OWASP 中国
The Open Web Application Security Project

# OWASP Top 10: Effectiveness of Web Application Firewalls

**David Caissy**

AppSec Asia 2016
Wuhan, China

# Agenda

OWASP 中国
The Open Web Application Security Project

- Commercial vs Open Source Web Application Firewalls (WAF)
- Bypassing WAF Filtering
- Effectiveness against the OWASP Top 10


## Web Application Firewalls

- Specialized firewalls
- ***Understand*** web technologies (HTML, SQL, etc.)
- Intrusion Detection System (IDS)
  - Raise alarms
- Intrusion Prevention System (IPS) ⟵
  - Block malicious traffic


## Commercial WAFs

- Dynamic profiling
  - Learn from "known good traffic"
- Central Management and Reporting
- Other functionalities
  - Database Activity Monitoring (DAM)
  - Anti-virus

OWASP 中国
The Open Web Application Security Project

## Open Source WAFs

- Free!
- Good community support
- Some are mature projects
  - ModSecurity
  - IronBee


# The threat

- Script kiddies and automated tools
- Hackers
- Advanced Persistent Threat (APT)
  - Team of expert hackers
  - Lots of resources


## Attacker's Perspective

1. Finding WAF
2. Fingerprinting WAF
3. Test WAF in a lab
4. Launch attacks
5. When blocked, change IP and start again!


## Defender's Perspective

- WAFs often protect many web applications
- No time for custom fine-tuning…
  - Focus on ***basic configuration***


## Problems with Filtering

- Logical errors
- White listing
  - Positive Security Model
  - Could be a support nightmare…
- Black listing
  - Negative Security Model
  - Ok, but not perfect…


Example:

***SQL Injection*** attack on an ***Address*** text field:

```
Elm Street" UNION ALL SELECT pwd
FROM users WHERE username = "freddy" #
```

Blocked by the WAF !

# Problems with Filtering


- But we know these values are valid:
  - 36 O'Connor Street
  - 1025-B Main Blvd., Ottawa (Ontario)
- Therefore WAFs must allow:

  Numbers and characters

  ' - . , ( )

# Problems with Filtering


So instead of:

```
Elm Street" UNION ALL SELECT pass
FROM users WHERE username = "freddy" #
```

We can have:

```
Elm Street' UNION ALL SELECT pass
FROM users WHERE username LIKE 'freddy' --
```


# Encoding and obfuscating the word ***SELECT***

| | |
|---|---|
| SELECT | Blocked by the WAF |
| sElEcT | Upper and lower cases |
| SEL/*comment*/ECT | SQL comment |
| concat('SEL' , 'ECT') | MySQL concat function |
| SE%4CE%43T | URL encoding |
| char(83,69,76,69,67,84) | MySQL ASCII to char function |


What if we mix them all together?

con/*my%20address*/cat%28'S%65',ch/*hello*/ar(76),"e",'Ct')

Black listing can often be bypassed...

OWASP 中国
The Open Web Application Security Project

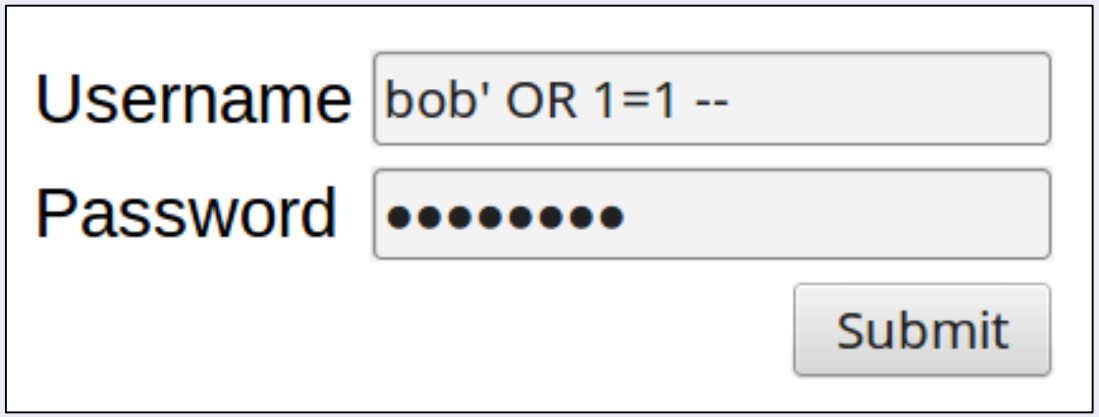
# OWASP Top 10

## Ten Most Critical Web Application Security Risks

- WAFs block the vast majority of attacks, very effective
- WAFs block only automated tools
- WAFs are not an effective safeguard

# A1 - Injection Attacks: Command Injection


- SQL injection, command injection, etc.
- Malicious data sent to an interpreter



# A1 - Injection Attacks: File Injection


WAFs will block or alert when:

- Parameter's length is too long
- Read-only parameters are changed
- Unexpected characters
- Malicious signatures encountered

# A1 - Injection Attacks: File Injection

OWASP 中国
The Open Web Application Security Project

WAFs are very good against injection attacks!

- Stop automated tools and script kiddies
- But they can however be bypassed by experts

**Verdict**

- Block most attacks 
- Won't stop expert hackers and APTs 


# Securing the HTTP Session

```
POST /BigBank/checkCredentials HTTP/1.1
Host: bigbank.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:45.0) Gecko/2010010
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
Referer: http://bigbank.com/BigBank/login
Cookie: JSESSIONID=009363C4309F1B1CDD73127516917F1F
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 33
```


Commercial WAFs ***profile*** the application

- Identify read-only cookies
- Sign and encrypt cookies

With Database Activity Monitoring (DAM)

- Can match web sessions with database queries

OWASP 中国
The Open Web Application Security Project

WAFs can also:

- Track simultaneous user authentication from different IPs
- SSL Termination
- Traffic decryption

OWASP 中国
The Open Web Application Security Project

Broken Authentication vulnerabilities are often ***logical flaws***

<u>Example:</u>

- Invalid username and invalid password:

  "Your username and your password are invalid"

- Valid username and invalid password:

  "Your password is invalid"

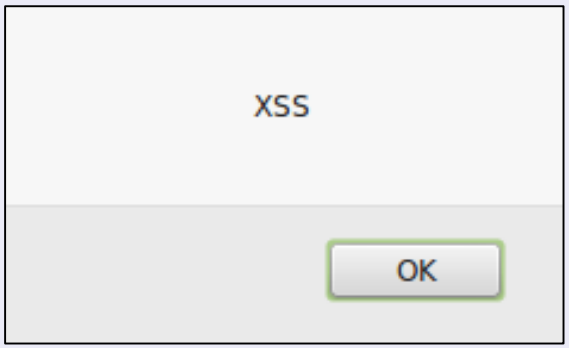
- Vulnerable to ***account harvesting***

# A3 - Cross-Site Scripting (XSS)

OWASP 中国
The Open Web Application Security Project

- Untrusted data in a web page without proper encoding
- An attacker can:
  - Execute scripts in the victim's browser
  - Hijack user sessions
  - Deface web sites

```
<SCRIPT>alert("XSS");</SCRIPT>
```



# A3 - Cross-Site Scripting (XSS)

OWASP 中国
The Open Web Application Security Project

- WAFs prevent XSS attacks using
  - Some white listing
  - Mostly black listing
- Bypassing filtering is difficult
- Overall, they are hard to beat!

# A4 - Insecure Direct Object References


Direct access to an object without proper access control

Shopping Cart Id:

```
www.company.com/shoppingCartId=127
```

What if we change the cart id?

```
www.company.com/shoppingCartId=126
```

# A4 - Insecure Direct Object References


- Most WAFs are <u>ineffective</u> against these attacks 
- Some commercial WAFs:
  - Learn "normal" user activity
  - Then block parameter tampering
  - Combined with a ***database firewall***
    - Slightly improves detection 

# A5 - Security Misconfiguration


**Lack proper hardening**

- Web server
- Database
- Application, including its framework
- Operating system

**Examples include:**

- Default accounts
- Unused services
- Missing patches
- Shared passwords

```
username=admin&password=admin
```


Commercial WAFs integrate with web app vulnerability scanners

- Import scan results
- Dynamically generate rules
  - Virtual patching
- But you need many expensive tools...
- Free WAFs don't support this feature


# Without importing vulnerability scan results

- Commercial WAFs block deviation from the norm
  - Stops script kiddies and automated tools
  - Not enough to stop most hackers...
- Open source WAFs are pretty defenseless...

# A5 - Security Misconfiguration


- Commercial WAFs +
  Vuln scans +
  Dynamic profiling

- Commercial WAFs +
  Dynamic profiling

- Free and open source WAFs

# A6 - Sensitive Data Exposure

OWASP 中国
The Open Web Application Security Project

When sensitive data is expose to an attacker

- Confidential data in clear text within the database
- HTTPS not used

**http://bigbank.com/accountsInfo**

WAFs can:

- Detect common data types (ex: credit card numbers)
- Block <u>server responses</u> containing these values

OWASP 中国
The Open Web Application Security Project

## But WAFs don't know:

- If DBAs have access to unencrypted data
- If HTTPS should be used
- If a given document is confidential
- If cryptographic algorithms are strong enough
- If private keys are stored properly

Only humans can assess sensitive data exposure…

# A7 - Missing Function Level Access Control


When low privilege users can access restricted functions

- Create users
- Assign privileges
- Delete information
- Approve requests

**www.company.com/createUser**

# A7 - Missing Function Level Access Control


WAFs are better at preventing hackers from ***discovering*** the vulnerabilities

- Prevent automated crawlers/spiders
- Prevent enumeration of files and directories

```
www.company.com/admin/
www.company.com/config/
www.company.com/script/
www.company.com/images/
...
```

# A7 - Missing Function Level Access Control

However, WAFs are weak at preventing their ***exploitation***

- Some commercial WAFs ensure that files and functions are accessed in the correct order
  - But this can easily be bypassed...

# A8 - Cross-Site Request Forgery (CSRF)

OWASP 中国
The Open Web Application Security Project

CSRF attacks force a logged-in victim's browser to send a forged HTTP request

All WAFs are ***very effective*** against CSRF attacks:

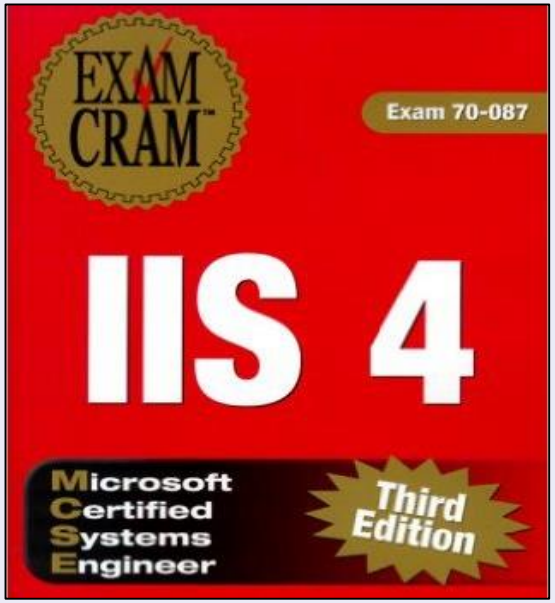
- Automatically add synchronized tokens

```
CSRFTOKEN=54AC455F45EE54638BCE8EE6A
```

# A9 - Using Known Vulnerable Components

OWASP 中国
The Open Web Application Security Project

- When applications use components with known vulnerabilities
- Hackers are motivated at finding them



# A9 - Using Known Vulnerable Components


- WAFs offer:
  - Attack signatures against specific vulnerabilities
  - Set of pre-define policies
- Automated attacks are blocked by WAFs
  - Represents only a small class of vulnerable components...
- WAFs are not aware of vulnerable libraries used internally by web applications

# A10 - Unvalidated Redirects and Forwards


When web applications use untrusted data to forward users to other websites

```
www.company.com/redirect=changePass.net
```

Frequently exploited through:

- XSS
- Phishing attack

# A10 - Unvalidated Redirects and Forwards


Open Source WAFs can:

- Detect when the HTTP request contains the redirected URL

Commercial WAFs can also:

- Block redirects to known malicious web sites
  - Vendor's reputation service
- Generate rules based on known good traffic

# WAFs vs the OWASP Top 10


| | Open Source WAFs | Commercial WAFs |
|---|---|---|
| A1 - Injection Attacks | 🙂 | 🙂 |
| A2 - Broken Authentication | ☹ | ☹ |
| Session Management | 🙂 | 🙂 |
| A3 - Cross-Site Scripting (XSS) | 🙂 | 🙂 |
| A4 - Insecure Direct Object References | 😐 | ☹ |
| A5 - Security Misconfiguration | 😐 | ☹ |
| A6 - Sensitive Data Exposure | ☹ | ☹ |
| A7 - Missing Function Level Access Control | 😐 | ☹ |
| A8 - Cross-Site Request Forgery (CSRF) | 🙂 | 🙂 |
| A9 - Using Known Vulnerable Components | 😐 | ☹ |
| A10 - Unvalidated Redirects and Forwards | 🙂 | 😐 |

# Conclusion


- WAFs are an ***essential component*** of any secure web application deployment
  - Commercial WAFs are better than open source
  - However, open source WAFs are very good against some types of attacks

# Conclusion

OWASP 中国
The Open Web Application Security Project

- WAFs are ***not a replacement*** for secure web development
- Perform **Vulnerability Assessments** and **Penetration Tests** before going to production

# Thank You!


**David Caissy** - IT Security Consultant

- Training for developers and IT security professionals
- Security assessments
- Penetration tests and vulnerability assessments