H2HC 2008 – 08 e 09 de Novembro.

Playing with Web Application Firewalls

Wendel Guglielmetti Henrique – http://ws.hackaholic.org
Sandro Gauci – http://www.enablesecurity.com

Who is Wendel ?

- Independent penetration test analyst.
- Affiliated to Hackaholic team.
- Over 7 years in the security industry.
- Discovered vulnerabilities in Webmails, Access Points, Citrix Metaframe, etc.
- Speaker at H2HC, Code Breakers, Defcon, etc.

Who is Sandro ?

- Founder and CSO of EnableSecurity.
- Over 8 years in the security industry.
- Published security research papers.
- Tools - SIPVicious and SurfJack.

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?

- WAFs are often called 'Deep Packet Inspection Firewall'.
- Some WAFs look certain 'attack signature' while others look for abnormal behavior.
- WAFs can be either software or hardware appliance.

What is WAF?

- Modern WAF systems work both with attack signature and abnormal behavior.
- WAFs can be installed as a reverse proxy, embedded or connected in a switch (SPAN or RAP).
- Nowadays many WAF products detect both inbound and outbound attacks.

Vendors

f5

BREACH
SECURITY LABS

eEye Digital Security®

iMPERVA

FORTIFY
SOFTWARE

AppliCure
Technologies

CITRIX®

armorlogic

Who uses WAF?

- Many banks around the world.
- Companies which need high protection.
- Many companies in compliance with PCI DSS (Payment Card Industry - Data Security Standard).

Type of operation modes:

- Negative model (blacklist based).
- Positive model (whitelist based).
- Mixed / Hybrid (mix negative and positive model protection).

Type 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.

Type of operation modes:

A positive security model enforces positive behavior by learning the application logic and then building a security policy of valid know good requests.

Example:

Page news.jsp, the field "id" only accept numbers [0-9] and starting at 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 policy.

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:

```
Host: www.domain.com.br
User-Agent: Mozilla/5.0
Accept:    image/png,image/*;q=0.8,*/*;q=0.5
Accept-Language: pt-br,pt;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://www.domain.com.br/
Cookie:    ASPSESSIONIDAQRBCRDA=HKCIAFFBGFJOCOGGKMLDMKBP;
ns_af=inomH/iNoBWnnKxOeTWogBHpYJwA0;
ns_af_.domain.com.br_%2F_wat=QVNQU0VTU0lPTklEQVFSQkNSREFf?NPkNTil264R
7Pi8zgH5vSKd/S6YA&
```

Tricks to detect WAF systems: Cookies

DEMO

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 (server cloaking).

Example - Armorlogic Profense:

```
Date: Sat, 08 Nov 2008 17:06:58 GMT
Server:    Profense
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma:  no-cache
Content-Type: text/html
Set-Cookie: SID=dkhtir88p1c6v2859rvqkpukg1; path=/
Vary: Accept-Encoding
Content-Encoding: gzip
Keep-Alive: timeout=1, max=10
Connection: Keep-Alive
Transfer-Encoding: chunked
```

Tricks to detect WAF systems: Header Rewrite

DEMO

Tricks to detect WAF systems:

- Different 404 error codes for hostile and non existent pages.
- Different error codes (404, 400, 401, 403, 501, etc) for hostile parameters (even non existent ones) in valid pages.
- Drop Action: "Immediately initiate a "connection close" action to tear down the TCP connection by sending a FIN packet.“
- 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 can help us to detect them.

Tricks to detect WAF systems: Different 404 error

DEMO

Specific techniques to evade WAF systems:

- In a penetration test made 7 months ago, we were able to bypass a Citrix Netscaler using the technique described above.

Example - Real life:

What we did after identifying the rules was rebuild the query like:

1) Removing all "NULL" words.
2) Use database SQL encoding features in some parts.
3) Remove the single quote character "'".
4) And have fun! :)

Attacking Negative Mode:

How can we bypass it?

DEMO

Attacking Positive Mode:

Is possible bypass it?

DEMO

BONUS:

- Armorlogic Profense static root password and sshd enable by default.
- Armorlogic Profense default password at web administration interface .

What we learned?

- Negative model will be bypassed.
- Positive model is harder...
  - nothing is impossible, review your source code!

What’s next?

Wendel Guglielmetti Henrique and Sandro Gauci working on:

- Developing tools to detect WAF systems.
- Developing tools to evade WAF systems.
- Bypass encoder
- Research papers
- Advisories

Questions?

Thank you!

wsguglielmetti [em] gmail.com
sandro [em] enablesecurity.com