Web Application Firewall Profiling and Evasion

Michael Ritter
Cyber Risk Services
Deloitte
Content

1. Introduction
2. WAF Basics
3. Identifying a WAF
4. WAF detection tools
5. WAF bypassing methods
6. Approach for my thesis
7. Output
8. Discussion
Introduction

Michael Ritter

• Study media informatics
• University for Applied Sciences Mittelhessen
• Part-time working student at Deloitte
• About to start my BA thesis
Student
Web Application Firewalls (WAFs)

- WAFs are used to detect and block attacks against vulnerable web applications
- WAFs can offer protection against a large-scale of vulnerabilities
- Often used as second line of defense
- WAFs are a crucial topic to secure a company's web environment
Vendors

modsecurity
Open Source Web Application Firewall

CITRIX NetScaler

IMPÉRVA
Protecting the Data That Drives Business

AQTRONIX

BarraCuda

OWASP
Open Web Application Security Project
Web Application Firewalls (WAFs)

• How do they work?
  – Using a set of rules to distinguish between normal requests and malicious requests
  – Sometimes they use a learning mode to add rules automatically through learning about user behaviour

• Operation Modes:
  – Negative Model (Blacklist based)
  – Positive Model (Whitelist based)
  – Mixed/Hybrid Model (Blacklist & whitelist model)

• Example (Blacklist based):
  – Do not allow in any page any user input like `<script>*</script>`
Implementation of a WAF

• 3 ways to implement a WAF
  – Reverse proxy
  – Inline
  – Connected to a Switch (SPAN->Port Mirroring)
Problems with the implementation

- Using the right rule set
  - Rule sets have an impact on the function of the Web Application behind the WAF
  - Problems
    - Blocking normal requests (false positives)
    - Rule set needs to be adjusted
- Rule set with exceptions 😊
  - Can result in (false negatives)
  - Attacker circumvents the WAF
    - Application exploitation
Identification Methods

**HOW TO IDENTIFY A WAF**
WAF Identification methods

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

https://pentestlab.wordpress.com/2013/01/13/detecting-web-application-firewalls/
WAF Identification methods

• Header alternation (also Citrix Netscaler)
  – Some WAF products change the original response header to confuse the attacker

Citrix Netscaler

```python
def isnetscaler(self):
    
    # NSC_ and citrix_ns_id come from David S. Langlands <dsl 'at' surfstar.com>
    if self.matchcookie('^\(nsAf=|\{citrix_ns_id\}|NSC\)\$'):  
        return True

    # First checks if a cookie associated with Netscaler is present, if not it will try to find if a "Cneoction" or "nnCoecion" is returned for any of the attacks sent
    if self.matchheader(('Cneoction', 'close'), attack=True):
        return True
    if self.matchheader(('nnCoecion', 'close'), attack=True):
        return True
    return False
```

wafw00f.py (Automated Detection Tool)

https://pentestlab.wordpress.com/2013/01/13/detecting-web-application-firewalls/
WAF Identification methods

• Inside the response
  – Some WAF identify themselves inside the response

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Content-Type: text/html
Vary: Accept-Encoding
Server: Microsoft-IIS/7.5
X-Powered-By: ASP.NET
Date: Thu, 05 Dec 2013 03:40:14 GMT
Content-Length: 2616

<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>dotDefender Blocked Your Request</title>
</head>
......
```

```
04-Dec-13

dotDefender Blocked Your Request

Please contact the site administrator, and provide the following Reference ID:

C5D7-93D0-04A0-5959
```

http://www.rafayhackingarticles.net/2013/12/bypassing-modern-wafs-xss-filters-cheat.html
WAF Identification methods

• Response Codes
  – Some WAF products reply with specific response codes

   WebKnight

   [HTTP Request/Response Example]

http://www.rafayhackingarticles.net/2013/12/bypassing-modern-wafs-xss-filters-cheat.html
The Sony Case

NO HACKING

PLEASE!!!!
WAF Identification methods

- Further known methods
  - Drop Action - Sending a FIN/RST packet (technically could also be an IDS/IPS)
  - Pre Built-In Rules - Each WAF has different negative security signatures
  - Side-Channel Attacks (Timing behavior)

http://tacticalwebappsec.blogspot.de/2009/06/waf-detection-with-wafw00f.html
Profiling WAFs

WAF DETECTION TOOLS
WAF detection tools

• imperva-detect.py (Specialised on imperva)
• runs a baseline test + 5 additional tests
• Very quick results

Test 0 - Good User Agent...
Test 1 - Web Leech User Agent...
Test 2 - E-mail Collector Robot User Agent Blocking...
Test 3 - BlueCoat Proxy Manipulation Blocking...
Test 4 - Web Worm Blocking...
Test 5 - XSS Blocking...

--- Tests Finished on [https://www.example.com] -- 4 out of 5 tests indicate Imperva application firewall present ---
WAF detection tools

• More vendor based detection tools:
  – Paradox WAF detection
  – F5 Cookie Decoder Burp extension
  – FatCat SQL Injector

http://wafbypass.me/w/index.php/Bypass_Tools
Nmap script (http-waf-detect)

- script can detect numerous IDS, IPS, and WAF products
- Works with: ModSecurity, Barracuda WAF, PHPIDS, dotDefender, Imperva Web Firewall, Blue Coat SG 400

**Example Usage**

```
nmap -p80 --script http-waf-detect <host>
```

**Script Output**

```
PORT   STATE SERVICE  
80/tcp open   http   
|_http-waf-detect: IDS/IPS/WAF detected
```
Wafw00f.py

- Wafw00f can identify the common patterns of more than 25 WAFs

https://github.com/sandrogauci/wafw00f/blob/master/README.md
Wafw00f.py

• Problem
  – Smart WAFs will hide their identity from cookie values as well as http responses e.g. they give 200 OK responses
• Solution
  – Additional test need to be performed
  – like imperva-detect.py
  – Built-in feature of wafw00f.py

https://github.com/sandrogauci/wafw00f/blob/master/README.md
Bypass the security system

**WAF BYPASSING METHODS**
BYPASSING METHODS

• Five bypassing methods
  – Brute forcing
    • Running a set of payloads
    • Tools like sqlmap use this approach
    • often fails
      – Automated tools
  – Reg-ex Reversing
    • WAF’s rely upon matching the attack payloads with the signatures in their databases
    • Payload matches the reg-ex the WAF triggers alarm

http://www.rafayhackingarticles.net/2013/12/bypassing-modern-wafs-xss-filters-cheat.html
History of payloads

Example:

```html
<script>alert(1);</script>
(normal payload)

==
&lt;/script&gt;&amp;&lt;/script&gt;&amp;alert(1)&amp;&lt;/script&gt;
(HTML mix with upper/lowercase)

==
&lt;script&gt;alert(1)&lt;/script&gt;
==
%3C%73%63%72%69%70%74%3E%61%6C%65%72%74%28%31%29%3B%3C%2F%73%63%72%69%70%74%3E
(HEX-VALUE)

==
&amp;x3C;&amp;x73;&amp;x63;&amp;x72;&amp;x70;&amp;x74;&amp;x3E;&amp;x61;&amp;x6C;&amp;x65;&amp;x72;&amp;x74;&amp;x28;&amp;x31;&amp;x29;&amp;x3B;&amp;x3C;&amp;x2F;&amp;x73;&amp;x63;&amp;x72;&amp;x69;&amp;x70;&amp;x74;&amp;x3E;
(HTML with semicolons)
BYPASSING METHODS

• Vendors know about this issue
  – Preprocessing
  – Transformation of different encodings before the test runs
BYPASSING METHODS

• Browser Bugs
  – Alternative method in case everything fails
  – Using old browser bug to bypass the ruleset

• Google Dorks approach

• Using different language chars
  – e.g. ė instead of e
    • This one is an evasion technique used to circumvent the keyword „select“
Questions I want to answer

BACHELOR THESIS APPROACH
Why is this topic relevant?

• Identifying a WAF will
  – Improve productivity during a pentest
  – Known vulnerabilities in certain products

• How is it possible to evade the security of a WAF?
  – Are old methods still effective against modern WAFs?
  – Are there common weaknesses that can be used during a pentest?
Approach for my thesis – Stage 1

Building a testing lab with 2 environments

– WebApp without a WAF
Approach for my thesis – Stage 1

Building a testing lab with 2 environments

– WebApp with WAFs of several vendors
Approach for my thesis – Stage 2

• Profiling tests on WAF
  – Manual approach vs. Automated tools
  – Did vendors change patterns of their WAF?
Approach for my thesis – Stage 3

- Testing the vulnerabilities without a WAF
  - Documentation of existing vulnerabilities and payloads that I used
Approach for my thesis – Stage 4

• Creation of a payload sets based on the OWASP Top 10
  – SQLi
  – XSS
  – Directory Traversal
  – etc.
Approach for my thesis – Stage 5

• Testing the vulnerabilities with a WAF
  – Documentation of WAF responses
  – Payload passthrough statistics
Approach for my thesis – Stage 6

• Concept a methodology for pentesting web applications behind WAFs
Thesis output

- Pentest methodology for WebApps behind WAFs
  - Are automated tools always working?
  - How can you avoid that your WAF gets identified?
  - What can I do, to bypass a WAF

- Up to date identification patterns for several WAFs
  - In case, I find new patterns I will support the wafw00f project
THANK YOU

... FOR YOUR ATTENTION
Discussion/Exchange

• Further resources for evasion pattern?
• WAF vendors/products?
  – Do you have any suggestions?
  – Do you have experience with poor WAF solutions?
• Whitepapers that might be useful?
• More tools?
• Any ideas for further approaches?