Playing with Web Application Firewalls

Who is **Wendel Guglielmetti Henrique** ?

- Penetration Test analyst at SecurityLabs - Intruders Tiger Team Security division (http://www.intruders.com.br) - One of the leading companies in the segment in Brazil, among our clients are government, credit card industry, etc.

- Affiliated to Hackaholic team (http://hackaholic.org/).

- Has been working in IT since 1997, during the last 7 years he has worked in the computer security field.

- Discovered vulnerabilities in many software programs like Webmails, Access Points, Citrix Metaframe, etc.

- Wrote tools used as examples in articles in national magazines like PCWorld Brazil and international ones like Hakin9 Magazine.

- Speaker at famous Brazilian conferences such as H2HC, Code Breakers and invited as speaker to IT Underground 2006 - Italy and IT1TK1 2007 - Mexico.

Playing with Web Application Firewalls

AGENDA:

- What is WAF?
- Types of operation modes.
- Common topology.
- Passive or Reactive?
- Tricks to detect WAF systems.
- Tricks to fingerprint WAF systems.
- Generic evasion techniques.
- Specific techniques to evade WAF systems.
- What does it fail to protect ?

What is WAF?

Web Application Firewall (WAF): An intermediary device, sitting between a web-client and a web server, analyzing OSI Layer-7 messages for violations in the programmed security policy. A web application firewall is used as a security device protecting the web server from attack.

Source: Web Application Security Consortium Glossary.
http://www.webappsec.org/projects/glossary/#WebApplicationFirewall

What is WAF?

- Web Application Firewalls are often called 'Deep Packet Inspection Firewalls' because they look at every request and response within the HTTP/HTTPS/SOAP/XML-RPC/Web Service layers.

- Some Web Application Firewalls look for certain 'attack signatures' to try to identify a specific attack that an intruder may be sending, while others look for abnormal behavior that doesn't fit the websites normal traffic patterns.

- Web Application Firewalls can be either software, or hardware appliance based and are installed in front of a webserver in an effort to try and shield it from incoming attacks.

What is WAF?

Some notes about definitions:

- Some modern WAF systems work both with attack signatures and abnormal behavior.
- WAF systems do not necessarily need to be installed in front of a webserver, some products allow installation directly into the Webserver machine.
- WAF systems do not necessarily detect only incoming attacks, nowadays many products detect inbound and outbound attacks.

Playing with Web Application Firewalls

Types of operation modes:

Negative model (blacklist based).

Positive model (whitelist based).

Mixed model (mix negative and positive model protection).

Types of operation modes:

A negative security model recognize attacks by relying on a database of expected attack signatures.

Example:

Do not allow in any page, any argument value (user input) which match potential XSS strings like <script>, </script>, String.fromCharCode, etc.

Pros:
- Less time to implement (plug and play or plug and hack? :).

Cons:
- More false positives.
- More processing time.
- Less protection.

Types of operation modes:

A positive security model enforces positive behavior by learning the application logic and then building a security policy of valid known requests as a user interacts with the application.

Example:

Page news.jsp, the field id could only accept characters [0-9] and starting at number 0 until 65535.

Pros:
- Better performance (less rules).
- Less false positives.

Cons:
- Much more time to implement.
- Some vendors provide “automatic learning mode”, they help, but are far from perfect, in the end, you always need a skilled human to review the policies.

Types of operation modes:

A mixed mode uses both a negative and a positive model, in general one of them is predominant.

Common topology:

In general WAF systems can be used with 3 different network topologies:

- Between the webserver and the webclient (the most common).
- Integrated into the webserver (used in small environments).
- Connected in a switch via port mirror, also referred as Switched Port Analyzer (SPAN) or Roving Analysis Port (RAP). (Better performance).

Passive or Reactive?

- Most WAF systems work both: passive and reactive mode
- In general, passive mode is used during the first days, to prevent real users being blocked by false positives.
- In production environments most WAF systems runs in reactive mode.

Tricks to detect WAF systems:

WAF systems leave several signs which permit us to detect them, like:

- Cookies – Some WAF products add their own cookie in the HTTP communication.

Example – Citrix Netscaler:

```
GET /news.asp?PageId=254 HTTP/1.1
Host: www.SomeSite.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.2; en-US; rv:1.8.1.12)
Accept: image/png,*/*;q=0.5
Accept-Encoding: gzip,deflate
Keep-Alive: 300
Proxy-Connection: keep-alive
Referer: http://www.SomeSite.com
Cookie:ASPSESSIONCWKSPSVLTF=OUESYHFAPQLFMNBTKJHGQGXM;
ns_af=xL9sPs2RIJMF5GhtbxSnol+xU0uSx;
ns_af_.SomeSite.com_%2F_wat=KXMhOJ7DvSHNDkBAHDwMSNsFHMSFHEmSr?nmEkaen19mlrw
Bio1/lsrzV810C&
```

Tricks to detect WAF systems:

• Header Rewrite – Some WAF products allow the rewriting of HTTP headers. The most common field is Server, this is used to try to deceive the attackers.

Interesting behavior:

This different behavior allows us to detect the presence of WAF systems.

Some WAF vendors:

- Only rewrite the header in hostile requests.

- Depending on the hostile request, it removes the Server field in the HTTP response.

- If the request is valid and non hostile it keeps the original webserver response.

Tricks to detect WAF systems:

Example – HTTP response – Valid and non hostile request:

HTTP/1.1 200 OK
Date: Fri, 27 Jun 2008 23:14:50 GMT
Server: Apache/2.2.9 (Unix)
X-Powered-By: PHP/4.4.7
Content-Type: text/html
Content-Length: 71746

Example – HTTP response – hostile request:

HTTP/1.1 404 Not Found
Date: Fri, 27 Jun 2008 23:20:26 GMT
Server: Netscape-Enterprise/4.0
Content-Length: 213
Content-Type: text/html; charset=iso-8859-1

Tricks to detect WAF systems:

- Some WAF vendors return different HTTP response error codes in the same URL (valid one) if you insert a hostile parameter (even if the URL points to a file that doesn't exist).

Example – HTTP response – Valid URL and hostile parameter:

HTTP/1.1 501 Method Not Implemented
Date: Fri, 27 Jun 2008 23:30:54 GMT
Allow: TRACE
Content-Length: 279
Connection: close
Content-Type: text/html; charset=iso-8859-1

Tricks to detect WAF systems:

- Some WAF vendors provide a feature to “close connection” with the attacker when a hostile packet is found.

From mod_security documentation:

DROP Action: “Immediately initiate a "connection close" action to tear down the TCP connection by sending a FIN packet.”

Attackers requesting hostile pages or parameters can detect mod_security.

NOTE.: This feature is not available in old versions of mod_security.

Tricks to detect WAF systems:

NOTE: Some of this techniques can be used to detect IPS (Intrusion Prevention Systems) too.

Tricks to fingerprint WAF systems:

All (at least all that I know) WAF systems have a built-in group of rules in negative mode, these rules are different in each products, this rules can be:

- A specific rule for a specific well known vulnerability (for example: IIS Unicode attack).

- A generic rule for a specific well known class of vulnerability (for example: SQL Injections).

These rules are associated with an action (for example: DROP the request, Redirect to another Page, etc).

Tricks to fingerprint WAF systems:

Attackers can create a set of attacks that test for a range of vulnerabilities that most WAF systems protect against or not. In this way we are able to identify built-in rules of a product and consequently what product it is.

Example – Set of attacks to WAF “A”:

- Request using HTTP method different from 1.0 and 1.1 (detected and action taken).
- Request with Content-Length where the method is different than POST (not detected).
- URI with recursive path – even invalid path (detected and action taken).
- Request where Cookie name matches “cmd=”” (detected and action taken).
- Request where URI matches “/usr/X11R6/bin/xterm” (not detected).

Tricks to fingerprint WAF systems:

The attacker can go deeper, and create several mutations for the same attack with different evasion methods, allowing them to have more precise identification of WAF systems and the version running.

Some techniques presented in “Tricks to detect WAF systems” slides can also be useful to help in fingerprint a WAF system.

These techniques can be used to create a big database allowing us to detect most WAF systems and IPS on the market.

Generic evasion techniques:

Today we have a wide range of techniques to evade IPS and some WAF systems, most of these attacks works because:

- Bad normalization and canonicalization implementations in the WAF system.
- Weak rules in the WAF system.
- Evasion at network and transport layer in some cases affect IPS and some WAF systems (depending on topology and product).

Generic evasion techniques:

Common Examples:

- SQL comments in parameters to try to defeat some SQL Injection rules.
- Words in random case to try to defeat some SQL Injection rules.
- SQL query encoding (for example: hex encoding via database features).
- URI encoding (for example: Unicode forward slash).
- IP packet fragmentation.

Specific techniques to evade WAF systems:

Similarly as attackers can fingerprint WAF systems, as presented, they can use a technique to precisely identify which restrictions of a rule applies to a specific class of vulnerabilities.

Example – SQL Injection rule:

- An attacker can insert a hostile SQL Injection to a parameter and expect to be detected and an action taken.

- Using trial and error, is possible to identify specific combinations of strings and characters which are allowed or denied.

- This procedure can be repeated many times to identify for example which character combinations are allowed and when used in conjunction with other allowed combinations, the resulting combination becomes a denied one.

Specific techniques to evade WAF systems:

Once we are able to identify which is black-listed and white-listed, in many cases we are able to reconstruct our SQL query (or other attack) to match the requirements as a non hostile request.

Example – Real life:

In a recent penetration test, we were able to bypass a Citrix Netscaler using this technique.

Basically what we did after identifying the rules was rebuild the query like:

- Removing all “NULL” words.
- Use query encoding in some parts.
- Remove the single quote character “‘”.
- And have fun! :)

What does it fail to protect ?

Some classes of attacks are really difficult to prevent, even for WAF systems, like:

- XSS – Cross Site Scripting is extremely mutable and consequently very hard to effectively protect against.

- File Uploads: - Some WAF systems do a good job in protecting against hostile file uploads, but when dealing against webshell uploads (like php shell, asp shell, jsp shell, etc) they tend to fail when using advanced evasion techniques.

- Remote Command Execution based in Server Response: Is extremely hard to effectively detect remote command execution attacks based in Server Responses (like a rule to identify signs of a “uname -na” or “id” command), because if the attacker is somehow able to interact with a shell, he can use so many evasion methods to encode the output in hex-code, character replacement, etc.

NOTE: Hackaholic - We have a private forum and we are looking for skilled members.

Questions ?

wendel (at) security.org.br