

jetNEXUS Application Firewall

Protect and Secure your Web Application with the Next Generation in Enterprise Firewalls



What is the jetNEXUS Application Firewall?

JAF is the next generation Enterprise Web Application Firewall with attack detection and attack protection functions that are continually updated and customisable.

JAF enables centralised monitoring, reporting and alerting and provides custom protection for your Web applications against external attacks.

With its unique clustering architecture, JAF will scale in performance to even the largest web infrastructures.

Basic Principle

The jetNEXUS Application Firewall (JAF) can either run on the jetNEXUS Enterprise Traffic Manager or be directly installed onto a web server. In its detection mode, incoming and outgoing requests are checked based on the various security policies. When a ruleset is activated and the policy is infringed, the query is rejected and not passed to the Web application on the Web server.

JAF checks not only a request's transport layers, but the entire logic. Unlike conventional firewalls, it is possible to customise the protection specifically for the logic in your Web application. This is the only way in which you can block all unwanted traffic without also interfering with the traffic that you want to receive.

In its detection mode, JAF enables your Web applications' security status to be monitored transparently. By activating a ruleset, events can also be responded to immediately. JAF's architecture also enables rulesets to be tested on individual Web servers in a cluster without the entire application being subjected to the policy.

Assuming a suitable custom configuration, attacks no longer penetrate to your Web Application. If, despite all cautionary measures, an attacker succeeds in accessing security-related data, JAF also analyses your Web application's responses and, e.g., deletes credit card numbers from them before they exit your Web server.

It is possible to manually configure JAF in great detail or you can use various automated configuration algorithms and pre-prepared blacklists. JAF can also continually analyse your Web application's behaviour over time and, based on this, independently improve the protection.

While protecting your Web application, JAF logs and documents all the actions, requests and attempted attacks in a database.

Should specific events occur, actions are triggered and relevant individuals can be automatically notified.

JAF runs invisibly, without its own IP address, so it is protected from any direct attack.

Deployment in the Enterprise Environment

JAF is designed to scale in Enterprise environments. If you are running multiple Web applications, their protection can be administered separately by different individuals. If it is installed in a cluster, JAF will scale in performance to support the number of Web servers.

Summary

JAF is an enterprise Web application firewall of the latest generation and fulfils the following basic tasks

- Intrusion Detection
- Security Monitoring, Alerting and Reporting
- Protection when a Ruleset is activated

What are the benefits to your company?

With JAF:

- You protect against zero day attacks
- You detect known attacks
- You repel attacks and eliminate unwanted traffic
- You document attacks and defensive measures
- You satisfy compliance requirements (legal obligations, industry standards, service level agreements)
- You can safeguard your corporate image, sensitive data and trust relationships.

You protect against zero day attacks

Many vulnerabilities are often unknown and it is just a question of time before they are discovered and exploited. When protection is activated, one of JAF's main basic principles is: "Anything that is not expressly permitted is not permitted." In this way you also protect against such zero day attacks.

You protect against known vulnerabilities in the short term

Many Web applications use third party components or are based on frameworks that cannot immediately be maintained. For cases where source code is unavailable or the patches which would conflict with the maintenance process occur. A Web application firewall such as JAF is the only protection option.

But even if you have developed a Web application all by yourself, you will not always have the human resources to close off vulnerabilities in the source code at short notice.

You detect attacks on your Web applications

While providing protection against attempted attacks, JAF also serves as an intrusion detection system (IDS).

You satisfy compliance requirements

With JAF, you can continually assess and document which attempted attacks have actually been made on your Web application and which security measures are countering them.

You also eliminate other unwanted traffic

With JAF, as well as protecting against explicit attacks, you can also eliminate all other types of unwanted traffic on your Web application. Examples include deep linking, access via certain referrers, access from specific regions, at specific times, by specific robots and by your competitors. Not always does your access to your application have to be completely blocked. You can also simply restrict access to individual parts of your application, deliberately generate certain HTTP error messages, or re-route to a particular page.

What are the benefits to you Personally?

With JAF:

- You document your security measures for superiors
- You can also close off key security holes at short notice

You can show that you have complied with your duty of care

The documentation of security measures you have taken is not only important to your company. Just imagine the worst case scenario. Sensitive data (for example customers' credit card numbers and addresses) somehow leaves your company. The reporting of JAF documents loss of data conclusively assuring your responsibility for all Web applications.

- You stay at the cutting edge of technology thanks to regular software updates and blacklists

- You get information about possible attacks without interruption

As a result

- You can provide evidence that you have complied with your duties of care

- You are under no time pressure when security loopholes are detected

Thanks to JAF's reporting features, you can fully document the security measures you took in the past and that you complied with your duty of care. When you use JAF it is most likely that you will be able to prove from the log files that the Web application you are responsible for was not the security loophole.

You can close off security loopholes with no time pressure

With JAF, you can immediately and with minimal effort safeguard the vulnerabilities you discover in your Web applications ("external patching"). You can then remove the cause of the problem with no time pressure and thoroughly test the application before the new version goes online.

You remain at the cutting edge of technology

Updates of the rules used in automatically generated rulesets form part of your support contract as well as regular software updates. jetNEXUS specialists will provide you with professional support on specific issues wherever necessary.

What input does the system need?

The configuration is all done via an easy-to-use, web-based user interface. To get automatic basic protection, you just need to create your Web application in Basic Mode and then run a wizard.

Then, in Expert Mode, you can customise and fine-tune all the settings in great detail where necessary.

This procedure can also be automated, so it can be integrated as a fixed part of your workflow.

System Components

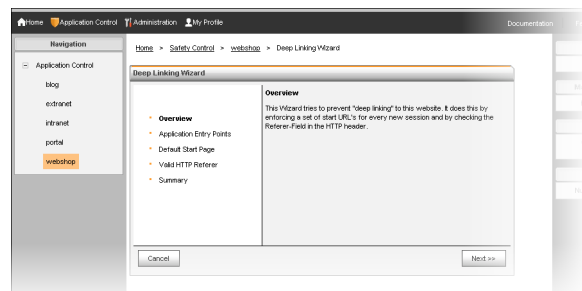
JAF consists of three central components:

The **Administration** interface is a web-based user interface for managing the security configuration and for accessing log files and statistics.

The **Enforcer** traps each HTTP request and passes it to the **Decider** for further examination. The Enforcer then implements the decisions made by the Decider: the HTTP request is either accepted, modified or rejected.

The Decider is a service that evaluates HTTP requests using a ruleset stored in a configuration database and makes decisions about the actions that should take place. When installed in a cluster, a master XML server runs on the master. It receives the commands from the Administration interface and takes care of administering the slaves. This particularly includes querying the availability of the slaves and updating the slaves with new configurations. A slave XML server runs on the slaves, which receive control information and new configurations from the master. Similarly, there is a Decider that runs on each slave to evaluate each request.

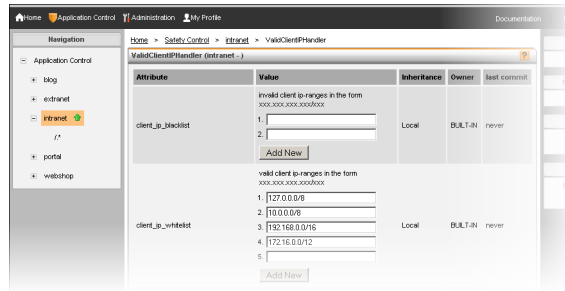
Basic Mode and Expert Mode



The configuration is all done via an intuitive, web-based user interface on which you can switch between a simple Basic mode and a powerful Expert mode.

In the Basic Mode you are helped in your configuration efforts by wizards and

intelligent learning algorithms. A few key basic inputs suffice, and JAF automatically configures all the required internal handlers.



Handlers are the Decider's program routines that examine requests using the rules stored in JAF. In the Expert Mode, you can customise the parameters of each handler in detail if necessary. In many cases you can also work with regular expressions.

A typical scenario: Combining automatic and manual configuration

You can get JAF to automatically propose rules and you can create rules yourself. A tried and trusted method is to first have rules suggested automatically. Then, in a second step, you can manually fine-tune or add to these rules as you require.

For some automatic rules, JAF first studies and analyses the behaviour of your Web application for a certain period of time. However, if you so wish, you can get basic protection straight away.

What output do you get?

JAF as an attack detection system:

- Logs all requests
- Generates detailed statistics and reports
- Issues an alarm if certain events occur
- Logs all changes to the configuration

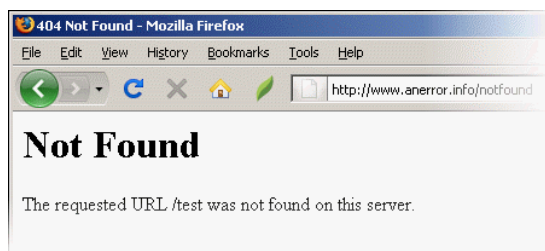
JAF with activated protection:

- Rejects non-permitted requests with an HTTP error code
- Deletes sensitive data from responses
- Logs all requests and the response to each request

JAF as an attack detection and protection system evaluates all requests

At runtime, JAF receives and analyses each request before it reaches your Web application. In so doing, every request is assigned to one of these classes:

- Legitimate requests are passed to the Web application.
- Definite attacks are repelled when protection is activated and, at the same time, as much data as possible is saved in order to identify and trace the attacker.
- Requests whose danger cannot be definitively assessed at local level can either be rejected or passed on, depending on the local rating and installed security policy. They are also logged internally and used to then classify subsequent requests.



JAF also analyses the responses

In the other direction, JAF also analyses your Web application's responses. Security-related information such as credit card numbers can thus be filtered out

from the responses and does not escape even when the attack is successful. As it continues to analyse your Web application over a period of time, JAF gathers key information about its behaviour and uses it to further optimise the protection.

JAF fully logs changes to the security configuration and all requests

The recording and analysis of access and server data is important for several reasons:

- You can see where possible attacks are being attempted, and so further optimise your counter-measures in the future.
- You can see where security measures may be too restrictive, possibly limiting certain users unnecessarily.
- You comply with possible legal or contractual rules relating to the duty of keeping records.

In JAF, you can access these records:



■ Statistics

- Statistics presented as graphics show the chronological distribution of accepted and rejected requests depending on the individual handlers.

- Special cluster slave statistics show the load and status of individual cluster slaves.

■ Log files

- Log files contain host-specific logs of all internal system events and error messages.

- The default error log records events that do not affect a specific Web application. These include, e.g., invalid requests or requests whose given host name does not match any of the hosts configured under JAF.

- An audit log contains a list of all the administrators' actions.

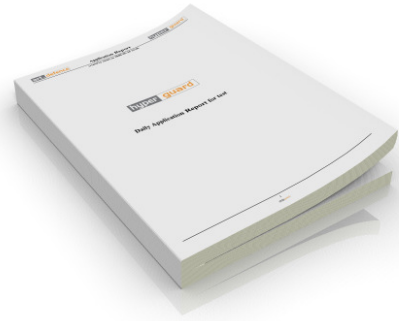
- An event log shows a table of all the status changes and events.

The screenshot shows the 'Logfiles' view in the JAF interface. It includes a table with the following columns: Timestamp, Session, Cluster Member, Host, Remote Address, and Log Entry. The table contains several rows of log data, including entries for denied requests and successful GET requests.

Timestamp	Session	Cluster Member	Host	Remote Address	Log Entry
15.04.2008 17:27:40		127.0.0.1-9086	localhost	127.0.0.1	GET /Avicon.sco HTTP/1.1 DenyHandler (SHADOW) denied request
15.04.2008 17:27:40		127.0.0.1-9086	localhost	127.0.0.1	GET /Avicon.sco HTTP/1.1 OK
15.04.2008 17:02:25		127.0.0.1-9086	localhost	127.0.0.1	GET /Avicon.sco HTTP/1.1 DenyHandler (SHADOW) denied request
15.04.2008 17:02:25		127.0.0.1-9086	localhost	127.0.0.1	GET /Avicon.sco HTTP/1.1 OK
15.04.2008 17:02:29		127.0.0.1-9086	localhost	127.0.0.1	GET /Avicon.sco HTTP/1.1 DenyHandler (SHADOW) denied request
15.04.2008 17:02:29		127.0.0.1-9086	localhost	127.0.0.1	GET /Avicon.sco HTTP/1.1 OK
15.04.2008 17:02:26		127.0.0.1-9086	localhost	127.0.0.1	GET /Avicon.sco HTTP/1.1 DenyHandler (SHADOW) denied request
15.04.2008 17:02:26		127.0.0.1-9086	localhost	127.0.0.1	GET /Avicon.sco HTTP/1.1 OK
15.04.2008 16:52:04		127.0.0.1-9086	localhost	127.0.0.1	GET /Avicon.sco HTTP/1.1 DenyHandler (SHADOW) denied request
15.04.2008 16:52:04		127.0.0.1-9086	localhost	127.0.0.1	GET /Avicon.sco HTTP/1.1 OK

■ Reports

Application-specific, consolidated reports provide a printable summary of the current configuration and the events that have occurred recently in the PDF format.



■ Key Features

- Despite the complex subject matter, administration is simple in the Basic Mode
- Wide range of functions in the Expert Mode; Python API
- Automatic basic protection or finely-granulated custom settings possible
- Automated rule-making based on the results of source code and vulnerability scans done on a Web application
- User friendly administration
- Cluster architecture
- Integrated version management

■ Administration and Configuration

- Integration into existing infrastructure as a software plugin. Existing structure and security policy need not be altered, though later changes are possible.
- Ergonomic administration via web-based user interface
- Download ready-made rulesets
- Configuration is assisted by intelligent learning algorithms. Service for analysing log files and automatically generating application-specific rule proposals.
- Automatic configuration for protecting Microsoft Outlook Web Access (OWA)
- ModSecurity rulesets can be imported
- Any number of hosts can be managed for each application
- Central administration of distributed installations (clusters)
- Role based management of multiple Web applications (client capability)
 - Central basic configuration and management
 - Support for different administrators
 - Personalised authorisations
 - Complete configuration history and audit log
- Prefixes for separately handling special file types and directories
- Preconditions to simplify configuring the rules for individual prefixes
- Object orientated inheritance mechanisms
- Regular expressions supported
- Integrated version management: all earlier versions can be re-edited and activated at any time.
- Export and import functions for more easily moving rules from a test system to a live system.

Protection

- Bi-directional HTTP request analysis
- Protection level can be customised to the risk level of the Web applications being protected
- Protection against all common attack patterns
- Regular updates mean that the protection is continually and automatically updated
- Web Services Security Gateway (XML/SOAP)
- White/Black/Grey listing
- Realtime blacklisting
- Proactive protection including
 - Secure session management (in this case, JAF sets up a separate, secure session with a cryptographically secure session ID between the Web server and the client).
 - Protection of cookies (in this case cookies no longer exit the Web server but are saved in the secure session by JAF. So protected cookies are no longer passed to the client.)
 - URL encryption
 - Site usage enforcement
- Prevention of attacks on the SSL stack on the Web server
- Optionally separate handling of SSL connections
- Checks run on
 - requests' syntactic validity
 - HTTP protocol
 - HTTP method
 - User agent
 - Referrer
 - URL
 - Arguments
 - Body text
 - Cookies
 - Validity of XML data in the reconciliation with a given DTD
- Protection of form fields

- Plus, e.g.:
 - Restriction on the maximum number of processed requests per time unit
 - Generation of redirects
 - Replacement of the host name in requests
 - Access restricted to particular days and times of day
 - Access from specific IPs or IP areas restricted
 - Deep linking prevented
 - URLs encrypted
 - The option for the user to have a HTTP BasicAuth based authentication look just like a Session and Form Based authentication without the need to reconfigure the web application.
- The option to connect to an ICAP server. In this case, JAF acts as an ICAP client and passes requests and responses to a specified ICAP server. The ICAP server then returns them to JAF.
- Freely programmable API (Python). Where necessary, the range of functions can also be extended to meet very specific requirements.

Security Monitoring, Alerting, Reporting

- Application control: Dashboard with an overview of all the protected Web applications
 - Separate rulesets for enforcement and monitoring can be used simultaneously (active ruleset / shadow ruleset)
 - Centralised monitoring and reporting
 - Centralised log file analysis and statistics
 - Configurable alarm systems when certain events occur. Notification e.g. by email, post request or as an entry in a separate log file.
 - Statistics presented as graphics to show the distribution of accepted and rejected requests, depending on the time and handler
 - Statistics to give an overview of the load and status of all the cluster slaves
 - Log files with host specific logs of all internal system events and error messages
Default error log with events that affect no specific application and thus affect no specific host (e.g. invalid requests)
 - Audit log recording all administrators' individual actions
 - Event log with a table of all status changes
 - Reports in PDF format with a printable summary of the current configuration and all events that have occurred recently

Contact Layer 47

Tel: +44 (0) 870 382 5050
Fax: +44 (0) 870 382 5520

Email: info@layer47.com
Web: www.layer47.com

Cedar Court,
Grove Business Park,
Waltham Road,
Maidenhead,
Berks,
SL6 3LW

Delivery options

JAF is available in the following versions:

- JAFE - jetNEXUS Application Firewall for Enterprise. Designed for Enterprise Businesses with no restrictions.
- JAFL- jetNEXUS Application firewall Lite. Designed for smaller web infrastructures.
- JAFET jetNEXUS Application Firewall for Enterprise Traffic Manager. Designed to run on the jetNEXUS Enterprise Traffic Manager appliance.

Supported operating systems

- Solaris / Linux / BSD
- Windows

Supported Web servers

- Apache 2.0 or 2.2
- IIS 5, 6 or 7
- ISA 2004 or 2006
- J2EE Server (e.g. Apache Tomcat)
- Lighttp Web server