OWASP

Open Web Application Security Project

Web Application Firewall Profiling and Evasion

Michael Ritter

Cyber Risk Services

Deloitte

Content

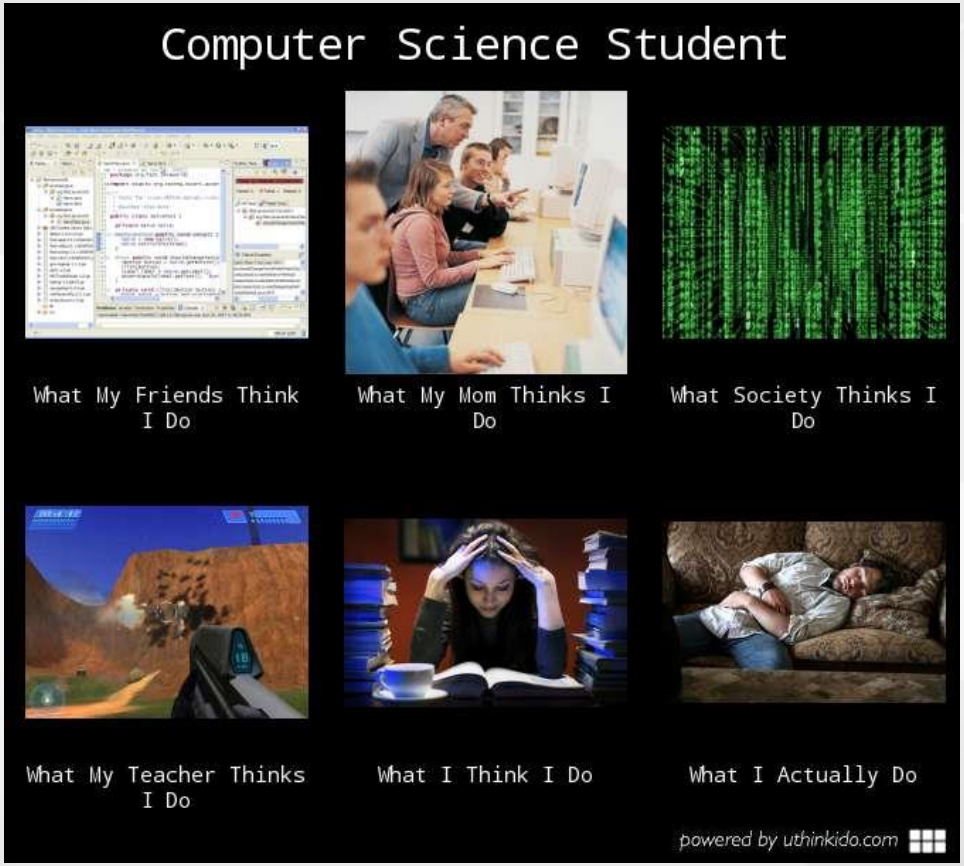
1. Introduction
2. WAF Basics
3. Identifying a WAF
4. WAF detection tools
5. WAF bypassing methods
6. Approach for my thesis
7. Output
8. Discussion

Introduction

Michael Ritter

- Study media informatics
- University for Applied Sciences Mittelhessen
- Part-time working student at Deloitte
- About to start my BA thesis

Student


Computer Science Student
What My Friends Think I Do
What My Mom Thinks I Do
What Society Thinks I Do
What My Teacher Thinks I Do
What I Think I Do
What I Actually Do
powered by uthinkido.com

Basics

WEB APPLICATION FIREWALLS

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 companies web enviroment

Vendors


modsecurity
Open Source Web Application Firewall


CITRIX
NetScaler


IMPERVA
Protecting the Data That Drives Business®


f5


AQTRONIX


Barracuda

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>

http://foxtrot7security.blogspot.de/2012/01/real-world-waf-detection-and-bypass.html

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.

Citrix Netscaler

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

```
def isnetscaler(self):
    """
    First checks if a cookie associated with Netscaler is present,
    if not it will try to find if a "Cneonction" or "nnCoection" is returned
    for any of the attacks sent
    """
    # NSC_ and citrix_ns_id come from David S. Langlands <dsl 'at' surfstar.com>
    if self.matchcookie('^(ns_af=|citrix_ns_id|NSC_)'):
        return True
    if self.matchheader(('Cneonction','close'),attack=True):
        return True
    if self.matchheader(('nnCoection','close'),attack=True):
        return True
    return False
```

wafw00f.py (Automated Detection Tool)

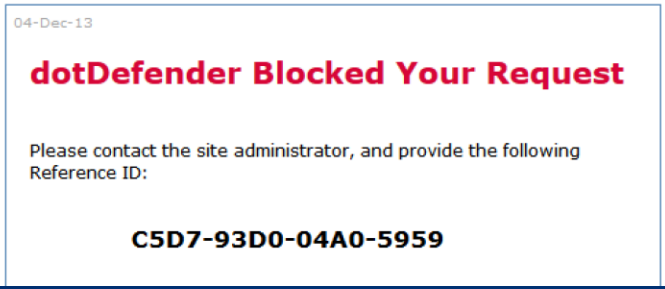
WAF Identification methods

- Inside the response
  - Some WAF identify themselves inside the response

dotDefender

```
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>
……
```



WAF Identification methods

- Response Codes
  - Some WAF products reply with specific response codes

WebKnight

Request

```
GET /?PageID=99<script>alert(1);</script>HTTP/1.1
Host: www.aqtronix.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:25.0) Gecko/20100101
Firefox/25.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

Response

```
HTTP/1.1 999 No Hacking
Server: WWW Server/1.1
Date: Thu, 05 Dec 2013 03:14:23 GMT
Content-Type: text/html; charset=windows-1252
Content-Length: 1160
Pragma: no-cache
Cache-control: no-cache
Expires: Thu, 05 Dec 2013 03:14:23 GMT
```


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

The Sony Case

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 ---
```

http://foxtrot7security.blogspot.de/2012/01/real-world-waf-detection-and-bypass.html
http://wafbypass.me/w/index.php/Bypass_Tools

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>
```

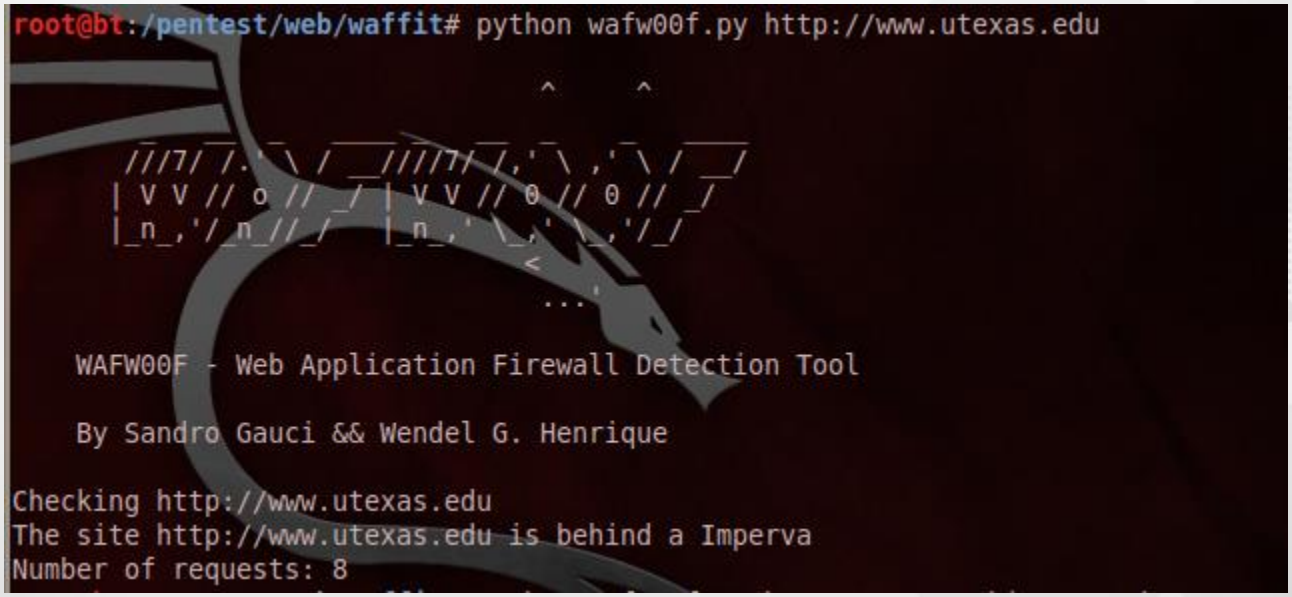
Script Output

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


http://nmap.org/nsedoc/scripts/http-waf-detect.html

Wafw00f.py

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



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

Bypass the security system

WAF BYPASSING METHODS

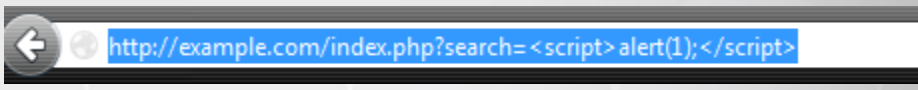
BYPASSING METHODS

- 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

BYPASSING METHODS

History of payloads



Example:

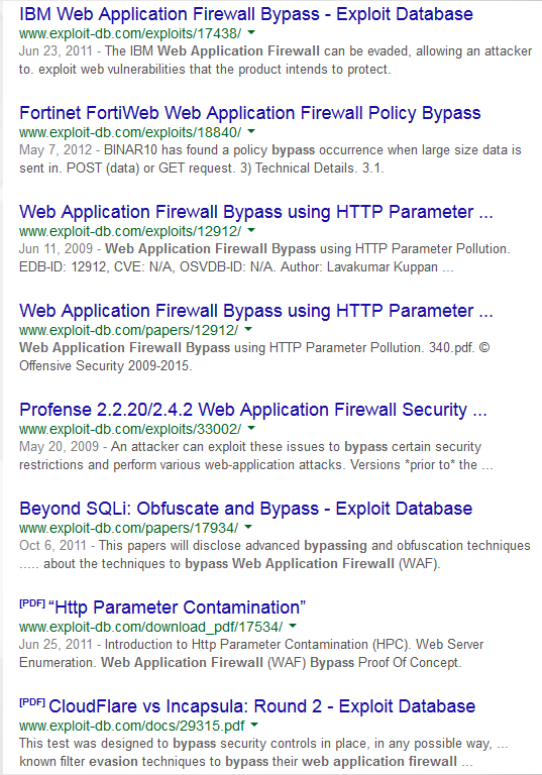
<script>alert(1);</script>
(normal payload)
==
</script><scRiPt>aLeRt(1);</script>
(HTML mix with upper/lowercase)
==
<scr<script>ipt>alert(1)</scr<script>ipt>
==
%3C%73%63%72%69%70%74%3E%61%6C%65%72%74%28%31%29%3B%3C%2F%73%63%72%69%70%74%3E
(HEX-VALUE)
==
<script>alert(1);</script>
(HTML with semicolons)

BYPASSING METHODS

- Vendors know about this issue
  - Preprocessing
  - Transformation of different encodings before the test runs

BYPASSING METHODS

- Brower Bugs
  - Alternative method in case everything fails
  - Using old browser bug to bypass the ruleset
- Google Dorks approach
- Using different language chars
  - e.g. ē instead of e
    - This one is a evasion technique used to circumvent the keyword „select“



IBM Web Application Firewall Bypass - Exploit Database
www.exploit-db.com/exploits/17438/
Jun 23, 2011 - The IBM **Web Application Firewall** can be evaded, allowing an attacker to. exploit web vulnerabilities that the product intends to protect.

Fortinet FortiWeb Web Application Firewall Policy Bypass
www.exploit-db.com/exploits/18840/
May 7, 2012 - BINAR10 has found a policy **bypass** occurrence when large size data is sent in. POST (data) or GET request. 3) Technical Details. 3.1.

Web Application Firewall Bypass using HTTP Parameter ...
www.exploit-db.com/exploits/12912/
Jun 11, 2009 - **Web Application Firewall Bypass** using HTTP Parameter Pollution. EDB-ID: 12912, CVE: N/A, OSVDB-ID: N/A. Author: Lavakumar Kuppan ...

Web Application Firewall Bypass using HTTP Parameter ...
www.exploit-db.com/papers/12912/
Web Application Firewall Bypass using HTTP Parameter Pollution. 340.pdf. © Offensive Security 2009-2015.

Profense 2.2.20/2.4.2 Web Application Firewall Security ...
www.exploit-db.com/exploits/33002/
May 20, 2009 - An attacker can exploit these issues to **bypass** certain security restrictions and perform various web-application attacks. Versions *prior to* the ...

Beyond SQLi: Obfuscate and Bypass - Exploit Database
www.exploit-db.com/papers/17934/
Oct 6, 2011 - This papers will disclose advanced **bypassing** and obfuscation techniques ..... about the techniques to **bypass Web Application Firewall** (WAF).

[PDF] "Http Parameter Contamination"
www.exploit-db.com/download_pdf/17534/
Jun 25, 2011 - Introduction to Http Parameter Contamination (HPC). Web Server Enumeration. **Web Application Firewall** (WAF) **Bypass** Proof Of Concept.

[PDF] CloudFlare vs Incapsula: Round 2 - Exploit Database
www.exploit-db.com/docs/29315.pdf
This test was designed to **bypass** security controls in place, in any possible way, ... known filter **evasion** techniques to **bypass** their **web application firewall** ...

Questions I want to answer

BACHELOR THESIS APPROACH

Why is this topic relevant?

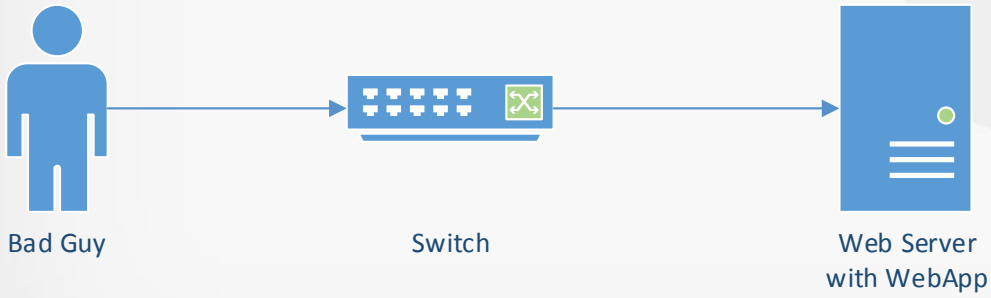
- Identifying a WAF will
  - Improve productivity during a pentest
  - Known vulnerabilities in certain products

- 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 enviroments

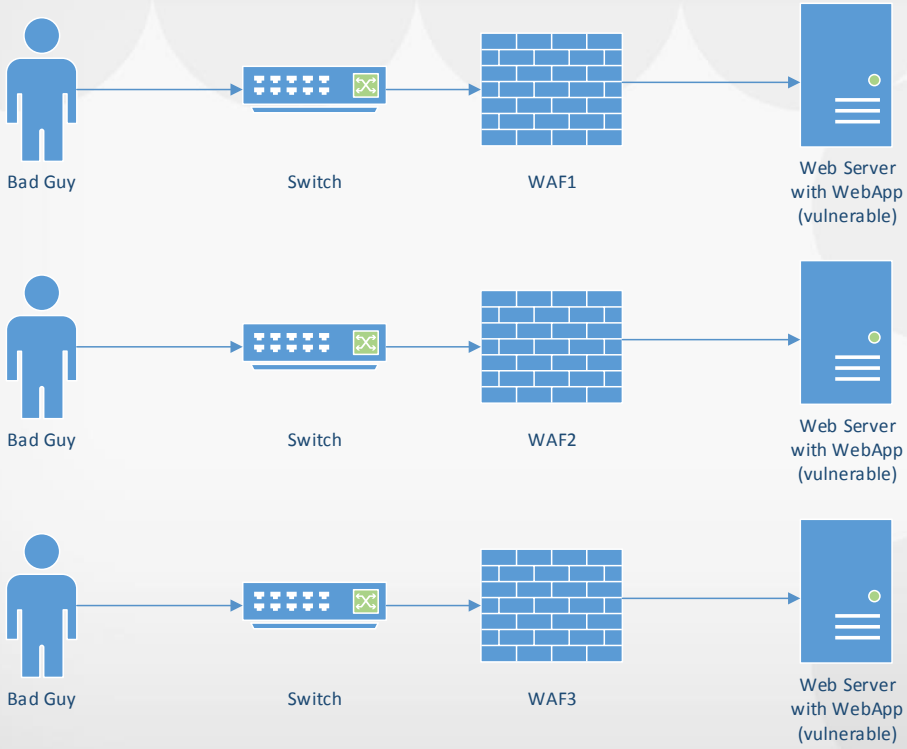
- WebApp without a WAF



Approach for my thesis – Stage 1

Building a testing lab with 2 enviroments

- 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 vulnerabilites 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

- 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
memegenerator.net

Discussion/Exchange

- Further ressources 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?