Protecting Web Applications with Web Application Firewall

Ong Poh Seng
Data Center Architect
ongps@cisco.com
Agenda

- Why Web Application Firewall (WAF)
- Cisco Web Application Firewall Overview
- Network Deployment Examples
- GUI Demo
- Q&A
Mox plays down the matter by saying that all he did was exploit some poorly written HTML code before suggesting that it was a cross-site scripting vulnerability that had been exploited. By allowing users to enter characters such as > and " into their blog JRLs, JavaScript could be injected into pages in the Community Blogs section and would be executed by subsequent visitors.

A YouTube clip from zennie62 demonstrates the attack. The clip shows a user clicking on the Community Blogs section of the Barack Obama site, which subsequently causes the browser to redirect to hillaryclinton.com. The author speculates that "Senator Clinton’s staffers possibly hired someone to hack into the Barack Obama website system." No evidence is offered to back up this statement.
Focus of today’s attacks

75% of Attacks Focused Here

Custom Web Applications
Customized Packaged Apps
Internal and 3rd Party Code
Business Logic & Code

No magic signatures or patches for your custom PHP script
The Payment Card Industry (PCI) Data Security Standard

| Build and Maintain a Secure Network | 1. Install and maintain a firewall configuration to protect data |
| Protect Cardholder Data            | 2. Do not use vendor-supplied defaults for system passwords and other security parameters |
| Maintain a Vulnerability Management Program | 3. Protect stored data |
|                                  | 4. Encrypt transmission of cardholder data and sensitive information across public networks |
| Implement Strong Access Control Measures | 5. Use and regularly update anti-virus software |
| Regularly Monitor and Test Networks | 6. Develop and maintain secure systems and applications |
| Maintain an Information Security Policy | 7. Restrict access to data by business need-to-know |
|                                     | 8. Assign a unique ID to each person with computer access |
|                                     | 9. Restrict physical access to cardholder data |
|                                     | 10. Track and monitor all access to network resources and cardholder data |
|                                     | 11. Regularly test security systems and processes |

Section 6.5: develop secure web apps, cover prevention of OWASP vulnerabilities

Section 6.6: Ensure all web-facing apps are protected against known threats, either of the following methods:

- secure coding practices
- installing a Web App FW

No change with PCI 1.2!
What sort of attacks are we talking about?

http://www.owasp.org

Top 10

A1 – Cross Site Scripting (XSS) .................................................................
A2 – Injection Flaws ..................................................................................
A3 – Malicious File Execution .................................................................
A4 – Insecure Direct Object Reference ....................................................
A5 – Cross Site Request Forgery (CSRF) .................................................
A6 – Information Leakage and Improper Error Handling ......................
A7 – Broken Authentication and Session Management ..........................
A8 – Insecure Cryptographic Storage .....................................................
A9 – Insecure Communications ..............................................................
A10 – Failure to Restrict URL Access .....................................................

How widespread these attacks are

Top 10 vulnerability classes by percentage likelihood.
Why Not Fix Current Applications?

Every 1000 lines of code averages 15 critical security defects
(US Dept of Defense)

The average business app has 150,000-250,000 lines of code
(Software Magazine)

The average security defect takes 75 minutes to diagnose and 6 hours to fix
(5-year Pentagon Study)

Even if you consider those figures are exaggerated (positively or negatively) the cost of fixing applications is prohibitive

WAF always a very financially sound option!
Question?

Which of the following are some examples of the PCI DSS requirements?

A. Install a Web Application Firewall
B. Need for Anti-Virus update
C. Encrypt transmission of cardholder data
D. Prevention of OWASP vulnerabilities
Product Overview
SDN Secured Data Center: big picture and where does ACE WAF play?

Data Center Edge
- Firewall & IPS
- DOS Protection
- App Protocol Inspection
- Web Services Security
- VPN termination
- Email & Web Access control

Web Access
- Web Security
- Application Security
- Application Isolation
- Content Inspection
- SSL Encryption/Offload
- Server Hardening

Apps and Database
- XML, SOAP, AJAX Security
- XDoS Prevention
- App to App Security
- Server Hardening

Storage
- Data Encryption
- In Motion
- At Rest
- Stored Data Access Control
- Segmentation

Mgmt
- Tiered Access
- Monitoring & Analysis
- Role-Based Access
- AAA Access Control
ACE Web Application Firewall (WAF)

Drop-in solution for

PCI Compliance, Virtual App Patching, Data Loss Prevention

- **Secure** – Deep packet protection of the most common vulnerabilities
- **Fast** – Processes 3,000+ TPS and 10,000+ concurrent connections
- **Drop-in** - Does not require recoding applications, deployable in under an hour
- **PCI 6.5/6.6 compliance is just a few clicks away**

Available Now!
# Not Just Signatures - Comprehensive Security

## Firewall Configuration

<table>
<thead>
<tr>
<th>Security Feature</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Active Security</strong></td>
<td></td>
</tr>
<tr>
<td>HTTP Header Processing</td>
<td>...</td>
</tr>
<tr>
<td>HTTP Exception Mapping</td>
<td>not configured</td>
</tr>
<tr>
<td>Referer Enforcement</td>
<td>enabled, do not require matching</td>
</tr>
<tr>
<td>Cookie Security</td>
<td>cookies processing is disabled</td>
</tr>
<tr>
<td>Data Overflow Defense</td>
<td>enabled</td>
</tr>
<tr>
<td><strong>Message Rewrite</strong></td>
<td></td>
</tr>
<tr>
<td>Credit Card Number Rewrite</td>
<td>-- disabled --</td>
</tr>
<tr>
<td><strong>Message Inspection</strong></td>
<td></td>
</tr>
<tr>
<td>SQL Injection</td>
<td>-- disabled --</td>
</tr>
<tr>
<td>Command Injection</td>
<td>-- disabled --</td>
</tr>
<tr>
<td>LDAP Injection</td>
<td>-- disabled --</td>
</tr>
<tr>
<td>Cross Site Scripting</td>
<td>enabled strict warning</td>
</tr>
<tr>
<td>SSL Injection</td>
<td>-- disabled --</td>
</tr>
</tbody>
</table>

### Active Security

- **Message Content Search and Replace**
- **Signature-Based Inspection**
ACE WAF Network Deployment

WAF deploys as a *full reverse proxy*

- Must have load balancer for HA, failover
  
  (ACE is great for this!)

- Always performs source NAT on traffic
  
  – HTTP header insert exposes source IP to server

- ACE WAF can optionally terminate SSL
  
  – No client cert validation without AXG license
Clustering: Separate Manager

- 2 or more appliances running Firewall component
- 1 appliance running Manager component
Perimeter security: One-armed proxy

- Traffic passes through ACE twice
- Easy to insert into existing ACE deployment
- Allows for fail-open or fail-closed configuration
**Perimeter Security: Two-armed proxy**

- Different contexts on same physical ACE can be used on both sides
- Best practice when backend is multiple hops from ACE WAF, need DMZ separation
One-armed: Terminate SSL at ACE

1: HTTP over SSL
SRC: 128.32.112.233
DST: 63.90.156.160 port 443

2: HTTP (no SSL)
SRC: 128.32.112.233
DST: 10.30.1.151 port 80

3: HTTP (no SSL)
SRC: 10.30.1.151
DST: 10.30.1.200 port 80

4: HTTP (no SSL)
SRC: 10.30.1.151
DST: 10.20.1.10 port 80

WAN

ACE Application Switch

Web Application Consumers

ACE WAFs

Web Services Providers

Key:

HTTP(S)

HTTP (no SSL)

Each flow carries request and corresponding response

• Request in direction of arrow
• Response in opposite direction
One-armed: Terminate SSL at ACE XML Gateway

1: HTTP over SSL
   SRC: 128.32.112.233
   DST: 63.90.156.160 port 443

2: HTTP over SSL
   SRC: 128.32.112.233
   DST: 10.30.1.151 port 443

3: HTTP (no SSL)
   SRC: 10.30.1.151
   DST: 10.30.1.200 port 80

4: HTTP (no SSL)
   SRC: 10.30.1.151
   DST: 10.20.1.10 port 80

Key:

- HTTP(S)
- HTTP (no SSL)
The AXG Web Application Firewall is a full reverse proxy.

In other words, you can have the DNS server point to the IP address of the AXG to represent the actual Web server.

At that point, the AXG accepts all requests destined to the Web server, filters them, and sends them out. The response comes back to the WAF as well for total control of the session.
WAF GUI Demo
Cisco ACE Web Application Firewall Summary

- Full-featured Web application firewall with integrated XML firewall
  Extend protection for traditional HTML-based Web applications to modern XML-enabled Web services applications.

- Access enforcement
  Secure applications from unauthorized access with AAA enforcement mechanism.

- Positive and Negative security enforcement
  Enjoy the best of both worlds by keeping bad traffic patterns out and allowing only good traffic through.

- Human assisted learning
  Deploy policies and profiles in monitoring mode to prevent application downtime due to false positives typical in an automated learning environment.

- Policy-based provisioning
  Achieve increased developer productivity and ease of deployment with sophisticated GUI, rollback, and versioning capabilities.

Q and A