TEST METHODOLOGY

Web Application Firewall

v6.2

Table of Contents

1 Introduction

1.1 The Need for Web Application Firewalls

Attackers have moved up the stack. They are no longer simply attacking the web server and its underlying operating systems, they are attacking the web applications running on the web server that are front-ending critical corporate data. Such applications are often incredibly complex and difficult to secure effectively, and simple coding errors can render them wide open to remote exploits.

To regain the upper hand against current attacks, enterprises must in turn evolve their network defenses to provide a different kind of protection. Web application firewalls (WAF) exist in order to prevent web servers and their applications from being exploited.

1.2 About This Test Methodology and Report

NSS Labs' test reports are designed to address the challenges faced by IT professionals in selecting and managing security products. The scope of this particular report includes:

- Security effectiveness
- Performance
- Stability and Reliability
- Management & Configuration
- Total cost of ownership (TCO)

As organizations come to rely on WAF technology to protect critical assets, the stability and reliability of WAF solutions is imperative. Therefore, regardless of any new deep inspection capabilities, a significant requirement of any WAF technology is that it must be as stable, as reliable, as fast, and as flexible as the existing network that it is protecting.

As part of its WAF test, NSS subjects each product to a brutal battery of tests that verify the stability and performance of each device tested, determine the accuracy of its security coverage, and ensure that the device will not block legitimate traffic. To assess the complex matrix of WAF performance and security requirements, NSS has developed a specialized lab environment that is designed to properly exercise a WAF product. The test suite contains a large variety of individual tests that evaluate the performance, reliability, security effectiveness and usability of WAF products, providing the most thorough and complete evaluation of WAF products available anywhere today.

WAF products offer various deployment options that may impact their ability to provide adequate security effectiveness:

- **In-line deployment options**
  - **Transparent bridge** – The WAF does not take an active role in intercepting the traffic it is inspecting prior to passing it through to the protected web application. Instead, the WAF behaves like an IPS, inspecting traffic as it passes through its mission segment and defeating attacks by blocking the malicious connection. In this scenario, the WAF cannot actively modify the communications between the client and the web application.
  - **Transparent reverse proxy** – The WAF takes an active role in intercepting all communications between the client and web application by inspecting both the requests and responses prior to passing the HTTP

communications to the protected web application or the client. In this scenario, the client and protected web application are communicating directly with each other and the WAF is not visible to either the client or the web application. The reverse proxy WAF can actively modify the communications between the client and the protected web application allowing for changes such as sanitization of potentially sensitive information from being disclosed (i.e., credit card numbers).

- **Reverse proxy** – As with the transparent reverse proxy, the reverse proxy WAF is also taking an active role by intercepting all communications. The main difference between this scenario and transparent reverse proxy is that the WAF is no longer transparent. The client sends its HTTP transactions directly to the WAF and the WAF communicates directly with the web application, and vice versa. In this scenario, the WAF can still actively modify communications between the client and protected web application as well as offer other advantages such as masking the true IP address of the protected web application.

- **Non in-line deployment options**
  - **Passive** – The WAF does not take an active role in the traffic flow between the client and the protected web application. Instead, the WAF is deployed on a SPAN port that is configured to mirror the traffic to be inspected. Some WAF products that offer this mode may also offer the ability to transmit a TCP RST as an attempt to block malicious attacks. However, in this model, race conditions can occur and not all attacks will be successfully mitigated.

1.3 Inclusion Criteria

NSS invites all vendors claiming WAF capabilities to submit products at no cost. Vendors with major market share, as well as challengers with new technology, will be included.

WAF products may be implemented as in-line devices or software residing on the host web server. If an **in-line appliance** (regardless of the type outlined above), the WAF should be supplied as a single appliance, where possible, with the appropriate number of physical interfaces capable of achieving the required level of connectivity and performance (minimum of one in-line port pair per Gigabit of throughput with a maximum of 10 Gigabits per second).

If the product is **host-based software/agent**, the WAF software will be installed by the vendor on a web server configured to work as an in-line reverse proxy to the test platform (e-commerce web server).

Once installed in the test environment, the product will be configured for the use-case appropriate to the target deployment (e-commerce datacenter). As such, the WAF should be configured to block all traffic when resources are exhausted or when traffic cannot be analyzed for any reason.

2 Product Guidance

NSS issues summary product guidance based on evaluation criteria that is important to information security professionals. The evaluation criteria are weighted as follows:

- **Security effectiveness** – The primary reason for buying a WAF is to block and alert staff to the presence of malicious attacks targeting web application vulnerabilities.
- **Performance** – Correctly sizing a WAF is essential.
- **Stability and Reliability** – Long-term stability is critical for accurate logging and detection/prevention of attacks
- **Management and Configuration** – In particular, how difficult it is to configure, filter events, develop and distribute reports.
- **Value** – Customers should seek low TCO and high effectiveness and performance rankings.

Products are listed in rank order according to their guidance rating.

2.1 Recommended

A *Recommended* rating from NSS indicates that a product has performed well and deserves strong consideration. Only the top technical products earn a *Recommended* rating from NSS — regardless of market share, company size, or brand recognition.

2.2 Neutral

A *Neutral* rating from NSS indicates that a product has performed reasonably well and should continue to be used if it is the incumbent within an organization. Products that earn a *Neutral* rating from NSS deserve consideration during the purchasing process.

2.3 Caution

A *Caution* rating from NSS indicates that a product has performed poorly. Organizations using one of these products should review their security posture and other threat mitigation factors, including possible alternative configurations and replacement. Products that earn a *Caution* rating from NSS should not be short-listed or renewed.

3 Security Effectiveness

The aim of this section is to verify that the system under test (SUT) is capable of detecting, preventing, and logging attack attempts accurately, while remaining resistant to false positives.

The SUT will be configured on-site by the vendor to protect the target websites, either by "training" the SUT – walking through the e-commerce sites (automatically, or manually) – or by creating rulesets and a security policy manually. NSS considers it unacceptable for a product of this nature to be sold without some standard approach and/or recommended settings, or without consultancy included to create a policy specific to the target environment. No custom builds will be accepted. The product version tested must be available to the general public at the time of testing.

3.1 Attack Types

NSS testing provides a demonstration of effectiveness of the SUT to protect vulnerable assets from targeted threats and exploitation. This asset/target and threat based approach forms the basis from which SUT security effectiveness is measured.

The NSS threat and attack suite contains thousands of publically available exploits (including multiple variants of each exploit) and a number of complex web applications which have been constructed to include known vulnerabilities and coding errors. Groups of exploits are carefully selected from this library to test based on the intended *Attack Type* as listed below. Each exploit has been validated to impact the target vulnerable host(s) by compromising either the underlying OS, the web server or the web application itself. The level of compromise can vary between instigating a Denial of Service (DoS) condition, providing adminstrator/root access to the host server, allowing malicious users to amend system parameters or application data before submission, browse and/or retrieve files stored on the host server, escalating user privileges, and so on.

3.1.1 URL Parameter Manipulation

Altering URL data to gain potentially protected information or access protected areas of a website.

3.1.2 Form/Hidden Field Manipulation

Constructing POST requests to access protected information or protected areas of a website, or to manipulate "fixed" data directly (such as pricing information).

3.1.3 Cookie/Session Poisoning

Manipulation of cookie or session variables to access protected information or protected areas of a website.

3.1.4 Cross-Site Scripting (XSS)

The process of manipulating user input in such a way that, when rendered in the context of a webpage, it will be interpreted by the browser as code.

3.1.5 Directory traversal

Altering the URL to access areas of the web server that should not otherwise be accessible.

3.1.6 SQL Injection

Manipulating user input in such a way that, when processed by the database server, it will be interpreted as code, potentially providing direct access to private data.

3.1.7 Padding Oracle attacks

Altering a block-cypher cryptographic hash in such a way as to decrypt encrypted information.

3.1.8 Cross-Site Request Forgery (CSRF)

The process of executing a request on behalf of a user without their knowledge, using a trusted session between a vulnerable website and the user's browser.

3.2 Evasion

This section verifies that the SUT is capable of blocking and logging basic exploits when subjected to varying common evasion techniques, both at the network and application level.

3.2.1 Unmodified Exploit Validation

A number of common exploits are executed across the SUT to ensure that they are detected in their unmodified state. These will be chosen from a suite of older/common basic exploits for which NSS is certain that all vendors will have signatures/rules. None of the exploits that were used in Section 3.1 will be used as evasion baselines. This ensures that vendors are not provided with any information on the content of any part of the main NSS exploit library in advance of the test.

3.2.2 Packet Fragmentation Reassembly

These tests determine the effectiveness of the fragment reassembly mechanism of the SUT.

- Fragments from 8 – 32 bytes in size
- Ordered, out-of-order, or reverse order fragments
- Fragment overlap, favoring new and favoring old data
- Interleaved, duplicate, duplicate with or without incrementing DWORD, duplicate packets with random payload, or duplicate packets scheduled for later delivery
- Any combination of the above methods

It is a requirement of the test that the SUT submitted should have all IP fragmentation reassembly options enabled by default in the shipping product.

3.2.3 Stream Segmentation

These tests determine the effectiveness of the stream reassembly mechanism of the SUT.

- Segments from 1 - 2048 bytes in size
- Ordered, reverse ordered, or out-of-order segments, with favor old or favor new
- Duplicate, duplicate interleaved, duplicate last packet, or overlapping segments
- Invalid or NULL TCP control flags
- Sequence resync requests, random initial sequence number, or out-of-window sequence numbers
- Faked retransmits, PAWS elimination, or segments containing random data
- Endianness interchanged

- Any combination of the above methods

It is a requirement of the test that the SUT submitted should have all TCP stream reassembly options enabled by default in the shipping product.

3.2.4 URL Obfuscation And Normalization

Random URL encoding techniques are employed to transform simple URLs, which are often used in pattern-matching signatures, to apparently meaningless strings of escape sequences and expanded path characters using one or any combination of techniques such as:

- Escape encoding using various character sets
- Microsoft %u encoding
- Path character transformations and expansions
- Null-byte string termination
- HTML entities
- Base64
- Path references
- Padding
- Delimiters

These techniques are combined in various ways for each URL tested, ranging from minimal transformation, to extreme (every character transformed). All transformed URLs are verified to ensure they still function as expected after transformation.

3.3 False Positive Testing

The ability of the SUT to identify and allow legitimate traffic while maintaining protection against attacks and exploits is of equal importance to providing protection against malicious content. This test will include a varied sample of legitimate application traffic which should properly be identified and allowed.

4 Performance

This section measures the performance of the SUT using various traffic conditions that provide metrics for real world performance. Individual implementations will vary based on usage, however these quantitative metrics provide a gauge as to whether a particular SUT is appropriate for a given environment.

4.1 Maximum Capacity

The use of traffic generation appliances allows NSS engineers to create "real world" traffic at multi-Gigabit speeds as a background load for the tests.

The aim of these tests is to stress the inspection engine and determine how it handles high volumes of TCP connections per second, application layer transactions per second, and concurrent open connections. All packets contain valid payload and address data, and these tests provide a representation of a live network at various connection/transaction rates.

Note that in all tests the following critical "breaking points" (where the final measurements are taken) are used:

- **Excessive concurrent TCP connections** – Unacceptable increase in open connections on the server-side.
- **Excessive response time for HTTP transactions** – Excessive delays and increased response time to client.
- **Unsuccessful HTTP transactions** – Normally, there should be zero unsuccessful transactions. Their occurrence indicates that excessive latency is causing connections to time out.

4.1.1 Maximum Concurrent TCP Connections

This test is designed to determine the maximum concurrent TCP connections of the SUT with no data passing across the connections. This type of traffic would not typically be found on a normal network, but it provides the means to determine the maximum possible concurrent connections figure.

An increasing number of Layer 4 TCP sessions are opened through the device. Each session is opened normally and then held open for the duration of the test as additional sessions are added up to the maximum possible. Load is increased until no more connections can be established, and this number is recorded.

4.1.2 Maximum TCP Connections Per Second

This test is designed to determine the maximum TCP connection rate of the SUT with one byte of data passing across the connections. This type of traffic would not typically be found on a normal network, but it provides the means to determine the maximum possible TCP connection rate.

An increasing number of new sessions are established through the SUT, ramped slowly to determine the exact point of failure. Each session is opened normally, one byte of data passed to the host, and then the session is closed immediately. Load is increased until one or more of the breaking points defined earlier is reached.

4.1.3 Maximum HTTP Connections Per Second

This test is designed to determine the maximum TCP connection rate of the SUT with a 1 byte HTTP response size. The response size defines the number of bytes contained in the body, excluding any bytes associated with the HTTP header. A 1 byte response size is designed to provide a theoretical maximum HTTP connections per second rate.

Client and server are using HTTP 1.0 without keep alive, and the client will open a TCP connection, send one HTTP request, and close the connection. This ensures that all TCP connections are closed immediately once the request is satisfied, thus any concurrent TCP connections will be caused purely as a result of latency the SUT introduces on the network. Load is increased until one or more of the previously defined breaking points is reached.

4.1.4 Maximum HTTP Transactions Per Second

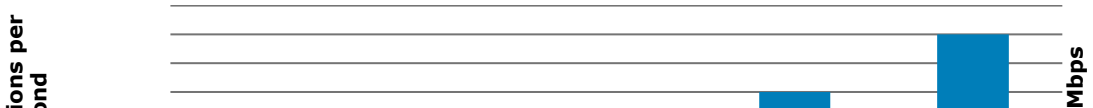
This test is designed to determine the maximum HTTP transaction rate of the SUT with a 1 byte HTTP response size. The object size defines the number of bytes contained in the body, excluding any bytes associated with the HTTP header. A 1 byte response size is designed to provide a theoretical maximum connections per second rate.

Client and server are using HTTP 1.1 with persistence, and the client will open a TCP connection, send ten HTTP requests, and close the connection. This ensures that TCP connections remain open until all ten HTTP transactions are complete, thus eliminating the maximum connection per second rate as a bottleneck (one TCP connection = 10 HTTP transactions). Load is increased until one or more of the breaking points defined earlier is reached.

4.2 HTTP Capacity With No Transaction Delays

The aim of these tests is to stress the HTTP detection engine and determine how the SUT copes with network loads of varying average packet size and varying connections per second. By creating genuine session-based traffic with varying session lengths, the SUT is forced to track valid TCP sessions, thus ensuring a higher workload than for simple packet-based background traffic. This provides a test environment that is as close to “real world” as it is possible to achieve in a lab environment, while ensuring absolute accuracy and repeatability.

Each transaction consists of a single HTTP GET request and there are no transaction delays (i.e., the Web server responds immediately to all requests). All packets contain valid payload (a mix of binary and ASCII objects) and address data, and this test provides an excellent representation of a live network (albeit one biased towards HTTP traffic) at various network loads.



	44Kbyte Response	21Kbyte Response	10Kbyte Response	4.5Kbyte Response	1.7Kbyte Response
CPS	2,500	5,000	10,000	20,000	40,000
Mbps	1,000	1,000	1,000	1,000	1,000

4.2.1 44 KB HTTP response size – 2,500 Connections Per Second

Maximum 2,500 new connections per second per Gigabit of traffic with a 44 KB HTTP response size – average packet size 900 bytes – maximum 140,000 packets per second per Gigabit of traffic. With relatively low connection rates and large packet sizes, all hosts should be capable of performing well throughout this test.

4.2.2 21 KB HTTP response size – 5,000 Connections Per Second

Maximum 5,000 new connections per second per Gigabit of traffic with a 21 KB HTTP response size – average packet size 670 bytes – maximum 185,000 packets per second per Gigabit of traffic. With average connection rates and average packet sizes, this is a good approximation of a real-world production network, and all hosts should be capable of performing well throughout this test.

4.2.3 10 KB HTTP response size – 10,000 Connections Per Second

Maximum 10,000 new connections per second per Gigabit of traffic with a 10KB HTTP response size – average packet size 550 bytes – maximum 225,000 packets per second per Gigabit of traffic. With smaller packet sizes coupled with high connection rates this represents a very heavily used production network.

4.2.4 4.5 KB HTTP response size – 20,000 Connections Per Second

Maximum 20,000 new connections per second per Gigabit of traffic with a 4.5KB HTTP response size – average packet size 420 bytes – maximum 300,000 packets per second per Gigabit of traffic. With small packet sizes and extremely high connection rates this is an extreme test for any host.

4.2.5 1.7 KB HTTP response size – 40,000 Connections Per Second

Maximum 40,000 new connections per second per Gigabit of traffic with a 1.7KB HTTP response size – average packet size 270 bytes – maximum 445,000 packets per second per Gigabit of traffic. With small packet sizes and extremely high connection rates this is an extreme test for any host.

4.3 HTTP Capacity with Transaction Delays

Typical user behavior introduces delays in between requests and reponses, e.g., as users read web pages and decide which links to click next. This next set of tests is identical to the previous set except that these include a 5 second delay in the server response for each transaction. This has the effect of maintaining a high number of open connections throughout the test, thus forcing the SUT to utilize additional resources to track those connections.

4.3.1 21 KB HTTP response with delay

Maximum 5,000 new connections per second per Gigabit of traffic with a 21 KB HTTP response size – average packet size 670 bytes – maximum 185,000 packets per second per Gigabit of traffic. 5 second transaction delay resulting in an additional 50,000 open connections per Gigabit over the equivalent no-delay tests. With average connection rates and average packet sizes, this is a good approximation of a real-world production network, and the SUT should be capable of performing well throughout this test.

4.3.2 10 KB HTTP response with delay

Maximum 10,000 new connections per second per Gigabit of traffic with a 10 KB HTTP response size – average packet size 550 bytes – maximum 225,000 packets per second per Gigabit of traffic. 5 second transaction delay resulting in an additional 100,000 open connections over the equivalent no-delay tests. With large average packet

sizes coupled with very high connection rates, this represents a very heavily used production network, and is a strenuous test for any SUT.

5 Stability & Reliability

Long-term stability is particularly important for an in-line device, where failure can produce network outages. These tests verify the stability of the SUT along with its ability to maintain security effectiveness while under normal load and while passing malicious traffic. Products that are not able to sustain legitimate traffic (or crash) while under hostile attack will not pass.

The SUT is required to remain operational and stable throughout these tests, and to block 100 percent of previously blocked traffic, raising an alert for each. If any non-allowed traffic passes successfully, caused by either the volume of traffic or the SUT failing for any reason, this will result in a FAIL.

5.1 Blocking Under Extended Attack

The SUT is exposed to a constant stream of security policy violations over an extended period of time. The device is configured to block and alert, and thus this test provides an indication of the effectiveness of both the blocking and alert handling mechanisms.

A continuous stream of security policy violations mixed with legitimate traffic is transmitted through the SUT at a maximum of 100 Mbps for a minimum of 8 hours with no additional background traffic. This is not intended as a stress test in terms of traffic load (covered in the previous section) – merely a reliability test in terms of consistency of blocking performance.

The SUT is expected to remain operational and stable throughout this test, and to block 100 per cent of recognizable violations, raising an alert for each. If any recognizable policy violations are passed, caused by either the volume of traffic or the SUT failing open for any reason, this will result in a FAIL.

5.2 Passing Legitimate Traffic Under Extended Attack

This test is identical to 5.1, where the SUT is exposed to a constant stream of security policy violatrions over an extended period of time.

The SUT is expected to remain operational and stable throughout this test, and to pass most/all of the legitimate traffic. If an excessive amount of legitimate traffic is blocked throughout this test, caused by either the volume of traffic or the SUT failing for any reason, this will result in a FAIL.

5.3 Protocol Fuzzing & Mutation

This test stresses the protocol stacks of the SUT by exposing it to traffic from various protocol randomizer and mutation tools. Several of the tools in this category are based on the ISIC test suite and other well known test tools/suites.

Traffic load is a maximum of 350Mbps and 60,000 packets per second (average packet size is 690 bytes). Results are presented as a simple PASS/FAIL – the SUT is expected to remain operational and capable of detecting and logging exploits throughout the test.

5.3.1 Protocol Fuzzing & Mutation – Detection Ports

SUT is exposed to protocol fuzzing and mutation traffic across its inspection ports

5.3.2 Protocol Fuzzing & Mutation – Management Port

SUT is exposed to protocol fuzzing and mutation traffic directed to the management port

5.4 Power Fail

Power to the SUT is cut whilst passing a mixture of legitimate and malicious traffic. On restoring power, the SUT should continue to log and block malicious traffic with no intervention required by the administrator.

5.5 Redundancy

Does the SUT include multiple redundant critical components (fans, power supplies, hard drive, etc.)? (YES/NO/OPTION).

5.6 Persistence Of Data

The SUT should retain all configuration data, policy data and locally logged data once restored to operation following power failure.

6 Management & Configuration

Security devices are complicated to deploy; essential systems such as centralized management console options, log aggregation, and event correlation/management systems further complicate the purchasing decision.

Understanding key comparison points will allow customers to model the overall impact on network service level agreements (SLAs), estimate operational resource requirements to maintain and manage the systems, and better evaluate required skill / competencies of staff.

As part of this test, NSS will perform in-depth technical evaluations of all the main features and capabilities of the centralized enterprise management systems offered by each vendor, covering the following key areas:

- **General Management and Configuration** – How easy is it to install and configure devices, and deploy multiple devices throughout a large enterprise network?
- **Policy Handling** – How easy is it to create, edit and deploy complicated security policies across an enterprise?
- **Alert Handling** – How accurate and timely is the alerting, and how easy is it to drill down to locate critical information needed to remediate a security problem?
- **Reporting** – How effective and customizable is the reporting capability?

For additional information concerning enterprise management testing, refer to the separate management questionnaire document.

7 Total Cost of Ownership

Organizations should be concerned with the ongoing, amortized cost of operating security products. This section evaluates the costs associated with the purchase, installation, and ongoing management of the SUT, including:

- **Product Purchase** – The cost of acquisition.
- **Product Maintenance** – The fees paid to the vendor (including software and hardware support, maintenance and updates.)
- **Installation** – The time required to take the device out of the box, configure it, put it into the network, apply updates and patches, initial tuning, and set up desired logging and reporting.
- **Upkeep** – The time required to apply periodic updates and patches from vendors, including hardware, software, and firmware updates.

Appendix A: Change Log

Version 6.2 – 5 September, 2013

- Minor grammatical issues fixed
- Changed identify and detect to block throughout methodology
- Reordered the items in Section 1.2 to match the order in the methodology
- Added description of deployment options to Section 1.2
- Removed two paragraphs from Section 1.3 that referred to SPAN ports/taps
- Removed Layer 2 requirements from in-line appliances in Section 1.3
- Moved maximum of 10 Gbps to paragraph about in-line appliances
- Reordered the items in Section 2 to match the order in the methodology
- Added 3.1.8 Cross-Site Request Forgery (CSRF) to Section 3.1
- Reworded Section 3.3
- Reworded the introduction paragraph of Section 3
- Reworded second paragraph of Section 4.1.1 to accurately describe how connections are measured
- Reworded second paragraph of Section 4.1.2 to accurately describe how the measurement is taken
- Added breaking points and missing HTTP maximum capacity tests to Section 4.1
- Removed the third paragraph of Section 4.2
- Removed "(even in passive devices)" as active detection/blocking is required by the test methodology
- Reworded the first sentence of the first paragraph in Section 5 to refer to in-line devices and network outages
- Removed "logging of" from the first paragraph of Section 5 as logging legitimate traffic is not a requirement
- Reworded the second paragraph of Section 5 to refer to blocked and non-allowed traffic
- Renamed Section 5.1 to "Blocking" from "Detecting"
- Changed the throughput to a maximum of 100 Mbps from 50% of the rated throughput in Section 5.1
- Added a new section, Section 5.2 Passing Legitimate Traffic Under Extended Attack and removed Section 5.1.1 as it overlapped with this new section.
- Removed references to specific test tools
- Changed "detection" to "inspection" ports in Section 5.3.1

Version 6.1 – 01 May, 2013

- No Change Log available. Change Log Appendix added with version 6.2.

Contact Information

NSS Labs, Inc.
206 Wild Basin Rd
Building A, Suite 200
Austin, TX 78746 USA
+1 (512) 961-5300
info@nsslabs.com
www.nsslabs.com

v.130905a

This and other related documents available at: **www.nsslabs.com**. To receive a licensed copy or report misuse, please contact NSS Labs at +1 (512) 961-5300 or sales@nsslabs.com.