GOING WHERE NO WAFS HAVE GONE BEFORE

Andy Prow
Aura Information Security

Sam Pickles
Senior Systems Engineer, F5 Networks NZ
Agenda:

• WTF is a WAF?
• View from the Trenches
• Example Attacks and Mitigation Methods
WTF is a WAF?
Surely not another security technology?

• We already have:
  • Intrusion Prevention,
  • Firewalls,
  • Strong Authentication,
  • Patch Management
  • Vulnerability Scanning
  • VPN
  • Antivirus
  • DDoS mitigators
  • ...
Virtually every organisation has vulnerabilities

“8 out of 10 websites vulnerable to attack”
- WhiteHat “security report”

“97% of websites at immediate risk of being hacked due to vulnerabilities! 69% of vulnerabilities are client side-attacks”
- Web Application Security Consortium

“75 percent of hacks happen at the application.”
- Gartner “Security at the Application Level”

“64 percent of developers are not confident in their ability to write secure applications.”
- Microsoft Developer Research
WAFs are a bit different

- They are ONLY for web applications and web services
- Securing vulnerable web applications is not easy for a product to deliver
- Impossible for a “jack of all protocols” security box
In front of the application:

Virtual Patch goes here

Vulnerabilities Exploitable Here

Web App Clients

Web App Servers

Internet

Firewall

Web Application Firewall (WAF)

Botnet/Attacker

Data
Application Protection Strategy

- Ideally there should be no vulnerabilities in the first place... However:
  - Difficult to enforce; especially with third-party code
  - Code changes may be a slow path to remediation, or impossible
  - More secure coding requires more skill and time (cost)
  - Some attack mitigation requires features developed within each application – expensive.

Web Apps

Best Practice Design Methods

Automated & Targeted Testing

WAF

- Should be done regularly – ideally daily
- Scanning technology must be continually evolving
- Multiple tools gives greater coverage
- Operator skill the most important element
- Human penetration testing still required

- Toolkit to improve security – not silver bullet
- Provides remediation, protection, visibility
- Real-time 24 x 7 protection
- Management is important but need not be onerous
- Often the shortest path to remediation
Who is responsible for application security?

Network Security?

Web developers?

Engineering services?

DBA?
Developers are asked to do the impractical...

Application Security?

Application Development

Application Performance

Application Patching

Application Scalability
How long to resolve a vulnerability?

Figure 6. Average number of days for vulnerabilities to be resolved (sorted by class)

Website Security Statistics Report
Challenges of traditional network solutions (FW, IPS)

• HTTP attacks are valid requests
• HTTP is stateless, application is stateful
• Web applications are unique
  • there are no IPS signatures for YOUR web application
• Good protection has to have session context and awareness
• Encrypted traffic facilitates attacks…
• Organizations are living in the dark
  • missing tools to expose/log/report HTTP attacks
Why Not Fix the Code?

Sometimes:

• End of Life applications may not warrant the investment
• Third Party Code may not be available to fix
• Developers have moved on, organisation lacks the resource
• Platform and system dependencies prevent code fix or patch
• Developers asked to focus on new strategic initiatives
  • Patching old apps is sunk cost
  • Building new apps is business growth
…From where I sit, **we NEED WAFs to work**, if nothing else but to provide development groups at least a few days of breathing room. I mean, consider the thousands of issues posted on [sla.ckers.org](http://sla.ckers.org), or [XSSed.com](http://XSSed.com)... Is anyone really under the impression these will get fixed one at a time or anytime soon? And we’re just talking about the XSS. What about the rest?

- Jeremiah Grossman
Pre-Conceived Perception

• No silver bullet
• Can always be bypassed by a skilled attacker
• No replacement for good code
• Only need one for PCI Compliance
  • Item 6.6 “Install a web-application firewall in front of public-facing web applications”
The Eye Opener

• Customer with very broken app (developed overseas)
  • Broken Auth
  • All data and feature restrictions on the client
  • All data validation on the client

• Advanced WAF able to “patch” all features
All of the Top 10?

- Injection: SQL, OS & LDAP Injection
- XSS (Cross-site Scripting)
- Broken Auth. & Session Management
- Direct Object Reference
- XSRF (Cross-site Request Forgery)
- Security Misconfiguration
- Poor Crypto
- Unrestricted URL access
- Insufficient Transport Layer Protection
- Unvalidated Redirects and Forwards
The Easy Bits

• Injection: SQL, OS & LDAP Injection
• XSS (Cross-site Scripting)
• Direct Object Reference
• XSRF (Cross-site Request Forgery)
• Unrestricted URL access
• Insufficient Transport Layer Protection
• Unvalidated Redirects and Forwards
SQL Injection
SQL Injection
Security Evasion using Encoding:

Basic SQL Injection via URI parameter:

' or 1=1 or '

Encoded version:

%27%20%6f%72%20%31%3d%31%20%6f%72%20%27
Evasion using Inline Comments:

'/*comment*/ or/*comment*/ 1=1/*comment*/ or/*comment*/'
Encoding and Commenting together:

Encoded, commented version:

%27%2f%2a%63%6f%6d%6d%65%6e%74%2a%2f%20%6f%72%2f%2a%63%6f%6d%6d%65%6e%74%2a%2f%27
Encoding and Commenting Together:
Signature Matches on Decoded Request:

<table>
<thead>
<tr>
<th>Signature Name</th>
<th>Signature ID</th>
<th>Learn</th>
<th>Alarm</th>
<th>Block</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL-INJ &quot;&quot;/&quot;&quot; (SQL comment) (Parameter)</td>
<td>200002306</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>View details...</td>
</tr>
<tr>
<td>Comments (1)</td>
<td>200016000</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>View details...</td>
</tr>
<tr>
<td>SQL-INJ expressions like &quot;or 1=1&quot; (3)</td>
<td>200002147</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>View details...</td>
</tr>
<tr>
<td>SQL-INJ expressions like &quot; or 1 --&quot;</td>
<td>200002419</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>View details...</td>
</tr>
<tr>
<td>SQL-INJ &quot; &quot; (SQL comment) (Parameter)</td>
<td>200002305</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>View details...</td>
</tr>
</tbody>
</table>

**Context Details for Attack Signature 200002147**

<table>
<thead>
<tr>
<th>Context</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Level</td>
<td>Global</td>
</tr>
<tr>
<td>Wildcard Parameter Name</td>
<td>*</td>
</tr>
<tr>
<td>Actual Parameter Name</td>
<td>username</td>
</tr>
<tr>
<td>Parameter Value</td>
<td>&quot;*/0x20 or/<strong>/0x20 1234=1234/</strong>/0x20#&quot;</td>
</tr>
<tr>
<td>Detected Keywords</td>
<td>username=&quot;*/0x20 or/<strong>/0x20 1234=1234/</strong>/0x20#&quot;</td>
</tr>
</tbody>
</table>
Not So Easy Bits...
Not so Easy Bits…

- Broken Auth. & Session Management
- Security Misconfiguration – Exposed Web Services
- And Business Logic Flaws…
Authorisation – Data Acess

- All data is returned to the client app
- Client only shows restricted data if you’re allowed to see it…
Server Response Scrubbing

- Parse outgoing data set
- Match user identity and group with content
- Remove unauthorised Records from XML
- Return only authorised data
Broken Auth and Session Management
Log In as One User...
View Another User’s Data:

```
10.0.201.10/user_menu.php?nick=charlie
```

### User's control panel

<table>
<thead>
<tr>
<th>Name</th>
<th>Credit Card</th>
<th>Email</th>
<th>Tel</th>
<th>Address</th>
<th>City</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>Charlie Cano</td>
<td>11111111111111111</td>
<td><a href="mailto:ccano@magnifire.com">ccano@magnifire.com</a></td>
<td>1111111111</td>
<td>42 Madison Ave</td>
<td>New York</td>
<td>221</td>
</tr>
</tbody>
</table>
View Everyone’s Data:

<table>
<thead>
<tr>
<th>Name</th>
<th>Credit Card</th>
<th>Email</th>
<th>Tel</th>
<th>Address</th>
<th>City</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assaf Three</td>
<td>258033333333333</td>
<td><a href="mailto:testme4@test.com">testme4@test.com</a></td>
<td>1234567</td>
<td>12 r st</td>
<td>NA</td>
<td>190</td>
</tr>
<tr>
<td>Mark Shahaf</td>
<td>233232-54544-65655</td>
<td><a href="mailto:testme4@test.com">testme4@test.com</a></td>
<td>1234567</td>
<td>12 r st</td>
<td>NA</td>
<td>190</td>
</tr>
<tr>
<td>Shahaf Mark</td>
<td>3333-45549-65655</td>
<td><a href="mailto:testme4@test.com">testme4@test.com</a></td>
<td>1234567</td>
<td>12 r st</td>
<td>NA</td>
<td>190</td>
</tr>
<tr>
<td>Charlie Cano</td>
<td>1234567890</td>
<td><a href="mailto:testme4@test.com">testme4@test.com</a></td>
<td>1234567</td>
<td>12 r st</td>
<td>NA</td>
<td>190</td>
</tr>
<tr>
<td>Automated User One</td>
<td>1234-1234-1234-1234</td>
<td><a href="mailto:testme4@test.com">testme4@test.com</a></td>
<td>1234567</td>
<td>12 r st</td>
<td>NA</td>
<td>190</td>
</tr>
<tr>
<td>Pasha</td>
<td>1234-4321-1234-4321</td>
<td><a href="mailto:testme4@test.com">testme4@test.com</a></td>
<td>1234567</td>
<td>12 r st</td>
<td>NA</td>
<td>190</td>
</tr>
<tr>
<td>Bill</td>
<td>1234-4321-1234-4321</td>
<td><a href="mailto:testme4@test.com">testme4@test.com</a></td>
<td>1234567</td>
<td>12 r st</td>
<td>NA</td>
<td>190</td>
</tr>
<tr>
<td>Him</td>
<td>1234-4321-1234-4321</td>
<td><a href="mailto:testme4@test.com">testme4@test.com</a></td>
<td>1234567</td>
<td>12 r st</td>
<td>NA</td>
<td>190</td>
</tr>
</tbody>
</table>
Dynamic Parameter

• Server sends out parameters
  • Form fields, URI parameters in links, Cookies, etc

• WAF will parse and sign these in a cookie

• Inbound requests must present valid signature
  • Any value is OK, as long as it is YOUR value
  • Server must have supplied the parameter value within your session
  • Can’t be changed on the client side
Blocking Response

Invalid Request

Your request is invalid, and has been recorded for security purposes. Please try again or click below to contact security administrators.

Click here to email security administrators

Back

Log Reference: 9679521228968912488
Exposed Web Services
Unauthorised Method Access

• App relies on Client side validation
• Back end methods all open

```xml
POST /items.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/EditItem"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"/>
<soap:Body>
    <EditItem xmlns="http://tempuri.org/"
        xsi:type="xsd:string">
        <sOID>string</sOID>
        <sName>string</sName>
        <sImageURL>string</sImageURL>
        <sDescription>string</sDescription>
    </EditItem>
</soap:Body>
</soap:Envelope>
```
Authorisation for Method Access

- XML Firewalls provide this function
- Client Identity and Role may be used to disallow Method Access
- VLAN or IP address, ID, Device type, etc

<table>
<thead>
<tr>
<th>Method</th>
<th>Namespace</th>
<th>Enabled</th>
</tr>
</thead>
<tbody>
<tr>
<td>EditItem</td>
<td><a href="http://tempuri.org/">http://tempuri.org/</a></td>
<td></td>
</tr>
<tr>
<td>GetItems</td>
<td><a href="http://tempuri.org/">http://tempuri.org/</a></td>
<td></td>
</tr>
<tr>
<td>GetItems2</td>
<td><a href="http://tempuri.org/">http://tempuri.org/</a></td>
<td></td>
</tr>
</tbody>
</table>
Business Logic Flaws
Advanced Mitigation

• Authentication and Authorisation Wrapper
  • Auth proxy
  • 2 factor
  • Certificate, Kerberos, Forms based, NTLM, etc

• Response Modification
  • EXIF tag XSS example
  • CSRF token example

• Enforcing Order of Events (“Flow”)

• Full request and response parsing and modification
  • Session awareness – with session principles
  • Programmable framework used to mitigate app-specific cases
Responsive Actions:

- Drop Request
- Log, Email, SNMP trap
- Respond with Blocking content
  - HTML – Security warning
  - Link to email administrators in case of issues
  - SOAP Fault for web services
  - Javascript injection for AJAX
  - Honeypot silent redirect
- Query the client a bit further
  - Browser or Robot?
  - Send back Javascript to test client before trusting session
- Your ideas here...?
Questions...