cookie validation"). Default: `<enabled>`
Enable global parameters name grouping Check box	Enable / disable global parameters name grouping. If enabled, Web Security Manager™ will analyze the global parameter names to identify name similarities and build parameter groups based on common characteristics. If the number of parameter names in a group exceeds a configurable threshold a parameter name pattern will be built matching all parameter names in the group. Grouped parameter names with corresponding input validation classes are inserted in the Global Parameters Policy (Section 1.4.4, "Query and Cookie validation"). Default: `<enabled>`
Develop static extensions list Check box	Enable / disable static extensions learning. If enabled, Web Security Manager will treat static content separately and develop a static content policy (from learned static content. Default: `<enabled>` See Section 1.4.1, "Validate static requests separately" for more information

Enable autostart of policy generation Check box	Enable / disable autostart of policy generation. If enabled, when a (configurable) number of sample data chunks has been processed without resulting in a number of policy changes exceeding thresholds a policy will be generated, the operating mode will automatically be changed to `Detect` and the Learner will stop collecting data samples. Default: `<enabled>`
Learn applications only Check box	Only learn applications. If enabled, WSM will only learn from requests with parameters. This will keep WSM from constantly adding new URL paths to the learning database for sites that use natural URL versions of for instance blog entries. Default: `<disabled>`
Avoid learning from broken bots Check box	Enable / disable checks for broken robots. If enabled, Web Security Manager will try to identify requests originating from robots not behaving correctly. An example is robots that for example maps the URL /index.asp?page=8&print=1 but for some reason translates the print parameter to &print=1 when requesting it. Because the parameter &print is not requested in in general from many different sources it will never exceed threshold values and consequently will not be included in the policy - but it is annoying to look at. Default: `<enabled>`
Learn from hostile sources (IPs) Check box	Enable / disable exclusion of sample data from hostime sources. If disabled, requests from sources from which entries in the deny log classified as attacks also originates will not be included in the sample population used for generating the policy. As all learning samples are valideted against negative policy rules obvious attacks will not be included no matter what the setting of this option is. But as attackers often "sneak around" trying different probes to detect vulnerability against for instance SQL injection (entering `O'Neill` instead of `'or 1=1`) chances are that the classes mapped for input validation becomes looser than they have to. Disabling learning from hostile sources reduces the likelihood that this will happen. Default: `<disabled>`
Auto enable request origin validation (CSRF protection) Check box	Enable / disable automatic activation of request origin validation. If Session protection and generation of request form validation tokens (CSRF protection) is enabled (see Section 1.4.7, “Session and CSRF protection”) the Learner will map applications taking input from forms generated by the web system by detecting the validation token parameter (___pffv___) inserted by Web Security Manager and correlating other input parameters to the presense of the validation token. If parameters are detected that are only present in requests where the validation token is also present (like "amount" or "submit") then the application is created in the web applications policy with one of these

	parameters as "validation parameter" - that is: a parameter which when present in requests will trigger a validation of the request form origin based on the validation token which is tied to the current user session. If Auto enable request origin validation (CSRF protection) is enabled the Learner will map the validation parameter and enable origin validation (CSRF protection) for that application. If disabled the Learner will only map the validation parameter. Default: `<disabled>`
Keep validation settings for enabled applications Check box	Enable / disable automatic overwriting of request origin validation activation settings. This input is only active when Auto enable request origin validation (CSRF protection) is disabled. Suppose you want the Learner to map validation parameters for request origin validation but that you only wan to activate it for certain applications. In order to avoid the Learner overwriting the activation settings for the activated applications next time it develops a policy activate this control. Default: `<disabled>`

3.3.2. Global parameters

Path duplication threshold Input field	Define how many unique paths (applications) are required to take the parameter as for the parameter to be regarded global. ***Valid input*** Number of paths ***Input example*** 5 ***Default value*** 3
Name grouping threshold Input field	Define how many occurrences of a global parameter with similar patterns in name it requires for the generation of a name pattern. ***Valid input*** Number of parameters ***Input example*** 5 ***Default value*** 3

3.3.3. Policy verification

Policy verification thresholds allow for granular control of when the Learner will generate a policy or notify by email that thresholds are reached.

Web application policy changes threshold Input field	Define the upper threshold of web application policy changes. ***Valid input*** Number of changes to the web application policy. ***Input example*** `0` ***Default value*** `0`
Static content policy changes threshold Input field	Define the upper threshold of static content policy changes. ***Valid input*** Number of changes to the static content policy. ***Input example*** `0` ***Default value*** `0`
Global parameters policy changes threshold Input field	Define the upper threshold of global parameters policy changes. ***Valid input*** Number of changes to the global parameters policy. ***Input example*** `0` ***Default value*** `0`
Global URL patterns policy changes threshold Input field	Define the upper threshold of global URL patterns policy changes. ***Valid input*** Number of changes to the global URL patterns policy. ***Input example*** `0` ***Default value*** `0`

Verification runs
Input field
	The Verification runs threshold controls how many trial policies without changes exceeding the threshold values below are required before a learned policy is considered verified and ready to be committed to WAF.
	The process of verifying the policy before committing to WAF is important because it reduces the risk of false positives.
	Valid input
	Number of trial policies built in a row without changes.
	Input example
	`30`
	Default value
	`20`

3.3.4. Learning thresholds

To avoid learning from worms, attacks and other unauthorized access the Learner employs a combination of heuristic attack classification, statistics and server responses.

The statistic analysis is based on aggregates, delta, min values etc.

The statistics based approving of request samples is divided into approving:

1. URL Paths based on the URL Paths group membership.

   This approval only affects the URL Path, not parameters and associated input values.

2. Parameters

3.3.4.1. Path groups

The approval of URL paths, applications as well as static resources, is based on the URL Path group membership.

The threshold values below control the statistics based approval of groups.

IP addresses
Input field
	The minimum number of unique IP addresses observed requesting URL paths belonging to a group.
	Valid input
	Number
	Input example
	`500`
	Default value
	`100`

Timestamps Input field	The minimum number of unique timestamps observed on requests for URL paths belonging to a group. The timestamp granularity is in seconds.

	Valid input
	A number in the interval `0 - 9999999999`.
	Input example
	`300`
	Default value
	`100`

Time spread
Input field
	The minimum difference in seconds between the first and the last request for URL Paths belonging to the group.
	Valid input
	A number in the interval `0 - 9999999999`.
	Input example
	`259200 (3 days)`
	Default value
	`36000 (ten hours)`

3.3.4.2. Query names

The threshold values below control the statistics based approval of query names.

Note that the threshold values applies to unique URL Path/Query combinations.

The term Query name refers to a request parameter name (ie. *name*=value).

IP addresses Input field	The minimum number of unique IP addresses observed requesting the URL Path/Query combination. ***Valid input*** Number ***Input example*** `100` ***Default value*** `100`
Timestamps Input field	The minimum number of unique timestamps observed requesting the URL Path/Query combination. The timestamp granularity is in seconds. ***Valid input*** A number in the interval `0 - 9999999999`. ***Input example*** `100`

	Default value `100`
Time spread Input field	The minimum difference in seconds between the first and the last request for the URL Path/Query combination. ***Valid input*** A number in the interval `0 - 9999999999`. ***Input example*** `36000 (ten hours)` ***Default value*** `36000 (ten hours)`

3.3.4.3. Input validation class selection for query values

The threshold values below control the statistics based selection of input validation class selection for approved Querys (above).

The term Query value refers to a request parameter value (ie. name=*value*).

The methods available depend on the license type.

Determine valid input class using value frequency analysis Check box + Input fields	If enabled input validation class selection will be based on value the relative frequency of class samples. This method utilizes that in most cases valid samples will vastly outnumber invalid samples (like for instance attack probes not matching signatures - searching for o'neill for instance to test for proper input handling in forms). Input validation classes are ranked according to possible complexity in input with simple classes having the lowest rank. When input values to a parameter are learned the values are mapped to input validation classes. The higher the rank of the class the more general input is accepted. When the policy is built the class with the highest rank is chosen provided enough samples of the class has been recorded with respect to its relative weight in the sample population in terms of hits, unique IP sources and unique timestamps. Input fields for relative thresholds: **Source frequency threshold** ***Valid input*** A value in the interval `0.0 - 99.9` ***Input example*** `1.5`

	Default value
	`1.0`
	Timestamp frequency threshold
	Valid input
	A value in the interval `0.0 - 99.9`
	Input example
	`1.5`
	Default value
	`1.0`
	Hits frequency threshold
	Valid input
	A value in the interval `0.0 - 99.9`
	Input example
	`1.5`
	Default value
	`1.0`

Determine valid input class using value counting
Check box +
Input field
	If enabled input validation class selection will be based on value counting.
	Class samples required for query value
	Input validation classes are ranked according to possible complexity in input with simple classes having the lowest rank.
	When input values to a parameter are learned the values are mapped to input validation classes. The higher the rank of the class the more general input is accepted.
	When the policy is built the class with the highest rank is chosen provided enough samples of the class has been recorded. This threshold is defined by
	Class samples required for query value.
	Valid input
	A number in the interval `0 - 9999999999`.
	Input example
	`10`
	Default value
	`1`
	The lower threshold selected the higher is the risk that a few invalid samples will affect the class selection resulting in a policy that is too lose.

3.3.5. Learn data sampling

Learn data sampling settings allow for limiting learn data sampling to specific source IP addresses or specific URL Paths. Similarly it is possible to exclude learning from IP addresses and URL Paths.

Path - Only learn from the paths below
Check box +
Input field
	If enabled the Learner will *only record* sample data from the URL Paths specified in the input area.
	In combination with very general global policies it is possible to learn and filter specific applications only.
	Valid input
	One or more URL path regular expresions separated by new-line.
	Input example
	`/cgi-bin/.*`
	Default value
	`<none>`

Path - Do not learn from the paths below
Check box +
Input field
	If enabled the Learner will *not record* sample data from the URL Paths specified in the input area.
	Valid input
	One or more URL path regular expressions separated by new-line.
	Input example
	`/admin/.*`
	Default value
	`<none>`

IP - Only learn from the IP addresses below Check box + Input field	If enabled the Learner will *only record* sample data from the IP addresses specified in the input area. ***Valid input*** IP address with net mask (IP/mask) in CIDR notation ***Input example*** `192.168.0.8/32` - the IP address 192.168.0.8 `192.168.0.0/24` - IP addresses 192.168.0.0 - 255 `192.168.0.8/29` - IP addresses 192.168.0.8-15 ***Default value*** `<none>`
IP - Do not learn from the IP addresses below Check box +	If enabled the Learner will *not record* sample data from the IP addresses specified in the input area.

Input field	***Valid input*** IP address with net mask (IP/mask) in CIDR notation ***Input example*** `192.168.0.8/32` - the IP address 192.168.0.8 `192.168.0.0/24` - IP addresses 192.168.0.0 - 255 `192.168.0.8/29` - IP addresses 192.168.0.8-15 ***Default value*** `<none>`

3.3.6. Lower button bar

Build policy Button	Builds a policy which is accessible and editable in the Global Patterns and Web applications windows. When clicked a confirm dialog is shown with the question: "Disable data sampling and switch to detect mode when a policy is generated?" Select *cancel* if the Learner should continue the data sampling and learning process and *OK* if you want the Learner to switch to *detect mode*. If *cancel* is selected the built policy willhave no effect but the editing and reporting tools will be available.
Reset learn data Button	Use with caution! When clicking this button and accepting the confirm pop-up window. *All learning data for that proxy will be deleted!* If learning is enabled the learning and data sampling process will start from scratch.
Default values	Revert to default values.
Save settings	Click **Save settings** to save settings.

4. Log

4.1. Deny log

The Deny log window provides access to all denied request to the proxy. Filtering functions allows for specification of fine grained filtering of log information.

4.1.1. Specifying filter criteria

The filter function allows you to specify conditions for showing a subset of the log entries. Until reset the filter conditions also apply to the log report.

When the log filter section is not expanded a **Filter button** and current filter criteria is shown on a general level. When filter criteria are defined a **reset button** will be available at the left of the filter button. When the reset button is pressed the filter criteria will be reset.

When the Filter button is clicked the filter section expands and filter criteria can be specified. Following filter criteria are available:

ID
Input field
	Number identifying a log entry.
	Valid input
	Number of type integer.
	Input example
	`20567`
	Default value
	`<none>`

Path Input field	Pattern or string specifying filter based on the URL path. ***Valid input*** A string or a simple wildcard. Use the following characters to specify wildcards: `*` = any string any length. `?` = one occurrence of any character. ***Input example*** • `/store/*` - matches all URL paths beginning with string /store/ including the string itself. • `*.php` - matches all url paths in all sub directories with the extension .php . ***Default value*** `<none>`
Parameters Input field	Filter based on the number of parameters. ***Valid input*** `digits` and operators `<`, `>` and `=`

	Input example • `>3` - more than 3 parameters. • `2` - exactly two parameters. ***Default value*** `<none>`

IP
Input field
	Source IP address of the originating client
	Valid input
	An IP address in the format xxx.xxx.xxx.xxx
	Input example
	`192.168.45.17`
	Default value
	`<none>`

Host
Input field
	Host information from the request blocked.
	Valid input
	A string or a simple wildcard.
	Use the following characters to specify wildcards:
	`*` = any string any length.
	`?` = one ocurrence of any character.
	Input example
	`www.mycompany.com`
	`*.mycompany.com`
	Default value
	`<none>`

Date from
Input field
	Filter based on request timestamp. Date from specifies the date of the oldest log records that should be included.
	Valid input
	A date string in the format: mm/dd/yyyy
	Input example
	`02/27/2008`
	Default value
	`<none>`

Date to Input field	Filter based on request timestamp. Date to specifies the date of the newest log records that should be included. Use *Date from* and *Date to* to specify a time interval.

	Valid input
	A date string in the format: mm/dd/yyyy
	Input example
	`02/29/2008`
	Default value
	`<none>`

Attack classification Multiple checkboxes	Filter based on attack classification. ***Valid input*** Any combination of checked items in the list of attack classes. ***Default value*** `<none>`
Policy violation Multiple checkboxes	Filter based on policy violation. ***Valid input*** Any combination of checked items in the list of policy violations. ***Default value*** `<none>`
Reset Button	Resets the filter criteria to default values.
Apply Button	Applies defined filter to deny log database.
Close Button	Closes the filter section.

4.1.2. Blocked and failed requests

Displays requests for resources for the selected proxy that were blocked by Web Security Manager.

HTTP headers, URL, parameters and values (if any) that were blocked in the request are highlighted in red color.

Also failed requests are shown in the deny log allowing for identifying broken internal and external links and broken robots not abiding the 404 not found message.

Total number of log entries matching the current filter criteria (if specified) is displayed as **Query returned #number records**. If the total number of records is larger then the **Entries per page** selection, use navigation arrows to navigate the log record back and forth.

Details are expandable: Click **details icon** in the rightmost column to expand.

Checkbox	Mark log entry for adding to the access policy. To allow further requests based on the information in the selected log entry/entries, select them and click on the **Add selected to ACL** button.

	Note: parameters that are defined as `regexp` in web applications and global policy are not automatically updated to allow new values based on the input from the logged requests. In this case, values need to be updated manually. If adding is not possible the checkbox is inactive.
Time	Date and time the request was logged.
Country	Country the requests originated from.
Host	Hostname from the original request or none if none was present.
Risk	Risk classification of the log entry. Options are: • Critical • High • Medium • Low • None
Source IP	Source IP the request originated from. Click on IP-address to get *whois information*.
Class	Attack classification of the log entry. Options are: • SQL injection • XPath injection • SSI injection • OS commanding • XSS (Cross Site Scripting) • Path traversal • Enumeration • Format string • Buffer overflow • DoS attempt • Worm probe • Access violation • Malformed request • HTML tags • Session invalid • XSRF (Cross Site Request Forgery) • Session expired • Detection evasion • Remote file inclusion • Information leak

	• Backend error • Broken robot • Broken int. link • Broken ext. link • Other • None • False positive • Friendly
Action	Block action taken on the request. Options are: ***Allow*** The request was allowed, either because the current mode and white list configuration or because the requests was allowed according to policy. If the request was allowed by policy the reason for the request being logged in the deny log is typically that the backend server responded with an error. Expand the request to see details. ***Block*** The request was blocked by Web Security Manager. ***Block-IP*** The request was blocked by Web Security Manager and the source IP was blacklisted resulting in further requests from that source being blocked at the network level. ***Strip*** The offending part of the request was stripped before allowing the request. Used for instance to remove session cookies for expired sessions.
URL Path	The URL path requested.
Method Detail - click details to view.	Offending method (if any)
Violation Detail - click details to view.	Shows the general violation description as defined by Web Security Manager. See the list of violations below
Resp. status Detail - click details to view.	If applicable shows the response status from the backend server like `404 not found` or `200 (OK)`.
Resp. time Detail - click details to view.	The time from Web Security Manager received the request and forwarded it to the backend server until the response is sent to the client from Web Security Manager.

Referer Detail - click details to view.	The refering source, internal or external, from which the request originated.
Header Detail - click details to view.	Offending header fields and values (if any).
Query Detail - click details to view.	Offending parameter names and values (if any).
Raw Detail - click details to view.	Shows the original request as send by the client. To view it, click on the **View RAW request** button.

To view all entries in the list expanded click the **Report** button in the lower button bar.

Note

In order not to lock the management interface by returning huge amounts of data a maximum of 500 log entries at a time will be displayed in the interactive log interface.

Use the **XML export** function to download larger lists (or the complete log) for off line analysis and archival purposes.

4.1.2.1. Violations

Content violations

Path unknown	No policy rules allow the path segment of the URL, either because it does not match a positive policy rule or because it matches a negative policy rule - a signature.
Path denied	The path is explicitly denied by an URL blocking policy rule.
Query unknown	No positive policy rules match the name of the request parameter.
Query illegal	No policy rules allow the value of the request parameter, either because it does not match a positive policy rule or because it matches a negative policy rule - a signature.
Session validation failed	The request session ID is not valid, either because the session token has been tampered with or hijacked.
Form validation failed	The form submitted cannot be verified as having been issued by the web application in a response to a request from the current user session. This is an indication of a CSRF attack.
Session expired	The request session has exceeded the idle expiration threshold configured in Web Security Manager for the web application.
Malformed XML	Submitted XML request is malformed and hence cannot be parsed and validated.
Multiple or %u encoded request	The request contains elements that are encoded more than twice or it contains elements that are encoded using %u-encoding.

Authorization failed	User is not authorized to access requested resource.
Header unknown	Request header not RFC 2616 compliant.
Header illegal	Header value failed strict validation.
Header validation failed	Header value failed pragmatic validation.
Output illegal	Server response contains illegal string.

Protocol violations

Generic protocol violation	Protocol violations like missing content length or content type headers for POST requests.
HTTP Protocol version	HTTP protocol version not allowed.
Method illegal	HTTP method not allowed.
Missing hostname	Request does not specify host name.
Invalid hostname	Not website proxy is configured for the requested host name.
Request line maximum length	Entire request line (URI?query) exceeds allowed maximum length.
Request path maximum length	Request path exceeds allowed maximum length.
Query string maximum length	Request query exceeds allowed maximum length.
Content type not enabled	Request content type is supported but not enabled.
Header name length	Header name exceeds allowed maximum length.
Header value length	Header value exceeds allowed maximum length.
Maximum number of headers	Header number exceeds allowed maximum.
Upload attempt	Upload attempted but upload not allowed.
Payload length exceeded	POST payload exceeds allowed maximum size.
Maximum number of upload files	Number of files to upload in a request exceeds allowed maximum.
Total upload size	Total size of upload files in request exceeds allowed maximum.
Maximum file size	Size of a single upload file exceeds allowed maximum.
Cookie version not allowed	Request cookie version not allowed.
Maximum number of cookies	Number of cookies in request exceeds allowed maximum.
Cookie name length	Name of a cookie exceeds allowed maximum length.
Cookie value length	Value of a cookie exceeds allowed maximum length.

Maximum number of GET parameters	GET parameter number exceeds allowed maximum.
GET parameter name length	GET parameter name exceeds allowed maximum length.
GET parameter value length	GET parameter value exceeds allowed maximum length.
GET parameter combined length	Combined length of GET parameter name and value exceeds allowed maximum length.
Maximum number of POST parameters	POST parameter number exceeds allowed maximum.
POST parameter name length	POST parameter name exceeds allowed maximum length.
POST parameter value length	POST parameter value exceeds allowed maximum length.
POST parameter combined length	Combined length of POST parameter name and value exceeds allowed maximum length.
General request violation	Other generic violations.

4.1.3. Lower button bar

The lower button bar contains the following buttons.

Flush log Button	Use with caution! When clicking this button and accepting the confirm pop-up window. *All log data for that proxy will be deleted!*
Log report Button	Generate a printable report based on defined filter criteria (if any).
Add selected to ACL Button	Adds selected log records to access policy.

4.2. Access log

When *access logging* is enabled all requests to the website is logged.

The access log is generated on a per day basis. The current log can be monitored and viewed and closed logs are made available for download.

The fields displayed depends on the selected access log format.

4.3. Access log files

When log files are available for download the filename is an active link. To download an access log file click on the filename.

When remote backup is enabled, the latest access log file made available for download will be compressed (using gzip) and copied to the remote backup destination along with the backup of the system configuration.

Several log file formats are available. A condensed Web Security Manager specific and some standardized formats, like NCSA Combined (Apache Combined), suitable for importing into log analysis and report generation tools.

See Access Log formats for log format definitions.

5. Reports

5.1. Reports

Generation of reports which can be saved or printed for offline viewing.

ACL Report Link	Generate a printable report based on the current ACL. The generated report shows the global URL and parameter settings, static content allowed and individual ACL entries in the database. The report will open a new browser window or tab.
Log Report Link	Generate a printable report with the current log entries. Note that the report is generated according to the current log-filter criteria. The report will open a new browser window or tab.
Log export (XML) Link	Export log to XML. If no filter criteria are specified the complete log will be exported. Depending on log size and filter criteria it may take a while to generate the report. It is therefore generated in the background and made available for download in the Generated proxy log reports section.
Log export (RAW) Link	Raw export of the complete log in sqlite database format. Depending on log size and filter criteria it may take a while to generate the report. It is therefore generated in the background and made available for download in the Generated proxy log reports section.

5.2. Generated reports

Reports generated in the background are made available for download in the *Generated reports* section.

File name Link	Click the file name to download or open the report.
Export date Read only	The date the report was generated.
Status Read only	Status of the report generation. When status is *Done*, the report is ready for download.

Chapter 6

System reference

1. Clustering

Clustering in Web Security Manager is based on VRRP. It allows for configuring high availability WSM pairs running Active/Passive with automatic fail-over within 3 seconds.

When deployed in combination with a load balancer in a separate load balancing pool many WSM nodes can be run Active/Active with the policy synchronized across all nodes by the master.

1.1. Cluster virtual IP configuration

The Cluster virtual IP configuration section allows for adding new virtual interfaces with virtual IP addresses.

It is important that the exact same number of interfaces are configured on the master and slave and that the interfaces are configured in the same order.

Virtual IP	Virtual IP address of the cluster. This is the IP address the nodes in the cluster are sharing.
Netmask	The netmask defining the virtual IP's subnet. The netmask should be the same as the netmask assigned to the IP address of the physical interface to which Inbound Traffic is bound.

Type
Drop down list
	The type of the virtual IP.
	Options
	`FAILOVER MASTER` and `FAILOVER BACKUP`
	Default
	`FAILOVER MASTER`
	To configure a failover IP address, on the master select `FAILOVER MASTER` and on the slave select FAILOVER BACKUP.
	See the examples below for more information.

Interface Drop down list	Which interface to bind the cluster intrface to.. ***Options*** System interfaces ***Default*** First interface in the list

1.2. Synchronization configuration

When Web Security Manager nodes are running a cluster one of the Web Security Manager nodes can be designated the TEACH role and the slave the LEARN role .

In order to keep load balancing and backup nodes up-to-date with the current configuration the TEACHER is keeping the LEARNER updated with changes to configured websites.

To keep the synchronization packages private in the cluster the messages are encrypted using a password as key. Synchronization messages can be sent using either MULTICAST or UNICAST.

Enable proxy settings synchronization Check box	Enable or disable proxy settings synchronization. If enabled, Web Security Manager will synchronize the current ACL database and other parameters with other Web Security Manager nodes.
Mode Drop down list	Synchronization role. If set to `Teach`, this Web Security Manager will multicast the ACL database to other Web Security Manager installations. If set to `Learn`, this Web Security Manager will update it's ACL database according to synchronization messages from other Web Security Manager installations. Synchronization settings affects the operation of the `Learner`. When synchronization is enabled and the node synchronization mode is set to `Learn`, the node will not sample learn data but wait for the node master to dispatch a policy. **Note** You need to configure an interface that will be used for synchronization before the ACL database synchronization will be activated.

Password
Input field
	Password used for synchronization message authentication.
	Valid input
	Any string.
	A long password is recommended as it do not have to be memorable by humans.
	Input example
	`98974953Q38512432324CU4859229842784`
	Default value
	`none`

Protocol
Drop down list
	Synchronization network protocol.
	Options
	`MULTICAST`
	`UNICAST`
	Default
	`MULTICAST`
	The MULTICAST method is selected by default. This method is the easiest to configure but as the name suggests the messages are sent to all nodes within the network and may not always work in complex networks. To keep network traffic at a minimum and to make things work in complex networks UNICAST should be preferred. This method requires the LEARN node to be specified on the TEACH node. When sending

	synchronization messages using UNICAST the TEACHER sends the messages directly to the LEARNERS ip address using UDP.

Sync type
 	
Drop down list
	How websites are synchronized are synchronized in a cluster.
	
	Options
	
	`FULL SYNC`
	
	`TEMPLATE`
	
	Default
	
	`FULL SYNC`
	
	This option applies to learning nodes and controls how websites are synchronized.
	
	FULL SYNC
	
	Everything, including "Listen IP", backend servers and health checking configuration is synchronized.
	
	For HTTPS websites and HTTP websites configured to listen to a specific IP address it is required that the same IP addresses are configured on the learn node - typically in the form of a Cluster IP address configured for high availability or load balancing. Otherwise configuring the proxy core will fail.
	
	TEMPLATE
	
	When new website configuration is received by slave node: All information, including listen IP is included but website is created with disabled status meaning it will not served by the learning node until the website is enabled in ADC : Virtual Host (Section 1, “Virtual host”).
	
	When synchronizing changes Listen IP, backend server configuration, load balancing settings and health checking configuration will not be synchronized. This allows for synchronizing across datacenters or for synchronizing a cluster that is used in combination with a network load balancer.

Peer(s) Input field	The IP address(es) of the other node(s) in the cluster. This input field is disabled if `MULTICAST` is selected. In this case it displays the multicast address which cannot be changed. ***Valid input*** The IP address(s) of the corresponding node(s) in the cluster - i.e. on the TEACHER it should be the LEARNER(s) and vice versa. Note that the IP address should be the IP address assigned to the network interface to which synchronization is bound on the corresponding node. To synchronize to more than one LEARNER node using UNICAST add a list of LEARNER IP addresses separated by comma or space.

	Default value `none`

1.3. Cluster configuration examples

Below are given examples of configuring a high availability cluster running in active/passive mode and a "self load balancing" cluster running in active/active mode.

1.3.1. Configuring a fail-over cluster

To configure a fail-over (active/passive) cluster of two Web Security Manager nodes do the following:

Node 1 configuration	Create a FAILOVER-MASTER interface by doing the following: 1. In **Cluster virtual IP configuration** enter the virtual IP address of the cluster in the the **Virtual IP** field. 2. In **Netmask** enter the `netmask` specifying the subnet for the virtual ip. 3. Select the interface to bind the cluster interface to. If the configured cluster IP is within one of the system interfaces' netmask the system interface with the mask that matches must be selected. 4. In the Type drop-down menu select `FAILOVER-MASTER`. 5. Click the **Add virtual IP** button. Enable cluster synchronization and designate the role TEACH in the **Synchronization configuration** section: 1. Select **Enable proxy settings synchronization** 2. Select `TEACH` in the Mode drop-down. 3. Enter a password for the cluster in the **Password** field. 4. In the **Protocol** drop-down select the default `MULTICAST` 5. Click the **Save** button.
Node 2 configuration	Create a FAILOVER-BACKUP interface for the same virtual IP by doing the following: 1. In **Cluster virtual IP configuration** enter the virtual IP address of the cluster in the **Virtual IP** field. 2. In **Netmask** enter the `netmask` specifying the subnet for the virtual ip. 3. Select the interface to bind the cluster interface to. If the configured cluster IP is within one of the system interfaces' netmask the system interface with the mask that matches must be selected. 4. In the Type drop-down menu select FAILOVER-BACKUP. 5. Click the **Add virtual IP** button.

	Enable cluster synchronization and designate the role LEARN in the **Synchronization configuration** section: 1. Select **Enable proxy settings synchronization** 2. Select `LEARN` in the Mode drop-down. 3. Enter the same cluster password as for node 1for the cluster in the **Password** field. 4. In the **Protocol** drop-down select the default `MULTICAST` 5. Click the **Save** button.

The cluster can also be configured to synchronize and maintain fail-over state using UNICAST targeting a specific peer IP see Section 1.2, “Synchronization configuration” and Section 1.1, “Cluster virtual IP configuration” for more information.

1.4. VRRP Interfaces

The **VRRP Interfaces** configuration section provides an overview of VRRP interfaces and allows for post configuration.

ID	The VRRP interface id on the node.
VIP	Virtual IP address of the cluster. This is the IP address the nodes in the cluster is sharing.
Netmask	The netmask defining the virtual IP's subnet.
VHID Input field	Virtual host identifier number of the VRRP group. On each Web Security Manager node VHIDs are required to be unique. VHIDs identify cluster groups accros Web Security Manager nodes. The same VHIDs are therefore required to be configured on both cluster nodes. ***Valid input*** An even integer in the range 2-254 ***Default value*** `Next available VHID number`
Interface	The physical network interface the VRRP interface is bound to.
State	State of the VRRP interface can be either `MASTER` or `BACKUP`. If a VRRP interface with a low priority (automatically set when selecting the types FAILOVER-BACKUP or LOADBALANCE-FAILOVER) is assuming the role of MASTER then probably the original MASTER node is experiencing problems.
Priority Input field	The priority of the interface in the VRRP group. Do not edit this property unless you are familiar with the VRRP protocol. The priority itself is an abstraction over the `advskev` VRRP parameter. When setting `priority` `advskev` is calculated as `254 - priority`.

	Interfaces of type FAILOVER-MASTER are configured with a high priority and interfaces of type FAILOVER-BACKUP are configured with a lower priority.
	Valid input
	An integer in the range 1-254
	Default value
	`FAILOVER-MASTER: 254`
	`FAILOVER-BACKUP: 154`

1.5. Fail-over status information

If the system is running in a fail-over configuration the following additional information will be displayed.

Virtual IP	Virtual IP address.
Role (config)	Shows the configured role (MASTER or BACKUP) for the specified virtual IP address.
Role (current)	Shows the current role (MASTER or BACKUP) for the specified virtual IP address. If the current role differs from the configured an error situation has occurred and the role information fields will be blinking red.
Interface	Shows the physical interface the specified virtual IP address is attached to.
Priority	Shows the virtual IP address priority for the physical interface.

2. Configuration

2.1. Network

Basic network configuration is performed in this section. Any changes made to this section are applied and saved by clicking on the Save" button.

Hostname
Input field
	Domain name of the Web Security Manager Web application firewall.
	Valid input
	Fully qualified domain name.
	Input example
	`proxy.mydomain.com`
	Default value
	`None`

Default gateway
Input field
	IP address of the default gateway.
	Valid input
	IP address assigned must be in the same network subnet as the IP address of one of the physical network interfaces.
	Input example
	`192.168.0.1`
	Default value
	`None`

DNS server(s)
Input field
	IP address of one or more DNS servers.
	Valid input
	IP addresses
	Use space to separate multiple hosts (only one required).
	Input example
	`192.168.0.2`
	Default value
	`None`

SMTP server Input field	SMTP server hostname or IP address. SMTP server is used for sending alert e-mails to the contact e-mail address specified. ***Valid input*** IP address or fully qualified domain name

	Input example
	`smtp.mydomain.com`
	Default value
	`None`

Syslog server
Input field
	External syslog server hostname or IP address.
	Proxies with external syslog alert enabled will send syslog alerts to the specified server.
	Syslog messages are sent to `user` facility and `informational level` (criticality) is configurable for each proxy.
	Valid input
	IP address or fully qualified domain name
	Input example
	`syslog.mydomain.com`
	Default value
	`None`

2.2. Static routes

Define static routes.

Click **Add new route** and enter route information for each route you want to add.

When routes are entered click **Save settings** in lower button bar to save.

Destination Input field	The route destination. Enter first IP address of destination network. ***Valid input*** A valid ip address. ***Input example*** 1. `192.168.5.0` 2. `192.168.6.8` 3. `192.168.7.10` ***Default value*** `None`
Subnet Input field	Network mask of the destination IP address. ***Valid input*** A valid network mask

	Input example 1. `255.255.255.0` 2. `255.255.255.248` 3. `255.255.255.255` ***Default value*** `None`
Gateway Input field	IP address of the gateway through which the destination can be reached. ***Valid input*** An IP address of a gateway which is directly reachable by the Web Security Manager node. ***Input example*** 1. `192.168.0.4` 2. `192.168.0.5` 3. `192.168.0.6` ***Default value*** `None`

The examples above would result in:

1. Access to IP addresses `192.168.5.0-255` (`192.168.5.0/24`) is routed through the gateway `192.168.0.4`.
2. Access to IP addresses `192.168.6.8-16` (`192.168.6.8/29`) is routed through the gateway `192.168.0.5`.
3. Access to IP address `192.168.7.10` (`192.168.7.10/32`) is routed through the gateway `192.168.0.6`.

2.3. Syslog - logging to external host

Configure threshold level and address of external Syslog server.

Syslog server Input field	External syslog server hostname or IP address. Proxies with external syslog alert enabled will send syslog alerts to the specified server. Syslog messages are sent to `user` facility and `informational level` (criticality) is configurable for each proxy. ***Valid input*** IP address or fully qualified domain name ***Input example*** `syslog.mydomain.com`

	Default value `None`

2.3.1. Mapping of Web Security Manager System Logs to Syslog facilities

Attack	`Local3`
Audit	`auth`
Proxy	`Local0`
Learner	`Local4`
Backup	`Local5`
WebGUI	`Local2`
Daemon	`Local1`
Syslog	Other facilities
Error	All facilities with informational level `error` and above

See Section 5, “Logs” for a description of the log mentioned above.

2.4. SNMP

Configure threshold level and address of external Syslog server.

Enable SNMP queries Check box	Enable or disable SNMP daemon. If checked, Web Security Manager will accept SNMP queries on the first of the IP addresses to which management is bound.

Public community
Input field
	Public community password.
	The read-only community password.
	Valid input
	Any string
	Input example
	`wdbhhaiedb`
	Default value
	`public`

System location
Input field
	Information about the system.
	Valid input
	Any string
	Input example
	`Facility 1, Server room 1`
	Default value
	`none`

Listening on Read only	If SNMP is enabled will display the IP address the SNMP daemon is listening on.

2.5. Date and Time

This section is used for configuration of time synchronization via NTP (Network Time Protocol).

It is strongly advised to configure an NTP server in order to have the correct date and time set on the system.

It is recommended to configure an internal NTP interface. If one is not available, a well-known NTP server time.nist.gov can be used. Also, have a look at www.ntpd.org for a more detailed list of NTP servers available for free on the Internet.

NTP server
Input field
	IP address or hostname of an NTP server.
	Remember to set up at least one DNS server if you enter a hostname here.
	Valid input
	IP address or fully qualified domain name.
	Use space to separate multiple hosts (only one required).
	Input example
	`time.nist.gov`
	Default value
	`None`

Timezone Drop down list	Timezone information. Select the systems timezone from the drop down menu. ***Valid input*** A timezone option from the drop down list. ***Default value*** `Europe/Copenhagen`
Date format Drop down list	Display dates in logs and reports in Month-Day-Year or Day-Month-Year format. Select the date format from the drop down menu. ***Valid input*** An option from the drop down list. ***Default value*** `Month-Day-Year`

2.6. Admin contact

Update notifications, attack alerts and system errors can be sent by email to the admin contact email address.

Contact Input field	E-mail address of the administrative contact. All alert e-mails and notifications are sent to this address. You need to define an SMTP server before any e-mails are sent. ***Valid input*** E-mail address ***Input example*** `admin@mydomain.com` ***Default value*** `admin@mydomain.com`

Sender domain
Input field
	The e-mail address domain.
	If not configured it will be extracted from the contact e-mail.
	Valid input
	a valid domain
	Input example
	`mydomain.com`
	Default value
	`extracted from contact`

2.7. Email system alerts

Critical events or conditions are logged both locally and to external syslog server (if enabled). However if an external syslog server is not available (or is not monitored) a subset of (potentially) critical alerts can be sent to the designated admin contact email.

Email system error messages to admin contact Check box	Enable or disable sending of error messages altogether. If checked, selected alert types will be sent.
Disk and memory Check box	If checked, disk and memory related errors at log level ERROR and CRITICAL will be sent.
Cluster interface events Check box	If checked, cluster interface related errors at log level ERROR and CRITICAL will be sent. The most common cluster interface event is STATE TRANSITION which, when sent by the slave node in a cluster, indicates that the master node

	is either down (backup > master) or has resumed operation (master > backup). When the nodes in a cluster are powered on/off or rebooted state transition messages are also logged to the syslog error log and may generate email alerts.
Administrative daemons Check box	If checked, any error at log level ERROR and CRITICAL from administrative daemons will be sent.

2.8. Forward HTTP proxy

Configure forward proxy to be used by the update system when connecting to the update server.

Use proxy for outbound HTTP Check box	Enable or disable the configured forward proxy.

Proxy address
Input field
	The address of the forward proxy
	Valid input
	A valid ip address.
	Input example
	`10.10.10.5`
	Default value
	`None`

Proxy port
Input field
	Proxy port number
	Valid input
	An TCP/IP port number
	Input example
	`8080`
	Default value
	`none`

Forward proxy authentication required Check box	Enable if forward proxy requires authentication.
Username Input field	User name used for authenticating to the Proxy. ***Valid input*** A valid username ***Input example*** `wsm1`

	Default value `none`
password Input field	Password to authenticate the proxy user.

2.9. Backup configuration

This section is used to configure an FTP/SCP server used for automated configuration backup/restore of Web Security Manager configuration.

2.9.1. FTP configuration

FTP server
Input field
	FTP hostname or IP address.
	Valid input
	IP address or fully qualified domain name
	Input example
	`ftp.mydomain.com`
	Default value
	`None`

FTP port Input field	FTP server port number ***Valid input*** An TCP/IP port number ***Input example*** `21` ***Default value*** `21`

Login
Input field
	Username used for login.
	FTP account used must be able to store files on the remote FTP server.
	Valid input
	A valid username
	Input example
	`wsm_backup`
	Default value
	`none`

Password Input field	Password used for SCP login.

	Valid input
	Any string.
	A long password is recommended as it do not have to be memorable by humans.
	Input example
	`s984ROf.dds&fdsfs)afa8343287`
	Default value
	`none`

Remote directory
Input field
	Full path to directory on FTP server used for storing Web Security Manager related files.
	Valid input
	A directory path ending with /
	Input example
	`/ftp/wsm/`
	Default value
	`none`

2.9.2. SCP configuration

SCP server
Input field
	SCP hostname or IP address.
	Valid input
	IP address or fully qualified domain name
	Input example
	`ftp.mydomain.com`
	Default value
	`None`

SCP port Input field	SCP server port number ***Valid input*** An TCP/IP port number ***Input example*** `21` ***Default value*** `21`
Login Input field	Username used for login. SCP account used must be able to store files on the remote SCP server.

	Valid input
	A valid username
	Input example
	`wsm_backup`
	Default value
	`none`

SCP key Button	Click **Download Web Security Manager Public SCP Key** to download key used for authentication. Make sure to add this key to the authorized keys list on the remote server.

Remote directory
Input field
	Full path to directory on SCP server used for storing Web Security Manager related files.
	Valid input
	A directory path ending with /
	Input example
	`/scp/wsm/`
	Default value
	`none`

Remote directory
Input field
	Full path to directory on SCP server used for storing Web Security Manager related files.
	Valid input
	A directory path ending with /
	Input example
	`/scp/wsm/`
	Default value
	`none`

2.10. Auto-backup

Auto-backup, if enabled, is performed daily at 03:00 AM based on your current timezone settings.

Enable FTP auto-backup Check box	Enable or disable FTP auto-backup. If checked, automated FTP configuration backup will be active.
Enable SCP auto-backup Check box	Enable or disable SCP auto-backup. If checked, automated SCP configuration backup will be active.

2.11. Remote access

The remote support feature allows for configuring Web Security Manager to allow requests from Alert Logic to port 22 on any of the systems ip addresses.

When enabled Alert Logic Support can connect to the underlying OS in order to help diagnose and troubleshoot problems.

Only requests originating from an Alert Logic support IP address will be redirected.

Enable SSH access to management IPs Check box	Enable or disable ssh access to management IPs. If checked, Web Security Manager will allow ssh connections to the same IP addresses as the GUI is bound to.
Enable remote support and monitoring from Alert Logic Check box	Enable or disable remote support access. If checked, requests from Alert Logic to port 22 on any of the systems interfaces will be allowed.

To view detailed settings and verify that remote support is disabled use the **system remotesupport status** command in the CLI (Section 2.22, “system remotesupport”).

If remote support is enabled the system will display a warning on the console when booted.

2.12. Management GUI

Manage password requirements, session and login restrictions and SSL certificate.

2.12.1. Password requirements

Minimum length Input field	Minimum password length in number of characters ***Valid input*** Number in the interval 6 to 64 ***Default value*** 8
Letter characters required Check box	Require one or more letter character, a-z + international.
One or more digits (0-9) required Check box	Require one or more digits.
Combination of upper and lower case required Check box	Require a combination of upper and lower case characters.
Non alphanumeric characters required	Require one or more special (non-alphanumeric) characters.

Check box	

2.12.2. Login and session restrictions

Idle timeout
Input field
	Number of seconds the management GUI can be idle before the user is logged out.
	Valid input
	timeout in seconds 20 to 86400.
	Input example
	`900 - 15 minutes`
	Default value
	`600`

Failed login delay Input field	Number of seconds to wait after a failed login attempt before a new attempt can be made. ***Valid input*** timeout in seconds 0 to 60. ***Default value*** `3`
Failed logins limit Input field	Number of failed login attempts allowed before the failed login action is taken. ***Valid input*** Number of attempts 1 to 100. ***Default value*** `5`
Failed logins action Dropdown	What to do if a user exceeds the failed logins limit. Options: ***None*** No action ***Lockout*** The user account is locked for the configured duration. After the configured the duration the user account is unlocked and the user can log in. ***Suspend*** The user account is suspended and cannot be used until is the account status has been set to OK by an administrator. User account status can be set in System : Users or in the console (Section 2.10, “set user”)

	Valid input
	`None, Lockout, Suspend`
	Input example
	`Lockout`
	Default value
	`None`

Notify user on lockout and suspend Check box	If enabled, user will receive an error message in the login page if the account has been locked or suspended.
Suspend inactive accounts Check box	Enable suspending of accounts that has not been active for a specified duration.
Account inactivity threshold Input field	Number of days a user account can be inactive before it is automatically suspended. ***Valid input*** Duration in days 1 to 1000. ***Default value*** `90`

2.12.3. SSL certificate

Management GUI SSL certificates can either be self signed or imported certificates.

In the SSL certificate section the current SSL certificate in use is displayed. To upload a new certificate click the **Manage GUI certificates** button.

2.12.3.1. Generate self-signed SSL certificate

To generate a self signed certificate enter the certificate information in the input fields.

Click **Save settings** in the lower button pane.

Importing the PKCS12 format

If the certicifate is in the PKCS12 format follow the guidelines below:

1. Enter the path to the certificate file in the **PKCS12 file** input field.
2. Enter Passphrase in the **Passphrase** input field.
3. Click **Save settings** in the lower button pane.

If Validate certificate chain is enabled Web Security Manager will validate and order the chain certificates.

Importing the PEM format

If the certificate is in the PEM format follow the guidelines below:

1. Open the .PEM file in a text-editor. Copy the public certificate section of the certificate.

The public key/certificate is the section of the certificate file between (and including) the certificate start and end tags. Example:

```
-----BEGIN CERTIFICATE-----
 Certificate characters
-----END CERTIFICATE-----
```

2. Select **Import SSL certificate** In the Web Security Manager management interface

   Paste the SSL public key/certificate into the **SSL-certificate** field.

3. Now copy the (SSL) private key section of the certificate. The (SSL) private key is the section of the certificate file between (and including) the private key start and end tags. Example:

```
-----BEGIN RSA PRIVATE KEY-----
 Private key characters
-----END RSA PRIVATE KEY-----
```

4. Enter the passphrase for the private key in the **passphrase field** (if the original private key was encrypted).
5. If a certificate authority chain is provided with your certificate enter the entire list of certificates (more than one certificate may be provided) in the SSL authority certificate(s) chain field

If Validate certificate chain is enabled Web Security Manager will validate and order the chain certificates.

2.13. FIPS 140-2 validated mode

Web Security Manager (WSM) provides the option for the appliance in the customer environment to be locked down to only run the OpenSSL FIPS Object Module in FIPS 140-2 validated mode (FIPS 140-2 certificate #1747).

The lockdown to FIPS 140-2 mode, including validation of the integrity of the FIPS validated crypto modules, is automated, irreversible, and locks down the operating system (CentOS) to run in FIPS 140-2 validated mode as originally specified in the OS provider's (Red Hat Inc.) FIPS 140-2 certificate #1758.

When the option is selected, the applicable package and libraries are converted to FIPS mode, library pre-linking is disabled, and the WSM appliance reboots. After the appliance has rebooted, all communication occurs using only the FIPS-validated algorithms and the appliance will only accept and use FIPS validated encryption for all inbound and outbound communication to and from all services on the appliance. This includes the WAF HTTP proxy service, the appliance's HTTPS web based UI, and SSH services used to remotely access the underlying appliance.

2.13.1. Validation of FIPS mode

When the WSM appliance is using only FIPS-validated encryption modules, the WSM User Interface running on the appliance displays the label **FIPS mode**. The **FIPS mode** label reflects the value of

```
/proc/sys/crypto/fips_enabled
```

Which is computed at startup when the system performs the self tests as required in the FIPS 140-2 certificate Security Policy.

If the self test validation at startup fails the system and crypto modules are not running as required for FIPS validated mode and the **FIPS mode** label is not displayed in the appliance UI.

2.13.2. Enabling FIPS 140-2 validated mode

When enabling FIPS mode:

- The appliance is converted and locked down irreversibly to run in FIPS mode.
- Depending on the disk size the conversion process will take between 2-10 minutes.
- When the conversion process is finished the appliance will reboot.
- During the process proxy services will be available and website availability will not be affected until the appliance reboots.

To enable FIPS 140-2 validated mode

1. Tick the box **Enable FIPS mode**
2. Click the button **Convert appliance into FIPS mode**

   The system will now display a confirmation dialog that outlines the conversion process.
3. Confirm that the appliance is irreversibly converted to FIPS mode.

   The conversion process begins.
4. Log out of the UI and log in again to have the FIPS mode validation label displayed.

   The FIPS mode validation flag that is displayed in the appliance user interface is stored in the currently logged in session in the UI layer so to have the mode validation label displayed immediately after conversion is is necessary to log in again to read the setting from `/proc/sys/crypto/fips_enabled`.

Note

In Amazon Web Services Auto Scaling deployments, the FIPS mode is embedded in the AMI that the auto scaling stack is built from. Consequently, the user interface option to convert the appliance to FIPS validated mode is not available. The configuration of FIPS validated mode and self test at startup to validate are no different from non-auto scaling WSM deployments.

3. Information

3.1. System

Displays basic system hardware information.

CPU	Number of CPUs detected, CPU speed in gigahertz (GHz) and CPU model.
BIOS	Information read from server BIOS.
Memory	Available server (hardware) memory.
Uptime	Time since the system was last started.
Localtime	The system time.

CPU model:

Displays the CPU model.

CPU speed

Displays the CPU speed in gigahertz (GHz).

CPU(s)

Displays the number of CPU(s) detected.

Architecture

Displays the architecture running. Eg. i386/32-bit or amd64/64-bit depending on the Web Security Manager version installed.

3.2. Web Security Manager

basic information about the Web Security Manager platform.

Version	Major and minor version information, e.g. `Web Security Manager-2.2.0-i386`.
Architecture	Architecture running. Eg. `i386/32-bit` or `amd64/64-bit` depending on the Web Security Manager version installed.

3.3. Devices

Detected devices like network interfaces and disk controllers.

Device	The device system id.
Description	Device manufacturer and name.

3.4. Disks

System disks available.

Disk	The disk id, e.g. `sd0`.
ID	Disk identification information
Type	The disk type, e.g. `SCSI`.

Info	Disk information like size, cylinders, etc.

3.5. Currently logged in users

The currently logged in users table displays who is logged in to the system management interface.

Username	The account username.
Remote IP address	The remote IP address of the session.
Last activity	The time when activity was recorded last.

4. Interfaces

Each detected network interface is configured in this section.

Configuration parameters are repeated for each interface allowing system administrator to configure one interface at a time. Parameters like IP, network mask, IP aliases, fail-over, etc. Short description of each detected interface is displayed in the sections.

For example, AMD 79c970 PCI and Intel PRO-1000MT.

4.1. IP configuration

IP configuration specific parameters.

IP address
Input field
	System IP address configured for the interface.
	Valid input
	A valid IP address
	Input example
	`192.168.0.200`
	Default value
	`None`

Netmask
Input field
	Network mask of the interface subnet.
	Valid input
	A valid network mask
	Input example
	`255.255.255.0`
	Default value
	`None`

Description
Input field
	A description of the interface.
	Valid input
	Any string
	Input example
	`proxy interface`
	Default value
	`None`

IP aliases Input field	IP address aliases for the interface. ***Valid input*** IP address separated by new-line

	Input example
	`192.168.0.201, 192.168.0.202`
	Default value
	`None`
	Note
	IP aliases will not be backed up or load balanced in a Web Security Manager cluster. IP addresses which are served by a cluster are configured in Clustering (Section 1, “Clustering”).

4.2. Role

Every configured network interface can be assigned different roles. Depending on the number of network interfaces present, roles should be assigned accordingly. It is recommended to assign a dedicated interface for each possible role.

Network interfaces can be assigned the following roles:

Inbound traffic Check box	Enable or disable inbound traffic for the interface. If checked, the interface (and all IP addresses attached to it) will respond to inbound HTTP/HTTPS requests from clients. If the selected network interface is exposed to clients, this role should be assigned. Default: `<unchecked>` **Note** Web Security Manager will not pass any traffic from clients to back-end servers before at least one network interface is assigned this role.
Synchronization Check box	Enable or disable `Synchronization` for the interface. If checked, the Interface (only it's system IP address) is used for synchronization. Fail-over must be active (on the same or any other network interface) before synchronization is active. Default: `<unchecked>`
Management Check box + input.	Enable or disable `Management` for the interface. If checked, the Interface (only it's system IP address) is used for web-based management. The **Management port** sets the port the management server answers. ***Valid input*** An TCP/IP port number ***Input example*** `8080`

	Default value `2000` Management interface is available via HTTPS/SSL on the configured port. Default: `<checked>`

4.3. Media settings

This section allows system administrator to configure network interface media settings like speed and duplex. Normally, a network interface is set to `autoselect` meaning that the speed and duplex settings are automatically negotiated with the uplink switch.

Media Drop down list	Media settings. Select the media settings from the drop down menu. ***Valid input*** Supported media settings for the interface is displayed in the drop down menu. ***Default value*** `Autoselect`

5. Logs

This section displays various log for major system components. Horisontal menu shows the following log options:

The log views autoupdate regularly.

Attack	Attack notifications sent to local Syslog server.
Audit	Audit log tracking administrative actions.
Proxy	Warnings and errors encountered in the proxy core components
Learner	Messages, warnings and errors from the Learner sub system.
Backup	Warnings and errors encountered during automated configuration backup operations
WebGUI	Warnings and errors encountered in the management interface.
Daemon	admd: Administrative daemon warning and error messages. syncd: Warnings and errors encountered during ACL (access control list) synchronization. Only relevant if clustering or fail-over is configured. alertd: Alert daemon warning and error messages. janitor: Messages, warnings and errors from the Log sub system. statsd: Messages, warnings and errors from the statistics sub system. getupdate: Warnings and errors related to automated patch/version download.
Syslog	Various system log messages.
Error	All messages sent to Syslog with informational level `error` and above.

6. Maintenance

This section contains tools for backup, restore and for maintaining disk space.

6.1. Backup and restore

A backup of Web Security Manager contains the current system configuration including configuration, policy and learn data for all website proxies. When restoring from a backup Web Security Manager will create all interfaces (including cluster interfaces) as specified in the configuration. As interfaces are identified using their id in the system a full system restore can only be performed on a system with the same types of and at least the same number of interfaces as the the system the backup was created on.

6.1.1. Best effort - restoring to different platforms

To restore to a hardware platform that is different from the platform the backup was created on enable **Best effort**. Web Security Manager will then:

- Create interfaces on the target platform in the same order as they appear in the restore file and will try to ignore errors. If the target platform has fewer interfaces than the platform the backup file was created on the excess interfaces will be skipped and the associated IP addresses (including virtual IPs and VRRP cluster interfaces bound to those interfaces will not be created.
- Bind the management role to all interfaces on the target platform in order to make it easy to access the management GUI on the target platform. It is recommended that management is not bound to interfaces which also has the inbound traffic role.

Note that this may result in errors for website proxies that are bound to specific IP addresses so after a best effort restore it is necessary to go through the process of configuring network interfaces and to go through all website proxies to make sure the configuration fits the new platform.

6.1.2. Local backup

When initiating a backup the backup file is stored in the local Web Security Manager file system and will appear in the list of backup files. From here it can be downloaded, deleted and selected for fast local restore.

To initiate a local backup click the **Backup button**.

6.1.3. Restore

Import - File upload	Imports saved system configuration from a file. To import (restore) system configuration previously saved to a file, click on the **Browse** button, select the configuration file and click on the **Upload** button. Enable **Best effort** to skip interface creation and try to ignore errors. Note that this will *overwrite current configuration* including the learn database.
Import - FTP download	Downloads and imports saved system configuration from FTP.

	To import (restore) system configuration from file located on an FTP server, enter the complete path to the configuration file located on the FTP server and click on the Download" button. FTP configuration configured under Auto-backup in **System Configuration**.
Import - SCP download	Downloads and imports saved system configuration from SCP. To import (restore) system configuration from file located on an SCP server, enter the complete path to the configuration file located on the SCP server and click on the Download" button. SCP configuration configured under Auto-backup in **System Configuration**.

6.2. Website templates list

When a template is created form a website proxy it is saved in the local file system. From here it can be applied directly to a website proxy when it is created or later if desired.

The websites templates list displays the website templates available in the system and allows for download and delete of templates of type Custom.

Templates of type Factory contains the default settings available when creating a website proxy and cannot be deleted or downloaded.

6.3. Databases

The databases list displays the databases in use in the system. The flush button will empty the database.

Deny log summary db - all websites	Contains the data displayed in the **Dashboards**+**Deny log** section.
Learn database - all websites	Contains learn data for all websites. Note that when flushing this database from this page **learn data for all websites will be deleted!**. To flush learn data for a specific website select that website, go to the learning data section and click the **Reset learn data** button.
Traffic stats db - all websites	Contains the data displayed in the **Dashboards**+**Traffic** section and in the Statistics page for each website proxy. Note that when flushing this database from this page **traffic stats data for all websites will be deleted!**. To flush learn data for a specific website select that website, go to the traffic statistics section and click the **Clear stats** button.
Website 0 .. n deny log	Each database contains deny log data for a website. The number in the name corresponds to the ID displayed in the left column of the website proxies list in the websites overview page.

6.4. Website access logs list

Displays all access logs currently stored in the system.

Note that access logs can potentially consume a lot of space so it is probably a good idea to download and delete some older access logs.

7. Tools

Tools for operation, maintenance and support.

7.1. Network tools

7.1.1. TCP connect test

Used for network connectivity debugging.

TCP connect test
Input field
	Attempts to establish a connection to the remote on the port specified. If the connection is successfully established, the remote host is considered reachable.
	Valid input
	A valid IP address:port string
	Input example
	`192.168.0.200:80`
	Default value
	`None`

7.1.2. Network debug

The network debug tool runs tcpdump with the selected options. Tcpdump intercepts packages sent to or from the selected interface and writes packet information to a debug log file which can be viewed in the System : Logs section.

The most common use of this tool is to debug connection issues with either clients or backend web servers.

Interface	The interface to intercept traffic to/from.
Packet count	The number of packets to capture. When the selected number of packets are captured the tool will stop.

Source/destination IP
Input field
	Limit packet capturing to a specified source / target IP.
	To debug problems with a backend server enter that servers IP address. To debug client problems enter the IP address of the client.
	This value is optional.
	Valid input
	A valid IP address
	Input example
	`10.10.10.88`
	Default value
	`None`

Port	Limit packet capturing to a specific port number.

Input field	This value is optional.
	Valid input
	A valid port number in the range 1-65534
	Input example
	`443`
	Default value
	`None`

Verbose	Print less protocol information so output lines are shorter.

7.2. Reboot and Shutdown

This section allow the system administrator to restart and shutdown Web Security Manager.

Reboot	Restarts Web Security Manager. Click on the button **Reboot** to initiate a restart. Reboot takes approximately 2 minutes depending on the hardware configuration.
Shutdown	Shutdown Web Security Manager. Click on the button **Shutdown** to initiate a clean shutdown of Web Security Manager™.

7.3. Technical information for support

This section allow the system administrator to view detail information about the current Web Security Manager system status. This information is typically intended for support cases.

Details	Detailed system information including hardware, network settings and running processes. To get the information, click on the **Download** button. You will be prompted to save the file locally on your computer.

7.4. License information

Shows current license information including validity and product type.

Product name	Name of the product license.
Major version	Major product version - e.g. 2.
Serial number	Shows the current applied serial number. This information is important when contacting Alert Logic for technical support.
Apply new key Input field	Apply new license key. Allows the system administrator to apply a new license key.

	Valid input A valid license key ***Default value*** `None` Type in or paste the new license key and click the **Apply** button for changes to take effect.

8. Updates

Web Security Manager checks for available updates every hour and automatically downloads available updates. When updates are available for installation a notification email is sent to the `system contact`.

8.1. Updates available for installation

Displays available Web Security Manager updates that are ready for installation.

ID	Unique update identifier.
Update name	Update name information.
Info	Brief update description.
Size	Size of the update.
Date	Update creation date.
Install Check box	Select update for installation.

8.1.1. Installing updates

To install available updates check the **Install** box right to the particular update on the list and click the **Install** button.

All updates are required and updates can only be installed in sequential order.

8.2. Installed updates

Displays already installed Web Security Manager updates.

ID	Unique update identifier.
Update name	Update name information.
Info	Brief update description.
Date	Update creation date.

8.3. Configuring for updates

In order for the automated update system to work, you need to specify an admin contact email address and configure a DNS-server in:

System → **Configuration**

When updates are ready a notification is sent to the admin contact email address.

Also Web Security Manager needs to be able to initiate the following outbound connections:

updates.alertlogic.com port 80 tcp

Querying of available updates.

updates.alertlogic.com port 8080 tcp

Download of available packages.

Make sure that these connections are allowed in the network firewall.

9. Users

The user section contains tools for user administration.

9.1. User accounts

Web Security Manager has two built in user accounts cannot be deleted. In addition the Administrator account can create and modify administrative accounts for additional users.

9.1.1. Built in user accounts

Web Security Manager has two built-in user accounts which cannot be deleted.

Administrator

The administrator account named `admin` is used for administration of Web Security Manager in the web based management interface.

The administrator account can perform any system or proxy task in the management interface including creating and deleting new administrative users.

Console operator

The console operator named `operator` is the user account used to access the system console CLI. This account can only perform basic system related administrative tasks and only in the system console.

Password for the console operator can only be changed using the command-line (CLI) interface. For information on how to change it, use set password CLI command

9.1.2. Additional accounts

Apart from user administration additional accounts created by the administrator have the same privileges as the built in administrator account.

9.2. Current user

This section allows the current user to change password. Minimum required length is 8 characters. Maximum length is 32 characters.

To change password enter the following information in the password fields:

Old password: enter the old password

New password: enter the new password

Repeat new: repeat the new password

To save the changes, click on the **Apply** button.

9.3. System users

The system users section allows the administrator to add, delete and modify other system users.

To add a new user enter the users information in an empty row and click the **Save changes** button.

User name Input field	The user name the user logs in with. ***Valid input*** Text and special characters.

	Input example
	`fester@somedomain.com`
	Default value
	`None`

Real name
Input field
	The users real name
	Valid input
	Text and special characters.
	Input example
	`Fester Bestertester`
	Default value
	`None`

Password
Input field
	The users password.
	Valid input
	Any string
	Input example
	`fdasdfdaqdbasdas`
	Default value
	`None`
	Note
	The password field is blank for existing users. To change a users password enter a new password in the field.

Add row Button	Click this button to add an empty row if necessary.
Save changes Button	Click this button to save changes to the table.

Chapter 7

The command line interface

Web Security Manager command-line interface is used for initial network configuration and basic network administrative tasks. Rest of the administration is performed using Web Security Manager web-based management interface.

This section provides the information about the command-line interface (CLI) Web Security Manager web application firewalls and how to use the CLI.

1. Accessing CLI

Web Security Manager CLI is available at the console a via SSH.

1.1. Console access

Make sure a screen and a keyboard is properly attached to the system before accessing the CLI.

```
Web Security Manager/i386 (ttyC0)
login:
```

To login, enter your username and password.

Note: The first time you log in to the CLI, use the default username "operator" and the default password "changeme". This should be changed using the set password command.

If the login is successful, you enter the CLI and are presented with a welcome greeting.

1.2. SSH access

If SSH is enabled in the web based administration interface the system can be accessed on port 22 on the same ip addresses as the web based management management interface is bound to.

Connect using an SSH client like Putty (a.o.) and follow the procedure above.

2. Command reference

This section provides detailed description of all available CLI commands.

2.1. show interfaces

To display a list of available interfaces use the **show interfaces** command.

```
psh> show interfaces
em0: Intel PRO/1000MT (82545EM) (00:0c:29:5c:42:82, UP/LINK)
em1: Intel PRO/1000MT (82545EM) (00:0c:29:5c:42:84, UP/LINK)
```

2.2. show interface

To display information about an interface use the **show interface** *`interface_alias`* command.

```
psh> show interface em0
ip: 192.168.0.10
netmask: 255.255.255.0
desc: DMZ interface
```

2.3. show gateway

To display information about the configured hostname use the **show gateway** command.

```
psh> show gateway
gateway: 192.168.0.1
```

2.4. show hostname

To display information about the configured hostname use the **show hostname** command.

```
psh> show hostname
hostname: wsm.lab.alertlogic.com
```

2.5. show routes

To display information about the configured routes and other routing information use the **show routes** command.

```
psh> show routes
Routing tables

Internet:
Destination        Gateway            Flags     Refs     Use    Mtu  Interface
default            192.168.0.1        UGS         0      113      -   em0
127/8              127.0.0.1          UGRS        0        0  33224   lo0
127.0.0.1          127.0.0.1          UH          3    40391  33224   lo0
192.168.0/24       link#1             UC          5        0      -   em0
192.168.0.1        8:0:2b:c3:7f:da    UHLc        2      277      -   em0
192.168.0.9        0:30:5:47:63:34    UHLc        1    15616      -   em0
192.168.0.11       0:d:60:76:7:5f     UHLc        0      553      -   em0
192.168.0.55       0:c:29:5c:42:84    UHLc        0     1512      -   lo0
192.168.0.93       0:d:60:60:2:e9     UHLc        7    81599      -   em0
224/4              127.0.0.1          URS         0        0  33224   lo0
```

2.6. show version

To display the current Web Security Manager version use the **show version** command.

```
psh> show version
version: Web Security Manager 2.8.0-release-i386
```

2.7. set gateway

To configure the default gateway use the **set gateway** *ip_address* command.

```
psh> set gateway 192.168.0.1
```

2.8. set interface

To configure the default gateway use the **set interface** *interface_alias* *ip* *ip_address* *netmask* *netmask* command.

```
psh> set interface em0 ip 192.168.0.10 netmask 255.255.255.0
```

2.9. set password

To configure the console operator password use the *set password* command.

```
psh> set password
Changing local password for operator.
Old password:
New password:
Retype new password:
```

2.10. set user

To set GUI user status use the command.

set user *username* status ok|locked|suspended

```
psh> set user reviewuser status suspended
```

2.11. system backup run

To run configured auto-backup (either FTP or SCP), use the **system backup run** command. This command can be used to force the backup to run on-demand.

```
psh> system backup run
backup started in the background
```

2.12. system cache flush

To remove all cached HTTP resources, use the **system cache flush** command. This command can be used to flush all locally cached documents.

```
psh> system cache flush
flushing document cache in the background
```

2.13. system ping

To send an ICMP ECHO request to a given IP address, use the **system ping** *ip_address* command. This command can be useful for testing network connectivity issues.

```
psh> system ping 192.168.0.1
PING 192.168.0.1 (192.168.0.1): 56 data bytes
64 bytes from 192.168.0.1: icmp_seq=0 ttl=255 time=1.666 ms
64 bytes from 192.168.0.1: icmp_seq=1 ttl=255 time=0.523 ms
64 bytes from 192.168.0.1: icmp_seq=2 ttl=255 time=0.462 ms
64 bytes from 192.168.0.1: icmp_seq=3 ttl=255 time=0.506 ms
64 bytes from 192.168.0.1: icmp_seq=4 ttl=255 time=0.421 ms
--- 192.168.0.1 ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/std-dev = 0.421/0.715/1.666/0.477 ms
```

2.14. system updates fetch

To force a up to date check on new available updates, use the **system updates fetch** command.

```
psh> system updates fetch
fetch started in the background
```

2.15. system updates query pending

To display pending updates, use the **system updates query pending** command.

```
psh> system updates query pending
AL-PF-1.2.4-i386, Performance improvements and feature updates
```

2.16. system updates query installed

To display installed updates, use the **system updates query installed** command.

```
psh> system update query installed
AL-PF-1.2.2-i386, Cache module configuration update
AL-PF-1.2.3-i386, Stability/security updates and improvements
```

2.17. system updates install

To install a pending update, use the **system updates install** *update_id* command.

```
psh> system updates install AL-PF-1.2.4-i386
done
```

2.18. system status

To display the system status use the **system status** command.

```
psh> system status
application server (as): OK (pid: 5958)
management interface (mi): OK (pid: 1768)
core components (cc): OK (pid: 7058)
rule daemon (rd): OK (pid: 20772)
sync daemon (sd): OK (pid: 2620)
```

2.19. system restart

To restart system components use the **system restart** *component* command.

Available components are:

as

Application server

mi

Management interface

cc

Core components

rd

Rule daemon

sd

Synchronization daemon

```
psh> system restart as
done
```

2.20. system shutdown

To shutdown Web Security Manager use the **system shutdown** command.

```
psh> system shutdown
```

2.21. system reboot

To reboot Web Security Manager use the **system reboot** command.

```
psh> system reboot
```

2.22. system remotesupport

To view status, enable and disable remote support (Section 2.11, “Remote access”) use the **system remotesupport** command.

```
psh> system reboot
```

2.22.1. View remote support status

To see the current status of remote support (i.e. are requests from Alert Logic being redirected from port 80 to port 22 enter **system remotesupport status**.

When remote support is enabled:

```
psh> system remotesupport status
Current remote support setting: Enabled
pf Status: Enabled for 0 days 00:00:11           Debug: Urgent
pass in inet proto tcp from 130.226.138.37 to any port = ssh flags S/SA keep state
rdr inet proto tcp from 130.226.138.37 to any port = www -> 127.0.0.1 port 22
```

When remote support is disabled (default):

```
psh>system remotesupport status
Current remote support setting: Disabled
pf Status: Disabled for 0 days 00:00:05          Debug: Urgent
```

2.22.2. Enable remote support

To enable remote support (i.e. allowing access to port 22 from Alert Logic) enter **system remotesupport enable**.

```
psh> system remotesupport enable
pf enabled
remote support set
Current remote support setting: Enabled
pf Status: Enabled for 0 days 00:00:00           Debug: Urgent
pass in inet proto tcp from 130.226.138.37 to any port = ssh flags S/SA keep state
rdr inet proto tcp from 130.226.138.37 to any port = www -> 127.0.0.1 port 22
```

2.22.3. Disable remote support

To disable remote support (i.e. disallowing access to port 22 from Alert Logic) enter **system remotesupport disable**.

```
psh> system remotesupport disable
pf disabled
remote support set
Current remote support setting: Disabled
pf Status: Disabled for 0 days 00:00:00          Debug: Urgent
```

2.23. quit

To quit the Web Security Manager CLI session, use the **quit** command.

```
psh> quit
```

Chapter 8

Network deployment

1. Simple single-homed Web Security Manager implementation

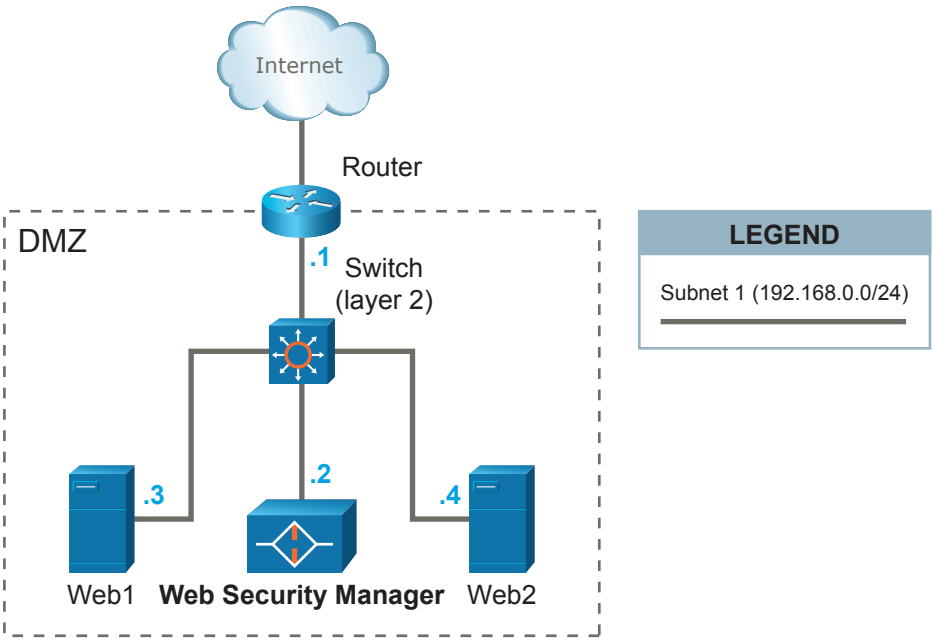


Figure 8.9. Simple single-homed Web Security Manager implementation

This scenario is the easiest to implement, since Web Security Manager can be introduced in the already established network without any major reconfigurations. A caveat with this setup is that all Web Security Manager traffic (both inbound from clients and outbound to the web systems) is using a single ethernet interface.

Web Security Manager is placed on the same network (DMZ) with the web systems web1 and web2 it is protecting.

HTTP/HTTPS traffic designated to the web systems (`192.168.0.3` and `192.168.0.4`) is redirected (either by forwarding IP packets via the router or by altering web systems' DNS settings) to Web Security Manager's IP address `192.168.0.2`.

The web systems' default gateway is unaltered and is still the router with IP address `192.168.0.1`.

2. Firewalled single-homed Web Security Manager implementation

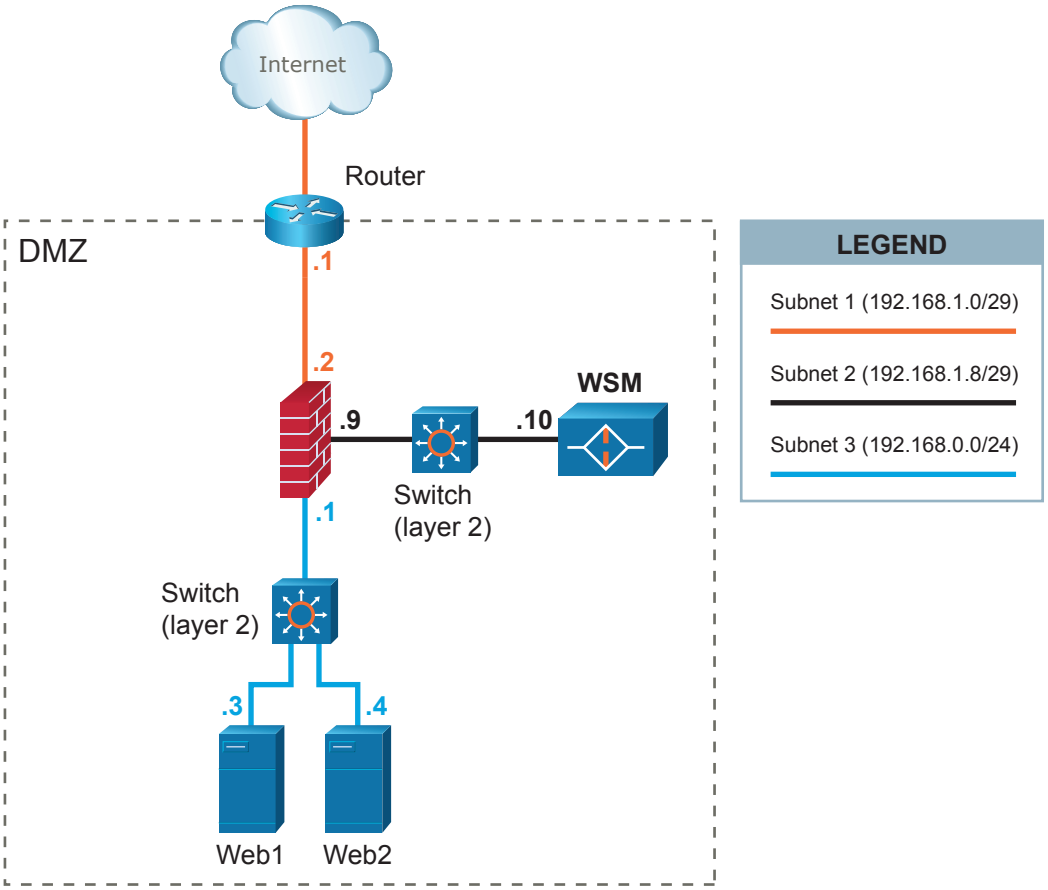


Figure 8.10. Firewall'ed single-homed Web Security Manager implementation

This scenario requires an extra interface in the firewall since Web Security Manager is deployed in a DMZ-segment separated from the segment in which the web servers are placed. A caveat with this setup is that all Web Security Manager traffic (both inbound from clients and outbound to web systems) is using a single ethernet interface.

A separate network segment (`subnet 2`) is configured between Web Security Manager and the firewall.

HTTP/HTTPS traffic designated to the web systems (`192.168.0.3` and `192.168.0.4`) is redirected (either by forwarding IP packets via the router or by altering web systems' DNS settings) to Web Security Manager's IP address `192.168.1.10`.

Outbound traffic from Web Security Manager to web systems is again inspected by the firewall and sent to the web systems on `subnet 3`.

The web systems' default gateway is the firewall with IP address `192.168.0.1`.

3. Firewalled Web Security Manager implementation with a fail-over/backup Web Security Manager

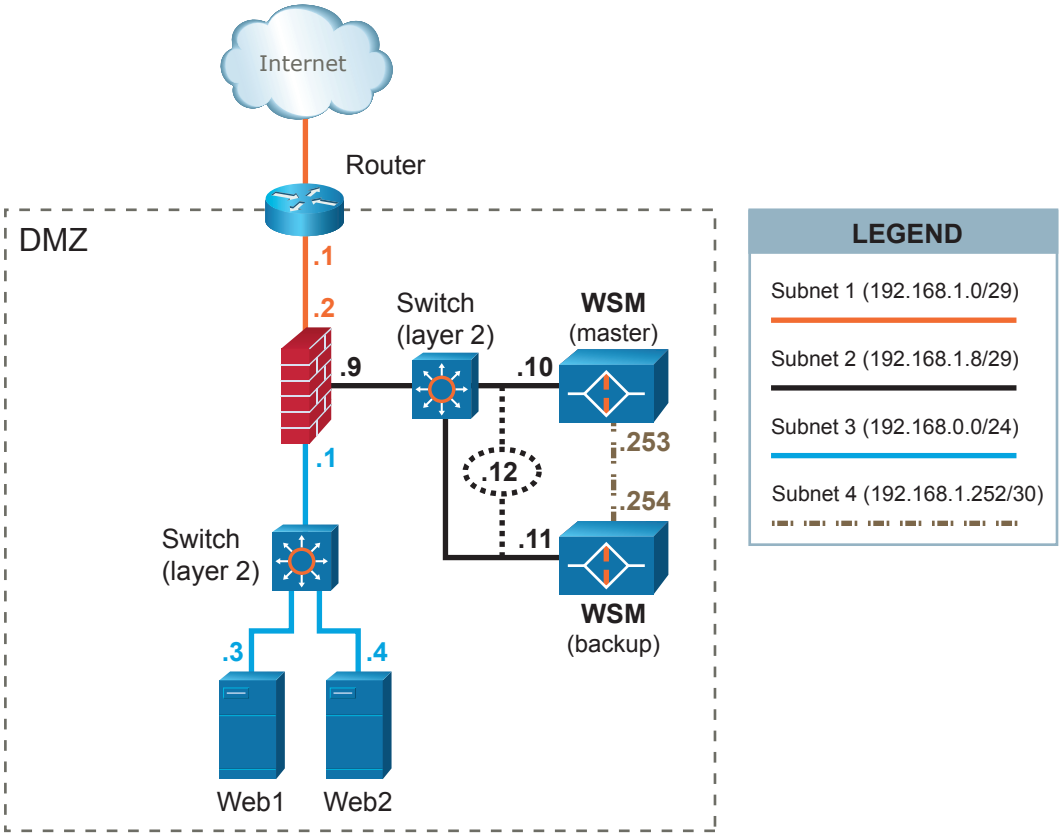


Figure 8.11. Firewalled Web Security Manager implementation with a fail-over/backup Web Security Manager

In this scenario Web Security Manager is deployed in a high avalibility configuration with an extra Web Security Manager (backup) used for fail-over. A dedicated network or crossover cable is used to connect the Web Security Manager cluster and a separate interface is used for synchronization of various information between the active and the backup Web Security Manager. Inbound and outbound traffic share the same interface.

The two Web Security Manager systems share a virtual (VIP) IP address `192.168.1.12`.

HTTP/HTTPS traffic designated to the web systems (`192.168.0.3` and `192.168.0.4`) is redirected (either by forwarding IP packets via the router or by altering web systems' DNS settings) to Web Security Manager's VIP address `192.168.1.12`.

In case the active Web Security Manager system fails or looses the connectivity, the backup will take over the VIP and start handling the requests from clients.

The web systems' default gateway is the firewall with IP address `192.168.0.1`.

4. Dual-homed performance optimized Web Security Manager implementation

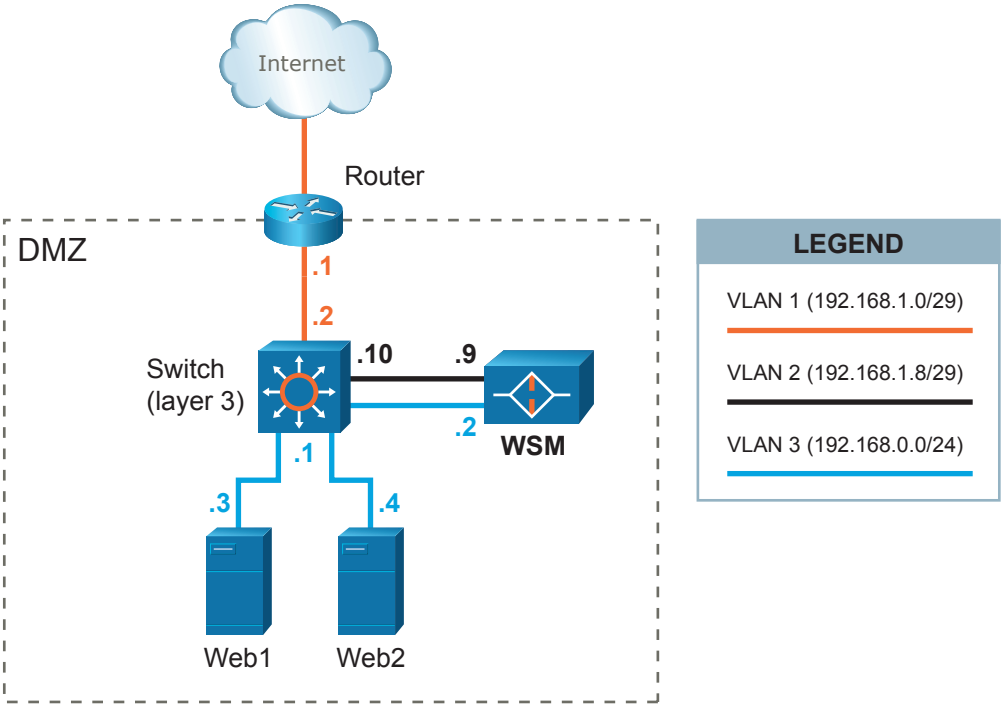


Figure 8.12. Dual-homed performance optimized Web Security Manager implementation

In this scenario Web Security Manager is configured in a dual-homed setup with separation of inbound and outbound web traffic. 2 ethernet interfaces are utilized. Client requests are terminated in VLAN2 and responses from web systems are terminated in VLAN3. This setup (or similar) potentially provides greater performance (since 2 interfaces are used) and security.

A separate network segment (VLAN2) is configured between Web Security Manager and the layer 3 switch.

HTTP/HTTPS traffic designated to the web systems (`192.168.0.3` and `192.168.0.4`) is redirected (either by forwarding IP packets via the router or by altering web systems' DNS settings) to Web Security Manager's IP address `192.168.1.9`.

Outbound traffic (downstream) from Web Security Manager is sent to web systems via `VLAN3`.

The layer 3 switch is configured only to allow traffic on the necessary ports (typically `80/tcp` for HTTP and `443/tcp` for HTTPS to pass from Web Security Manager to the web systems.

The web systems' default gateway is the layer 3 switch with IP address `192.168.0.1`.

Chapter 9

Frequently Asked Questions

1. Deployment

1.1. Network deployment

Where in the network is Web Security Manager deployed?

Web Security Manager web application firewall is installed in the network between the network firewall and the web server. It is a filtering reverse proxy that terminates all client requests validates them and, if benign, re-issues the requests to the protected web servers on behalf of the clients.

This means that the client (from the Internet) sees Web Security Manager as the web server that serves requests to the protected web site and the protected web server only communicates directly with Web Security Manager.

2. Client issues

2.1. Client IP address appears not to be available to backend webserver

Our web application is using the client IP for geo location and site statistics. After putting Web Security Manager in front of the website the web application only sees the IP address of Web Security Manager. Is there any way we can forward the client IP address to the backend web server?

Yes. As Web Security Manager is a reverse-proxy it terminates requests from clients and makes the request to the backend webserver on behalf of the client. That makes the source IP (of the client) disappear from the underlying IP packets. However, the original source IP address is forwarded to the backend server in a HTTP header. The header is called "X-Forwarded-For". Your application (or webserver) can be configured to use the information in the header instead

if you are running an IIS server and you want it to log the client IP try Googling for "IIS X-Forwarded-For ISAPI filter".

3. SSL Certificates

3.1. Wildcard SSL certificates

Do you support the use of wildcard SSL certs?

Yes, you can use a wildcard SSL cert, but you will have to apply it for all proxies or make one proxy with a number of aliases (in proxy manage->settings->servers (like entering *.alertlogic.com in the alias list to match a *.alertlogic.com wildcard certificate).

3.2. SSL certificate update

How do I update the SSL Certificate?

Updating SSL certificates can be done in several ways. The best approach is to have the CA generate a PEM certificate (it will do that when you select Apache as the destination server).

When updating certificates, simply copy and paste the content (including ---BEGIN CERTIFICATE--- and ---END CERTIFICATE--- lines) of your renewed certificate into the correct Public key field for the Web Security Manager proxy in question. Do the same for the private key (which you should have when you generated a CSR file). Copy and paste that?? in the Private key field and press save. Make sure to type in the pass phrase in the Passphrase field if your private key is encrypted.

Web Security Manager will then replace the copy of your Public and Private key with the new copies.

4. Troubleshooting

4.1. Database locked error messages in System Error log

We are getting a lot of database locked errors in the system error log and Web Security Manager is very slow.

Make sure Hyperthreading is disabled on platform on which Web Security Manager is installed. Start by checking the number of CPU's reported in System : Information. If the CPU count is doubled Hyperthreading is probably enabled. Disabling it will fix the problem.

5. Clustering

5.1. Clustering not working in VMware ESX

We have configured a cluster IP address in Web Security Manager running on VMware ESX but the IP address is unreachable.

Make sure `promiscuous mode` and `MAC address changes` are allowed on the VMware virtual switch (vswitch) or the port group in the VMware ESX network configuration.

6. Accessing Web Security Manager management interfaces

6.1. How do I login to the console?

username: operator

password: changeme

When logging in to the console as the `operator` user the shell will be the Web Security Manager command-line interface (CLI).

The Web Security Manager CLI is used for initial network configuration and basic network administrative tasks. All actions that can be made in the CLI also available in the web based management interface. All licenses, including trials, have access to the Web Security Manager CLI.

Remember to change the default password using command 'set password'.

Note: This is not the management GUI.

6.2. How do I access the underlying OS - not the CLI?

OS access is only available in non-trial Web Security Manager installations.

When a non-trial license key is applied the root password is set to the license key. Remember to change it.

In the console CLI issue the command 'enable' and provide the root password.

To use SSH, enable SSH access to management interfaces in System : Configuration.

Connect to port 22 using an SSH terminal program like Putty.

To change root password issue the command 'passwd' when logged in to the OS.

6.3. I cannot access the management GUI on HTTPS port 4849

If you are accessing Web Security Manager through a network firewall it may be blocking HTTPS on this port.

Change the management port by accessing the GUI from a node within the same network segment as the Web Security Manager node.

In the GUI go to in System > Interfaces and change the listen port in the management role.

7. Learning

7.1. Learner not learning from test requests

Learning is enabled for the website but Web Security Manager is not learning anything. We have run a vulnerability scanner on the proxied website. The scanning included a complete site crawl but nothing was learned.

The Learner learns from input, that is: from un-trusted sources. In order to avoid an attacker polluting the policy by hitting the website with something automated, thresholds have to be reached for the policy to be generated. On top of that - if you have performed any blocked requests which have classified as known attack types (DoS, SQL Injection, XSS, etc.) then the Learner will not learn anything from your IP-address. The address has been banned and marked as hostile.

To disable IP banning select "Learn from hostile sources (IPs)" in **Services** → **Websites** → **Learning** → **Learning settings**.

Also the learner will look at factors like spread in time, sources and number of hits. In **Services** → **Websites** → **Learning** → **Learning settings** click "help" in the upper right corner to get the relevant section of the manual.

To configure the Learner to learn from anything, right away, configure it with very low thresholds in Services > Websites > Settings->Learner.

For example: Consecutive sample status updates: 1 - Learning thresholds all set to 1.

Generate traffic to the site using for instance Microsoft Web Application Stress Tool which is freely available on the MS web site.

Do *not* use the generated policy for production unless the test cases used are representative of real life requests. It will almost always result in false positives when the website proxy meets "real life". We recommend deleting all learned data and letting Web Security Manager learn from real life requests when it is deployed into the production environment.

8. Filtering

8.1. Requests are blocked but not logged

I have installed Web Security Manager in a test environment and created a website proxy for our test website, but my requests are not getting through.

I get a “404 Not Found” message from Web Security Manager but the denied requests do not appear in the deny log of the website proxy.

The most common reason for blocked requests not showing in the website deny log is that they are logged in the Web Security Manager Generic website proxy (ID 0 in the overview). This is because the request is targeting an unknown host or is malformed in some other way.

For example, when you bind inbound traffic to an interface with IP address 192.168.10.10 and you configure Web Security Manager to proxy requests for www.mydomain.com and then try to access the website by entering the IP address (192.168.10.10) in the browser address field, the request will be rejected because the browser sends a request for the virtual host 192.168.10.10 not the virtual host www.mydomain.com you just configured.

If you want Web Security Manager to respond to the IP address (which is practical for test purposes but definitely not recommended for production), add the IP address to the virtual host aliases list in **Services** → **Websites** → **ADC** → **Load balancing**+**Virtual host aliases**.

8.2. NTLM authentication

We have a MS IIS based site using NTLM authentication. When requests are proxied through Web Security Manager it does not work. How can we make it work?

Enable "Add HTTP/1.1 VIA header information" in **Services** → **Websites** → **ADC** → **Load balancing**. This should do it.

Microsoft IIS is configured not to allow NTLM authentication if a request is coming from a proxy server (as with Web Security Manager). If it sees a NTLM request coming thru a proxy server, IIS will subvert to other authentication methods, such as Basic of Digest.

If "Add HTTP/1.1 VIA header information" is enabled Web Security Manager will insert a Via: header in forwarded requests which will inform the back-end server that the request is proxied.

8.3. Redirecting from HTTP to HTTPS

When our visitors request a resource in a specific path (like /secret/someapp.php), how do we redirect them to the same path on a corresponding HTTPS site.

Enter a redirect rule in **Services** → **Websites** → **ADC** → **Virtual host**.

For example: To redirect visitors requesting http://www.mydomain.com/secret/someapp.php to https://www.mydomain.com/secret/someapp.php enter the following redirect rule:

Match Type: "prefix"

Match: /secret/

Redirect externally to: http://www.mydomain.com/secret/

This will redirect any request for resources in /secret/ to the HTTPS site.

Note that the protocol and server address is required in the redirect to field. It can be any address including https://www.myotherdomain.com/secret/deep/in/the/directory/tree/ in

which case a request for http://www.mydomain.com/secret/someapp.php will be redirected to https://www.myotherdomain.com/secret/deep/in/the/directory/tree/someapp.php.

More advanced redirects are available using regular expressions see *redirect* in the manual or click *help* in the upper menu when you are on the server page.