Web Application Security: Connecting the Dots

Jeremiah Grossman
Founder & Chief Technology Officer

OWASP AsiaPac
04.13.2012

WhiteHat SECURITY

Jeremiah Grossman

- Founder & CTO of WhiteHat Security
- 6-Continent Public Speaker
- TED Alumni
- An InfoWorld Top 25 CTO
- Co-founder of the Web Application Security Consortium
- Co-author: Cross-Site Scripting Attacks
- Former Yahoo! information security officer
- Brazilian Jiu-Jitsu Black Belt

WhiteHat Security : Company Overview

- Headquartered in Santa Clara, CA
- WhiteHat Sentinel – SaaS end-to-end website risk management platform
- Employees: 170+
- Customers: 500+

2010 OnDemand Top 100 Private Companies Competition

RED HERRING GLOBAL WINNER 100

Gartner Cool Vendor

Inc. 500

JMP SECURITIES HOT 100 SOFTWARE COMPANIES —2010—

ChannelWeb 20 Coolest Cloud Security Vendors

BANK TECHNOLOGY NEWS btn The FutureNow List

Silicon Valley Innovation Summit AO250 Global —Winner—

We shop, bank, pay bills, file taxes, share photos, keep in touch with friends & family, watch movies, play games, and more.

Cyber-war **Cyber-crime** **Hacktivism**

PwC Survey:

“Cybercrime is now the second biggest cause of economic crime experienced by the Financial Services sector.”

"When you can measure what you are speaking about, and express it in numbers, you know something about it, when you cannot express it in numbers, your knowledge is of a meager and unsatisfactory kind; it may be the beginning of knowledge, but you have scarcely, in your thoughts advanced to the stage of science." - Lord Kelvin

8 out of 10 websites have serious* vulnerabilities

Average annual amount of new serious* vulnerabilities introduced per website by year

2007	2008	2009	2010	2011
1111	795	480	230	79

*** Serious Vulnerability:** A security weakness that if exploited may lead to breach or data loss of a system, its data, or users. (PCI-DSS severity **HIGH**, **CRITICAL**, or **URGENT**)

Vulnerabilities are counted by unique Web application and vulnerability class. If three of the five parameters of a single Web application (/foo/webapp.cgi) are vulnerable to SQL Injection, this is counted as 3 individual vulnerabilities (e.g. attack vectors).

Websites

676,919,707

+32.6 million since March

(producing more code / websites than the market is assessing)

http://news.netcraft.com/archives/2012/04/04/april-2012-web-server-survey.html

SSL Websites

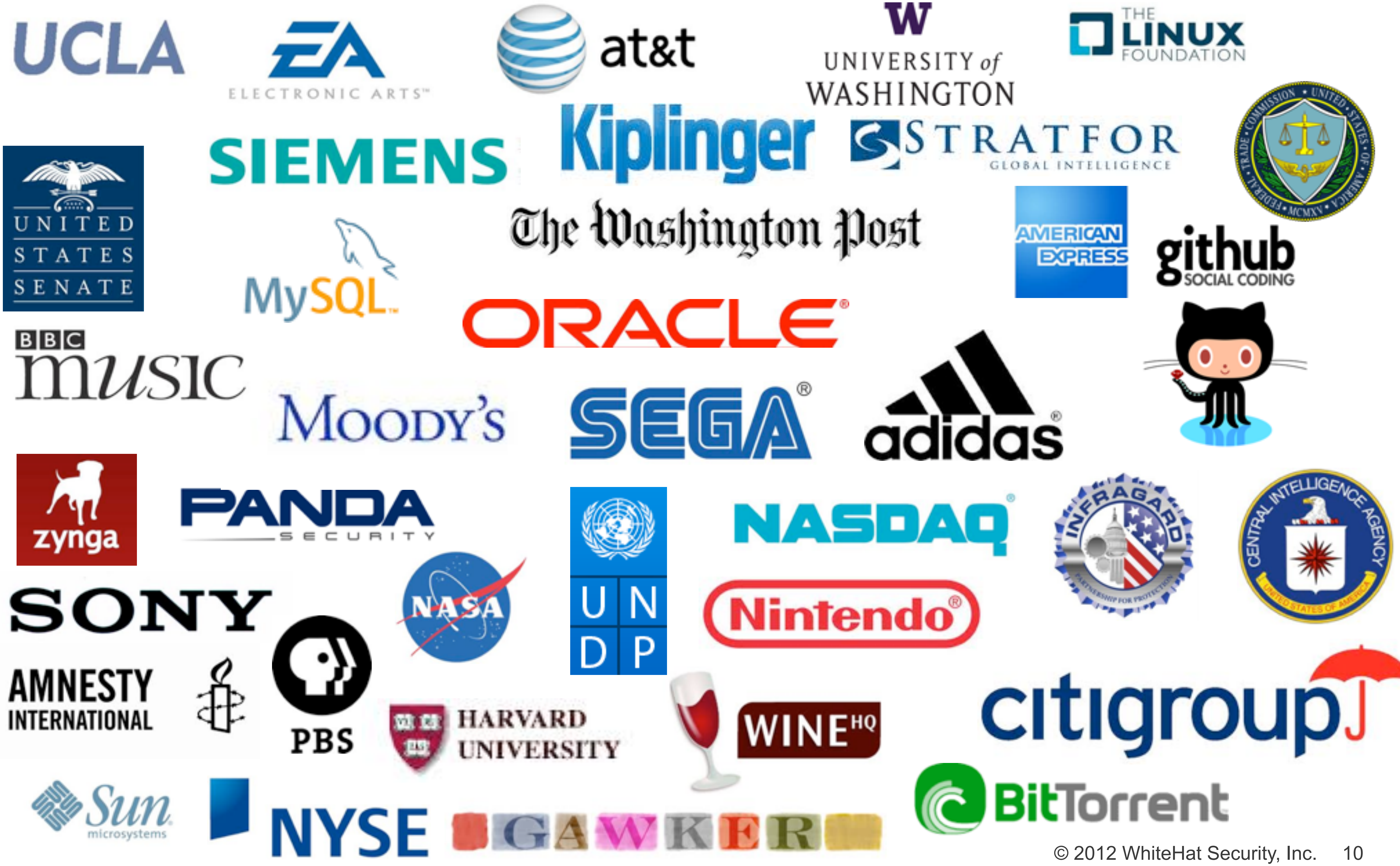
1,200,000

1.2 million x 148 vulns per year =

177,600,000

Undiscovered serious* vulnerabilities
on just the SSL websites.

Website Hacked



Verizon Data Breach Investigations Report:

2010 DBIR:

“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."

2011 DBIR:

“The number of Web application breaches increased last year and made up nearly 40% of the overall attacks.“

2012 DATA BREACH INVESTIGATIONS REPORT

A study conducted by the Verizon RISK Team with cooperation from the Australian Federal Police, Dutch National High Tech Crime Unit, Irish Reporting & Information Security Service, Police Central e-Crime Unit, and United States Secret Service.

verizon

“Web applications abound in many larger companies, and remain a popular (54% of breaches) and successful (39% of records) attack vector.”

2012 DATA BREACH INVESTIGATIONS REPORT

A study conducted by the Verizon RISK Team with cooperation from the Australian Federal Police, Dutch National High Tech Crime Unit, Irish Reporting & Information Security Service, Police Central e-Crime Unit, and United States Secret Service.

855 incidents, 174 million compromised records

Figure 18. Threat action categories by percent of breaches and percent of records – LARGER ORGS

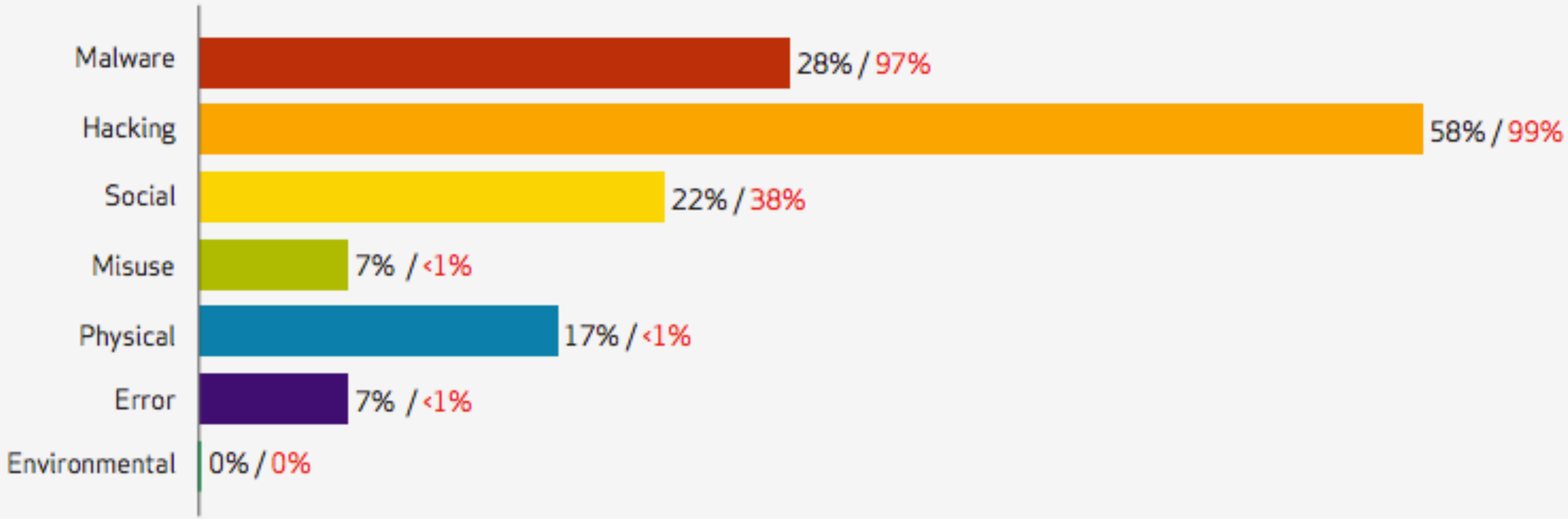


Figure 22. Hacking vectors by percent of breaches within Hacking

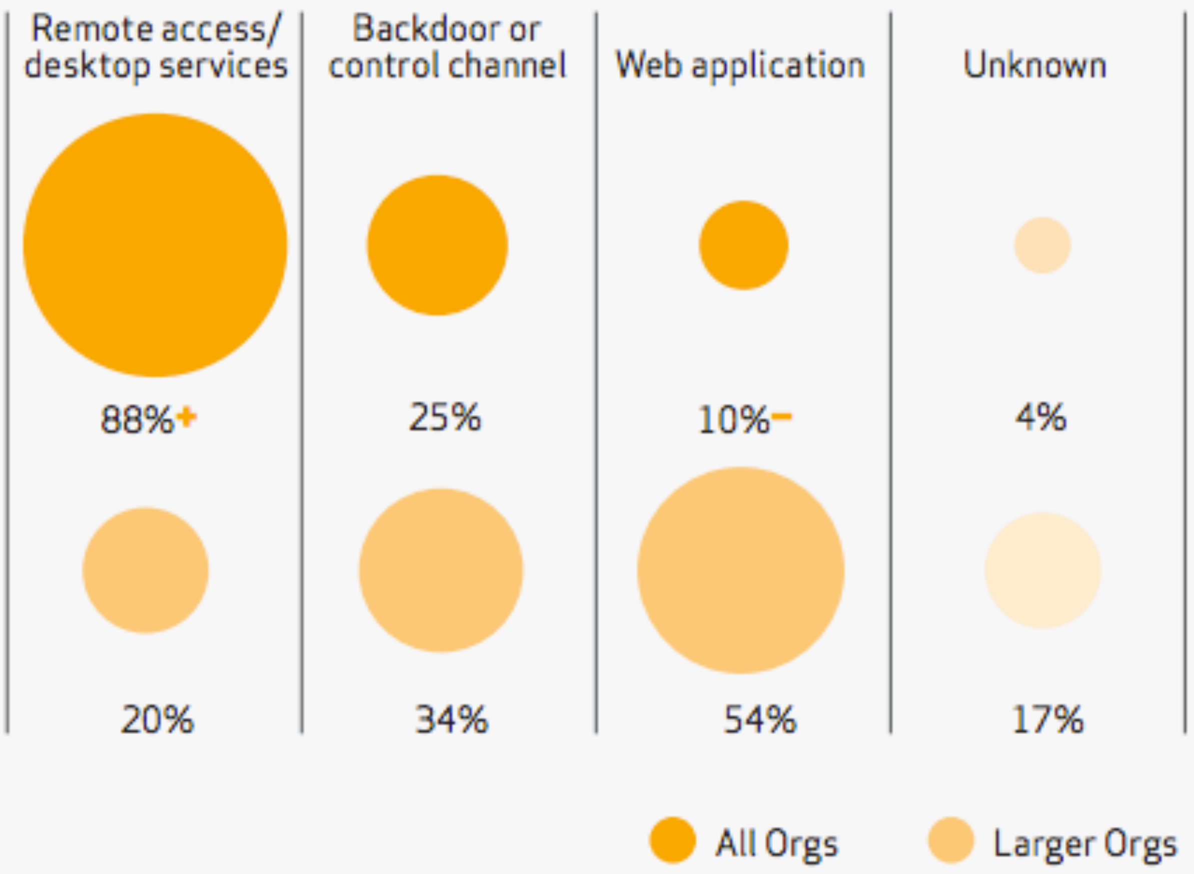
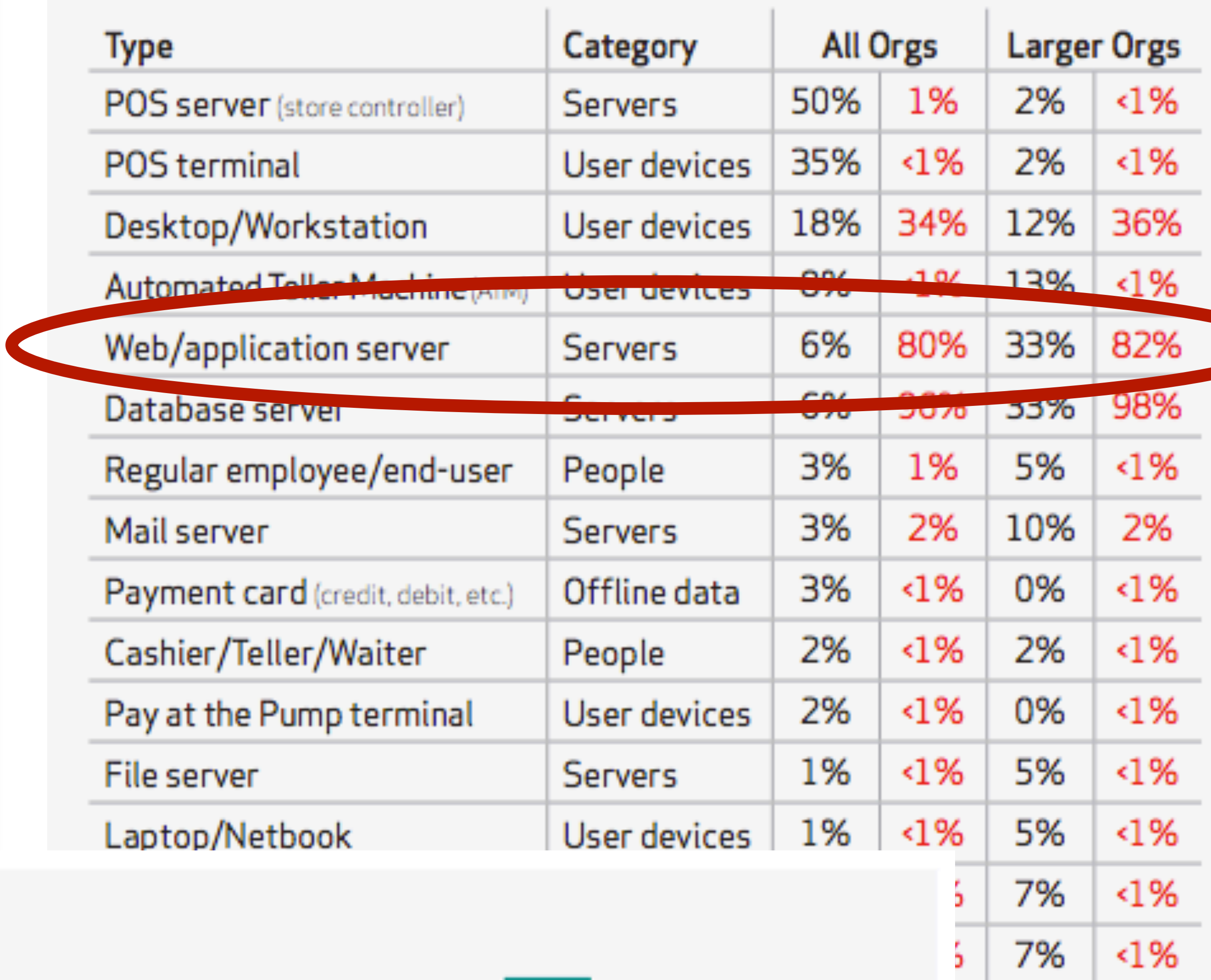


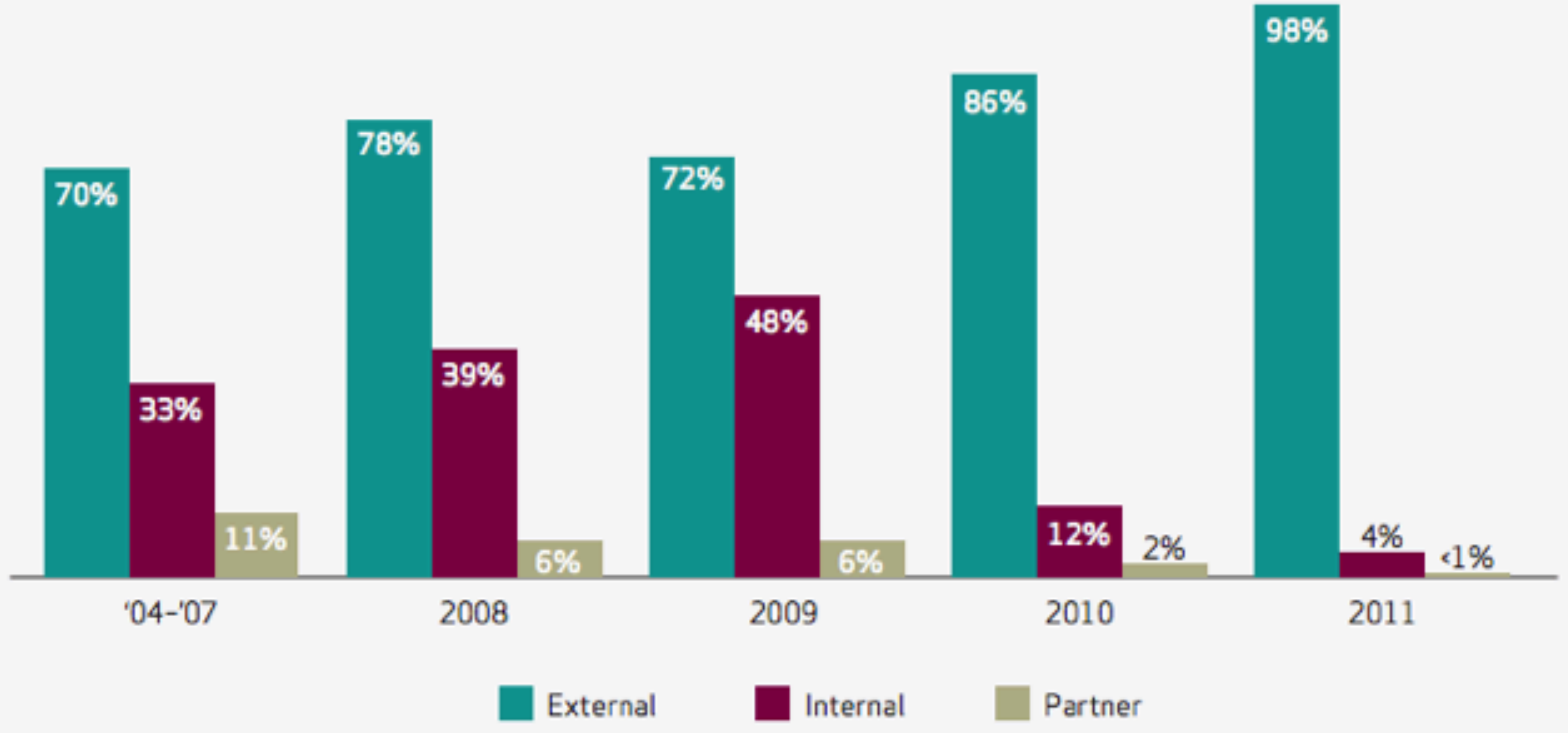
Table 10. Compromised assets by percent of breaches and percent of records*



Type	Category	All Orgs		Larger Orgs	
POS server (store controller)	Servers	50%	1%	2%	<1%
POS terminal	User devices	35%	<1%	2%	<1%
Desktop/Workstation	User devices	18%	34%	12%	36%
Automated Teller Machine (ATM)	User devices	8%	<1%	13%	<1%
Web/application server	Servers	6%	80%	33%	82%
Database server	Servers	6%	96%	33%	98%
Regular employee/end-user	People	3%	1%	5%	<1%
Mail server	Servers	3%	2%	10%	2%
Payment card (credit, debit, etc.)	Offline data	3%	<1%	0%	<1%
Cashier/Teller/Waiter	People	2%	<1%	2%	<1%
Pay at the Pump terminal	User devices	2%	<1%	0%	<1%
File server	Servers	1%	<1%	5%	<1%
Laptop/Netbook	User devices	1%	<1%	5%	<1%
				7%	<1%
				7%	<1%

reaches are not shown

Figure 10. Threat agents over time by percent of breaches



Attacker Profiles

Random Opportunistic

- Fully automated scripts
- Unauthenticated scans
- Targets chosen indiscriminately

Directed Opportunistic

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

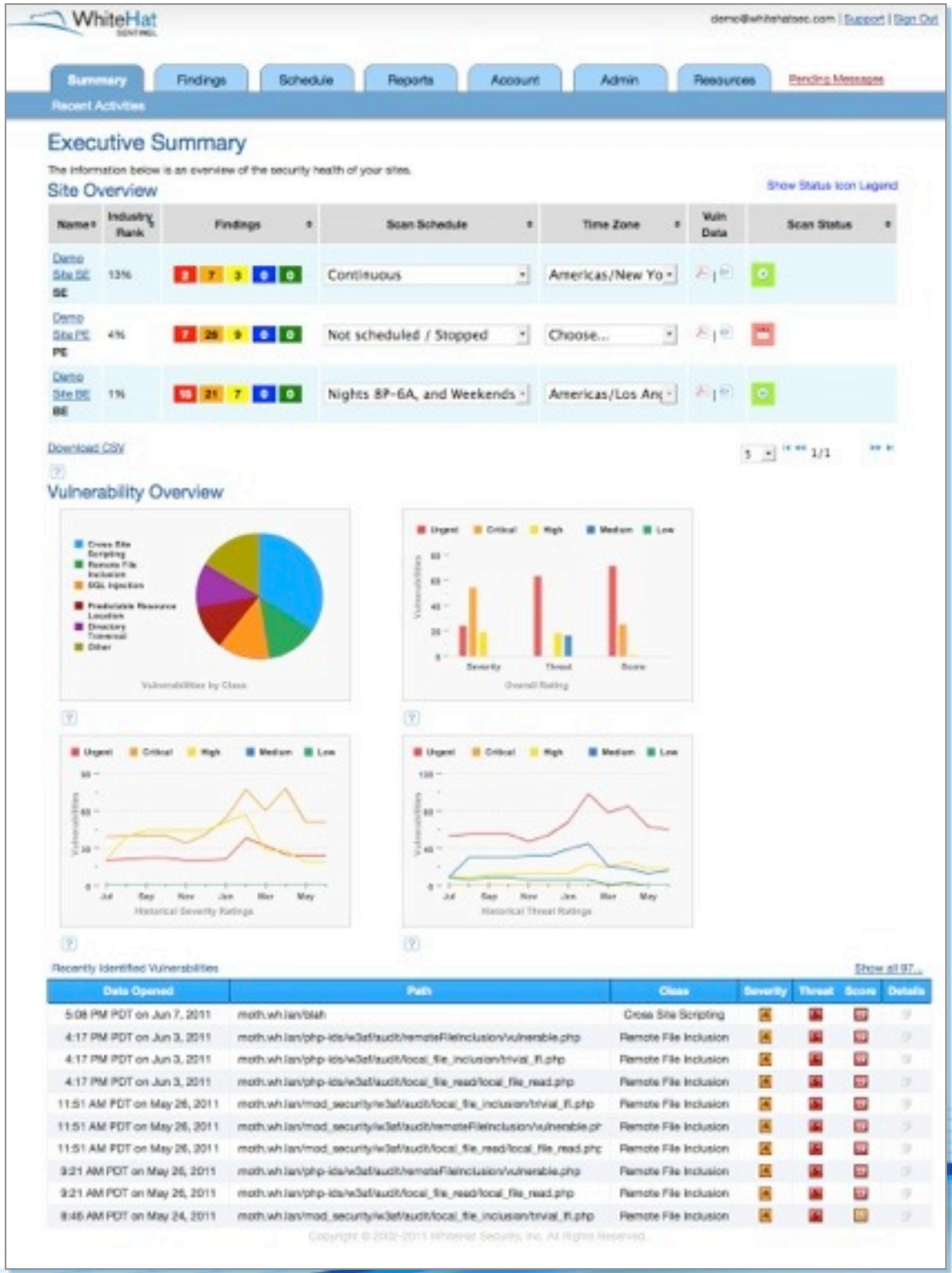
Fully Targeted

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

© 2012 WhiteHat Security, Inc.

WhiteHat Sentinel – Assessment Platform

- **SaaS (Annual Subscription)**
  - *Unlimited Assessments / Users*
- **Unique Methodology**
  - *Proprietary scanning technology*
  - *Expert website security analysis (TRC)*
  - *Satisfies PCI 6.6 requirements*
- **Vulnerability Verification** and prioritization – virtually eliminating false positives
- **XML API** links other security solutions
- **Easy to get started –**
  - *Need URL and Credentials*
  - *No Management of Hardware or Software*
  - *No Additional Training*



Executive Summary

The information below is an overview of the security health of your sites.

Site Overview

Show Status Icon Legend

Name	Industry Rank	Findings	Scan Schedule	Time Zone	Vuln Data	Scan Status
Demo Site SE SE	13%	2 7 3 0 0	Continuous	Americas/New Yo		
Demo Site PE PE	4%	7 26 9 0 0	Not scheduled / Stopped	Choose...		
Demo Site BE BE	1%	10 21 7 0 0	Nights 8P-6A, and Weekends	Americas/Los Ang		

Download CSV

5 1/1

Vulnerability Overview

Recently Identified Vulnerabilities

Show all 97...

Date Opened	Path	Class	Severity	Threat	Score	Details
5:08 PM PDT on Jun 7, 2011	moth.wh.lan/blah	Cross Site Scripting				
4:17 PM PDT on Jun 3, 2011	moth.wh.lan/php-ids/w3af/audit/remoteFileInclusion/vulnerable.php	Remote File Inclusion				
4:17 PM PDT on Jun 3, 2011	moth.wh.lan/php-ids/w3af/audit/local_file_inclusion/trivial_lfi.php	Remote File Inclusion				
4:17 PM PDT on Jun 3, 2011	moth.wh.lan/php-ids/w3af/audit/local_file_read/local_file_read.php	Remote File Inclusion				
11:51 AM PDT on May 26, 2011	moth.wh.lan/mod_security/w3af/audit/local_file_inclusion/trivial_lfi.php	Remote File Inclusion				
11:51 AM PDT on May 26, 2011	moth.wh.lan/mod_security/w3af/audit/remoteFileInclusion/vulnerable.pr	Remote File Inclusion				
11:51 AM PDT on May 26, 2011	moth.wh.lan/mod_security/w3af/audit/local_file_read/local_file_read.phc	Remote File Inclusion				
9:21 AM PDT on May 26, 2011	moth.wh.lan/php-ids/w3af/audit/remoteFileInclusion/vulnerable.php	Remote File Inclusion				
9:21 AM PDT on May 26, 2011	moth.wh.lan/php-ids/w3af/audit/local_file_read/local_file_read.php	Remote File Inclusion				
8:46 AM PDT on May 24, 2011	moth.wh.lan/mod_security/w3af/audit/local_file_inclusion/trivial_lfi.php	Remote File Inclusion				

Copyright © 2002-2011 WhiteHat Security, Inc. All Rights Reserved.

WhiteHat Sentinel

500+
enterprises from start-ups to fortune 500

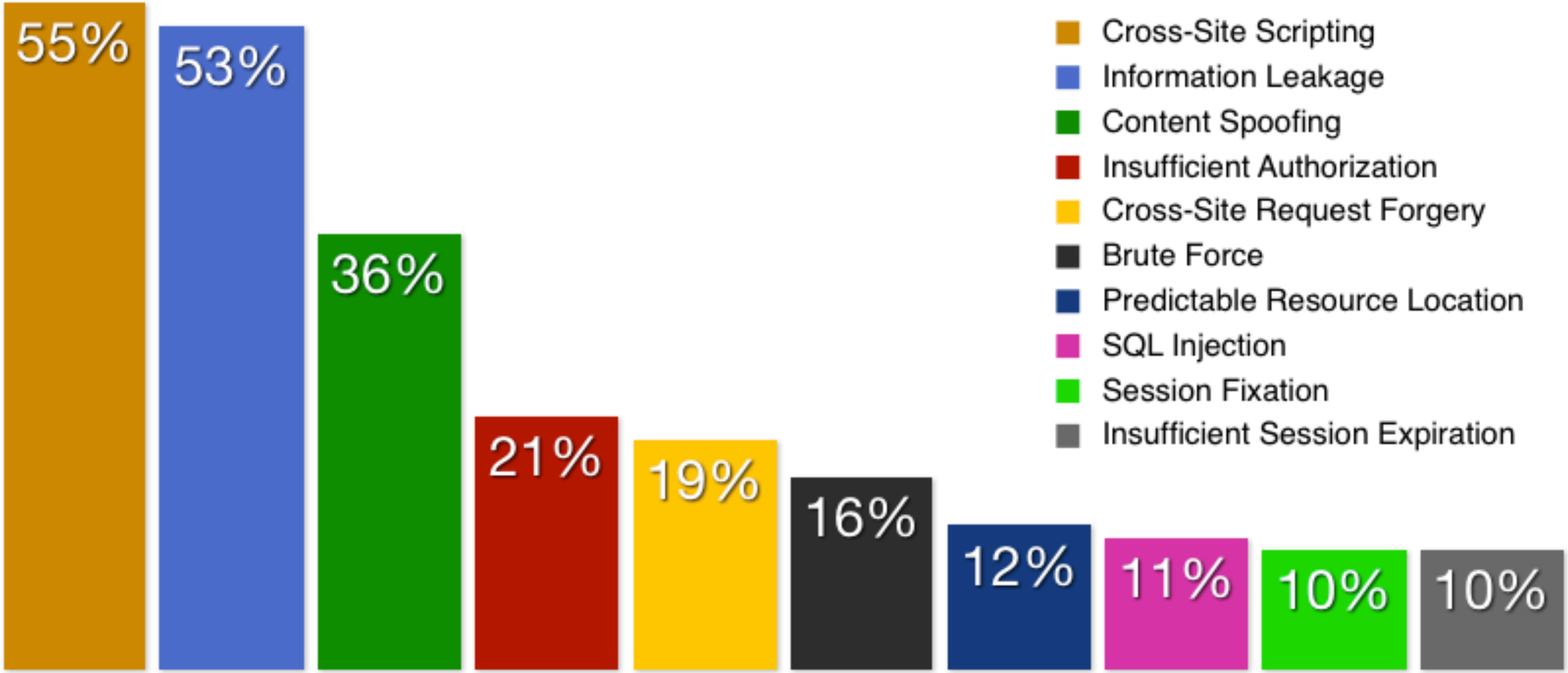
1,000,000
vulnerabilities processed per day

6 Terabytes
data stored per day

7,000+
websites receiving ~weekly assessments

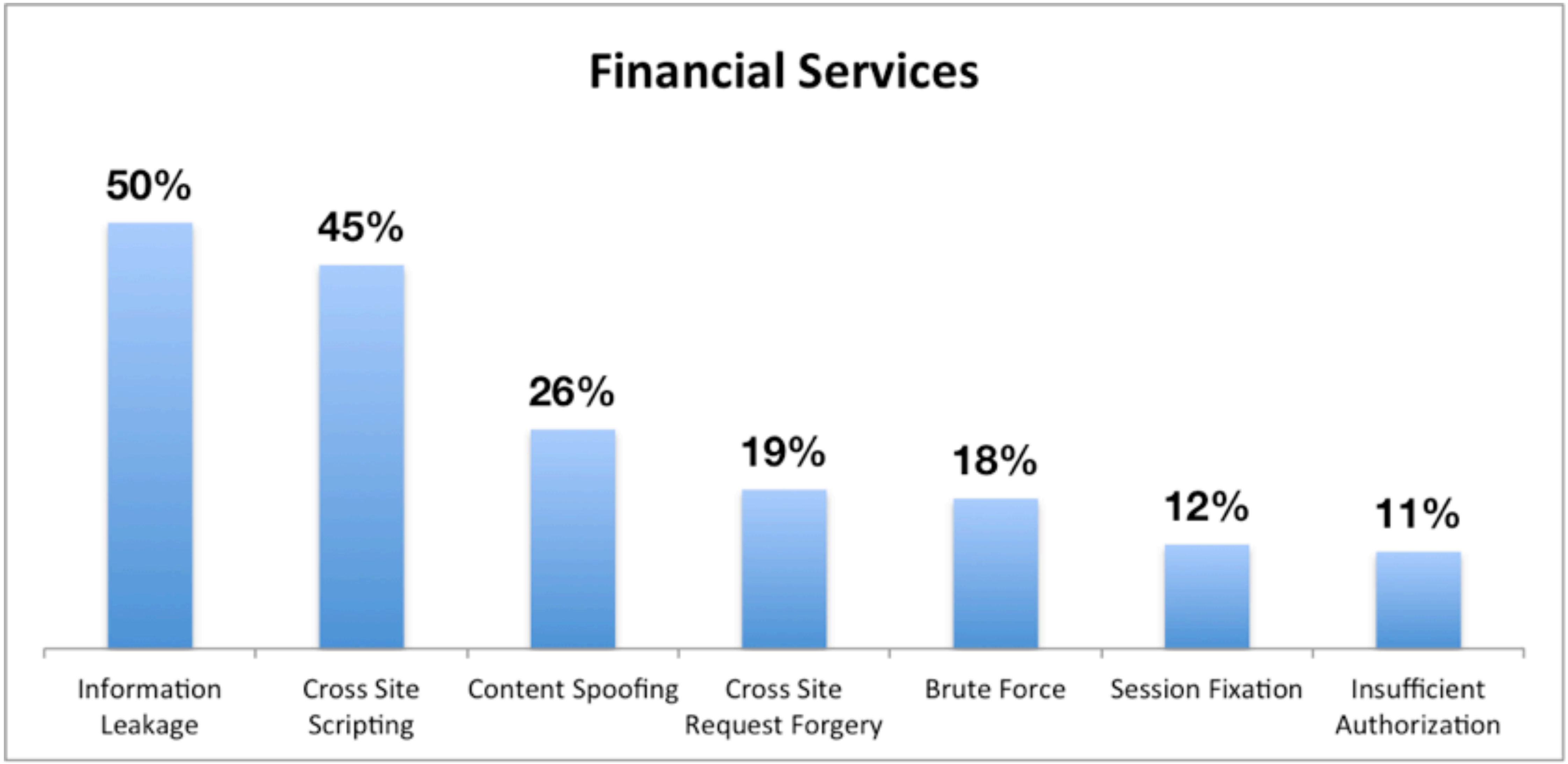
940,000,000
http(s) requests per month

WhiteHat Security Top Ten (2011)



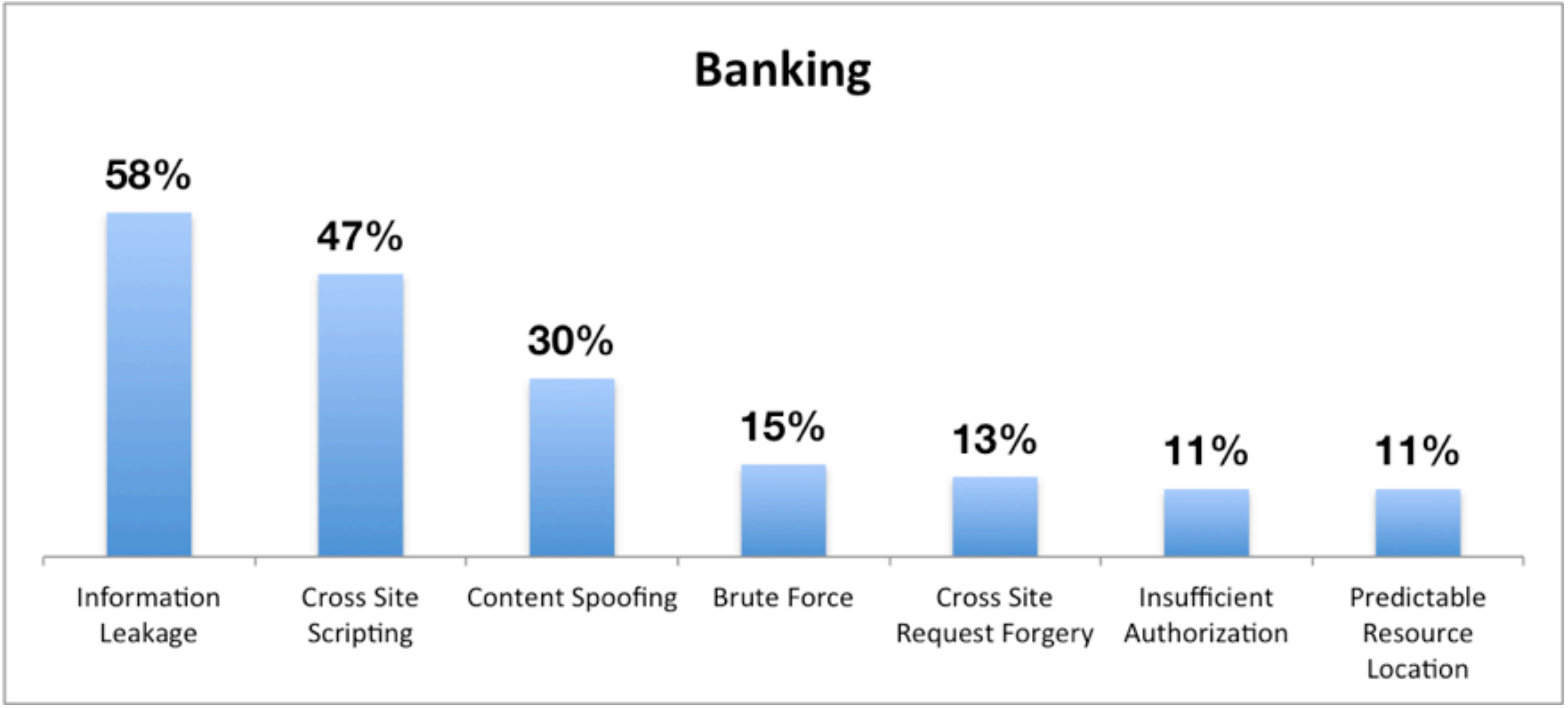
Percentage likelihood of a website having at least one vulnerability sorted by class

Top Seven by Industry (2011)



Percentage likelihood of a website having at least one vulnerability sorted by class

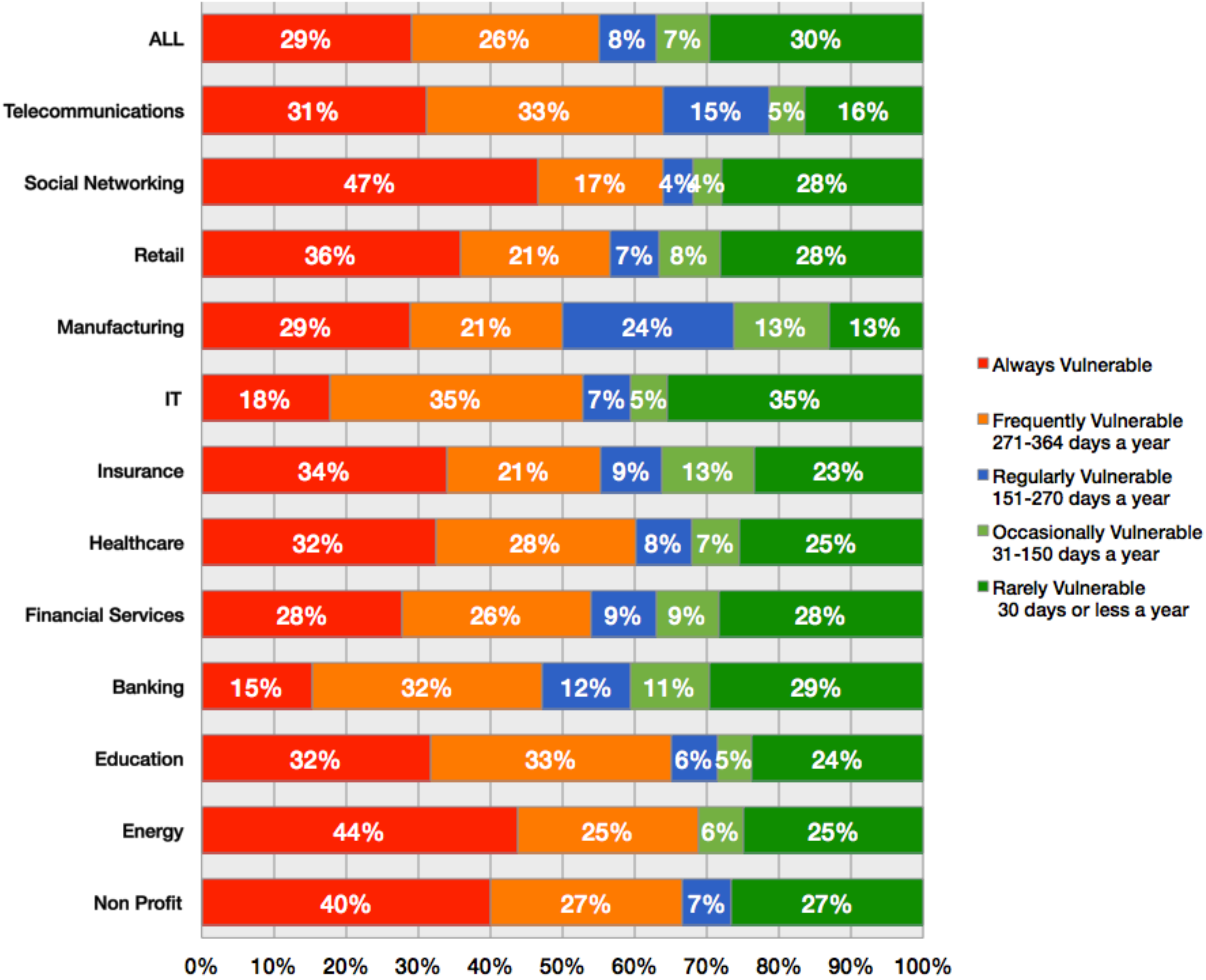
Top Seven by Industry (2011)



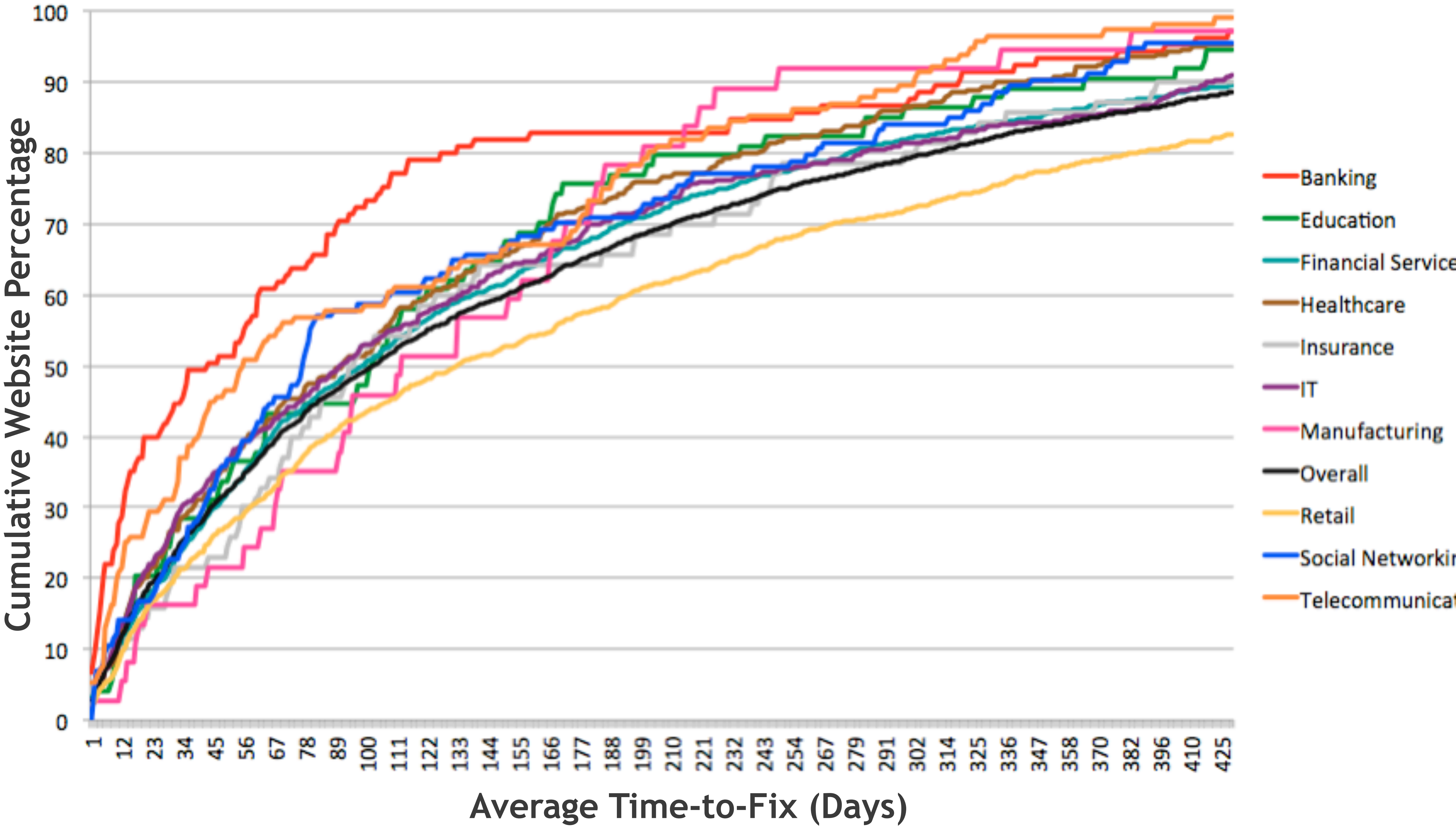
Percentage likelihood of a website having at least one vulnerability sorted by class

Window of Exposure (2011)

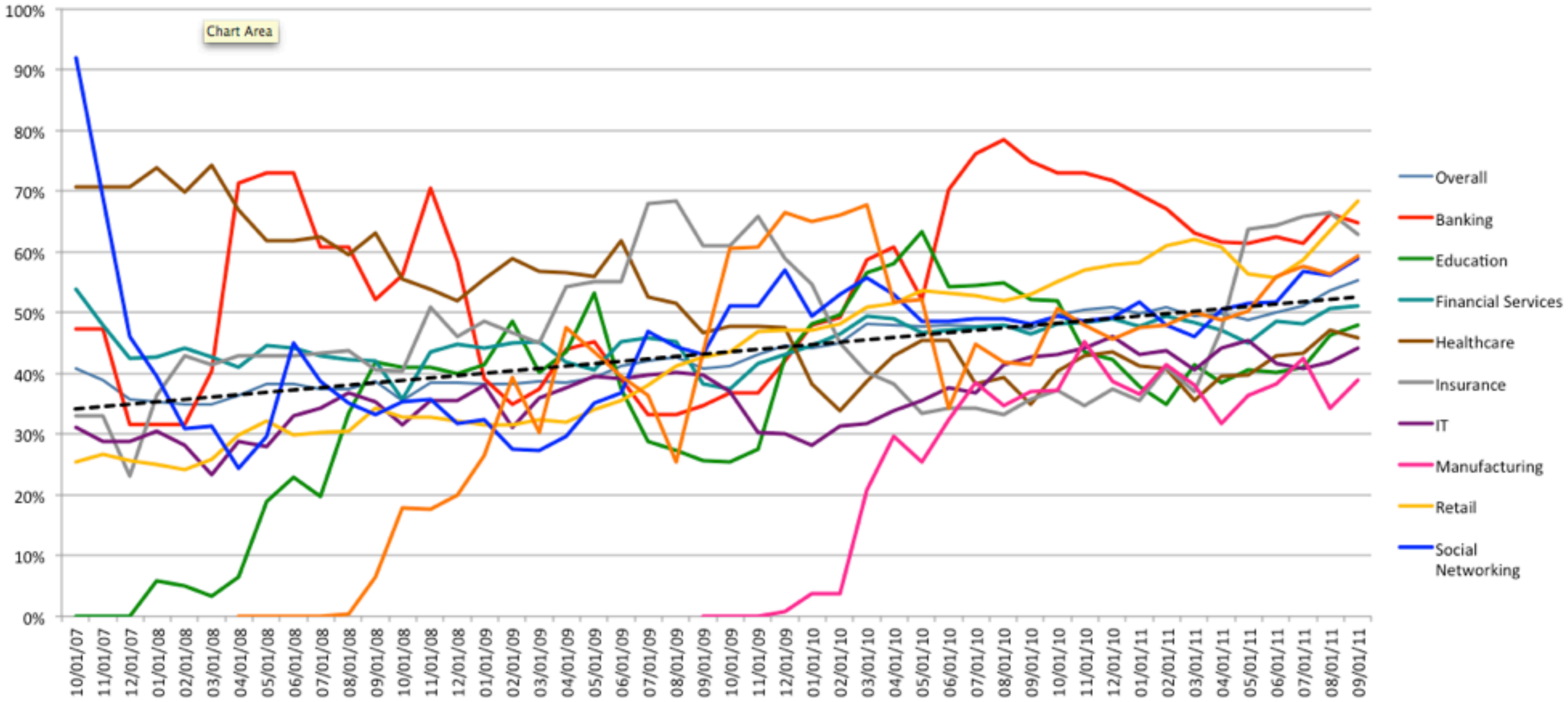
Number of days [in a year] a website is exposed to at least one serious* reported vulnerability.


ALL
29%
26%
8%
7%
30%
Telecommunications
31%
33%
15%
5%
16%
Social Networking
47%
17%
4%
4%
28%
Retail
36%
21%
7%
8%
28%
Manufacturing
29%
21%
24%
13%
13%
IT
18%
35%
7%
5%
35%
Insurance
34%
21%
9%
13%
23%
Healthcare
32%
28%
8%
7%
25%
Financial Services
28%
26%
9%
9%
28%
Banking
15%
32%
12%
11%
29%
Education
32%
33%
6%
5%
24%
Energy
44%
25%
6%
25%
Non Profit
40%
27%
7%
27%
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
🟥 Always Vulnerable
🟧 Frequently Vulnerable 271-364 days a year
🟦 Regularly Vulnerable 151-270 days a year
🟩 Occasionally Vulnerable 31-150 days a year
🟩 Rarely Vulnerable 30 days or less a year

Time-to-Fix in Days



Remediation Rates by Industry (Trend)

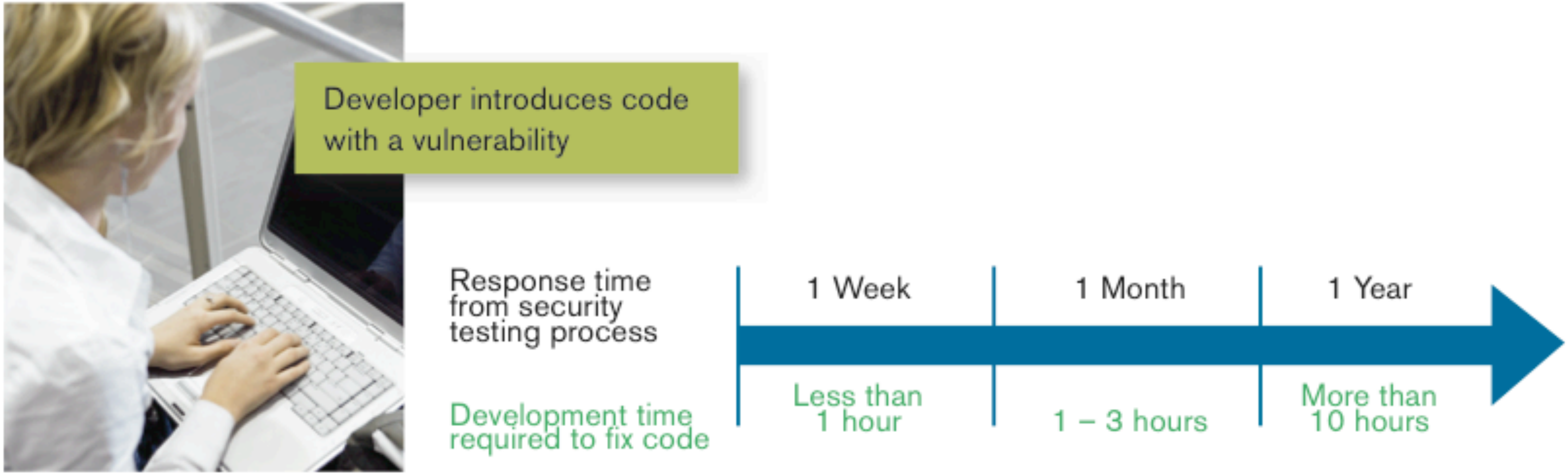


A steady improvement in the percentage of reported vulnerabilities that have been resolved during each of the last three years, which now resides at 53%. Progress!

Why do vulnerabilities go unfixed?

- No one at the organization understands or is responsible for maintaining the code.
- Development group does not understand or respect the vulnerability.
- 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.
- **Feature enhancements are prioritized ahead of security fixes.**

Testing Speed & Frequency Matters



Developer introduces code with a vulnerability

Response time from security testing process

1 Week

1 Month

1 Year

Development time required to fix code

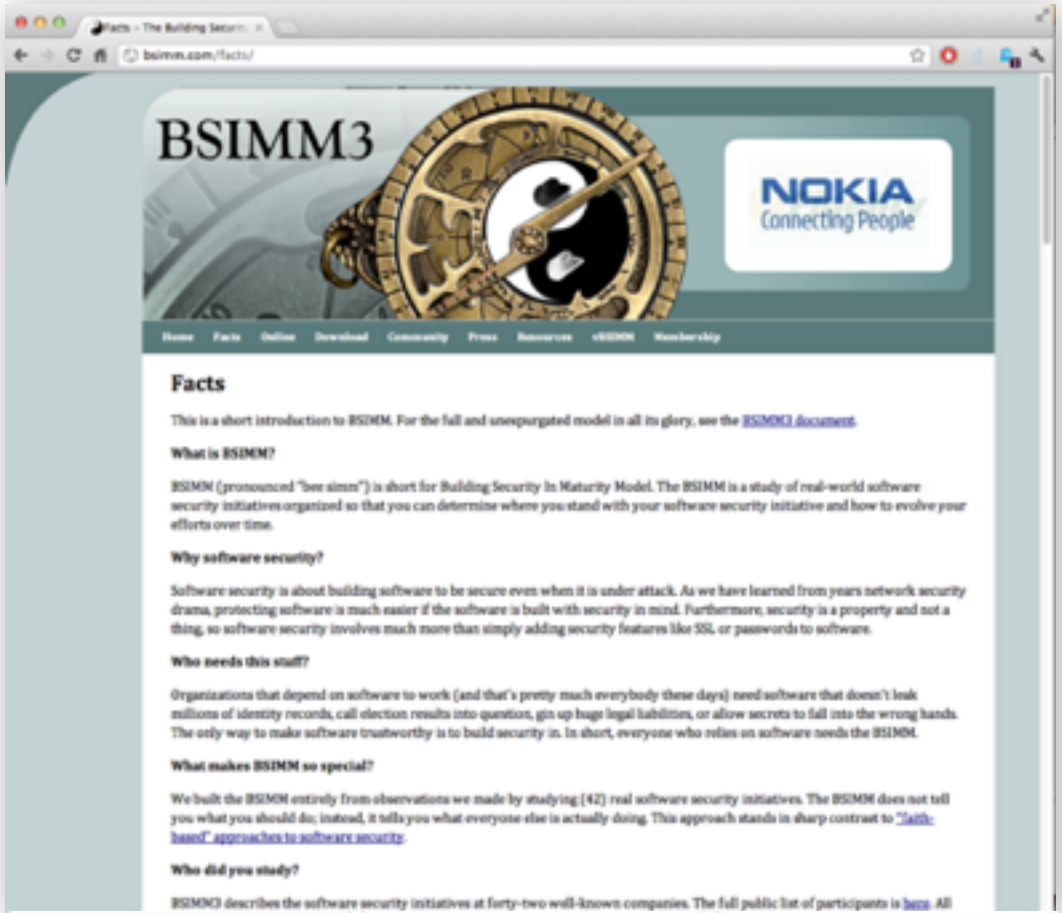
Less than 1 hour

1 – 3 hours

More than 10 hours

Figure 1. Relationship between the time that passes between testing for vulnerabilities and the time required to fix them:

How to develop secure-(enough) software?



Facts

This is a short introduction to BSIMM. For the full and unexpurgated model in all its glory, see the BSIMM3 document.

What is BSIMM?

BSIMM (pronounced "bee simm") is short for Building Security In Maturity Model. The BSIMM is a study of real-world software security initiatives organized so that you can determine where you stand with your software security initiative and how to evolve your efforts over time.

Why software security?

Software security is about building software to be secure even when it is under attack. As we have learned from years network security drama, protecting software is much easier if the software is built with security in mind. Furthermore, security is a property and not a thing, so software security involves much more than simply adding security features like SSL or passwords to software.

Who needs this stuff?

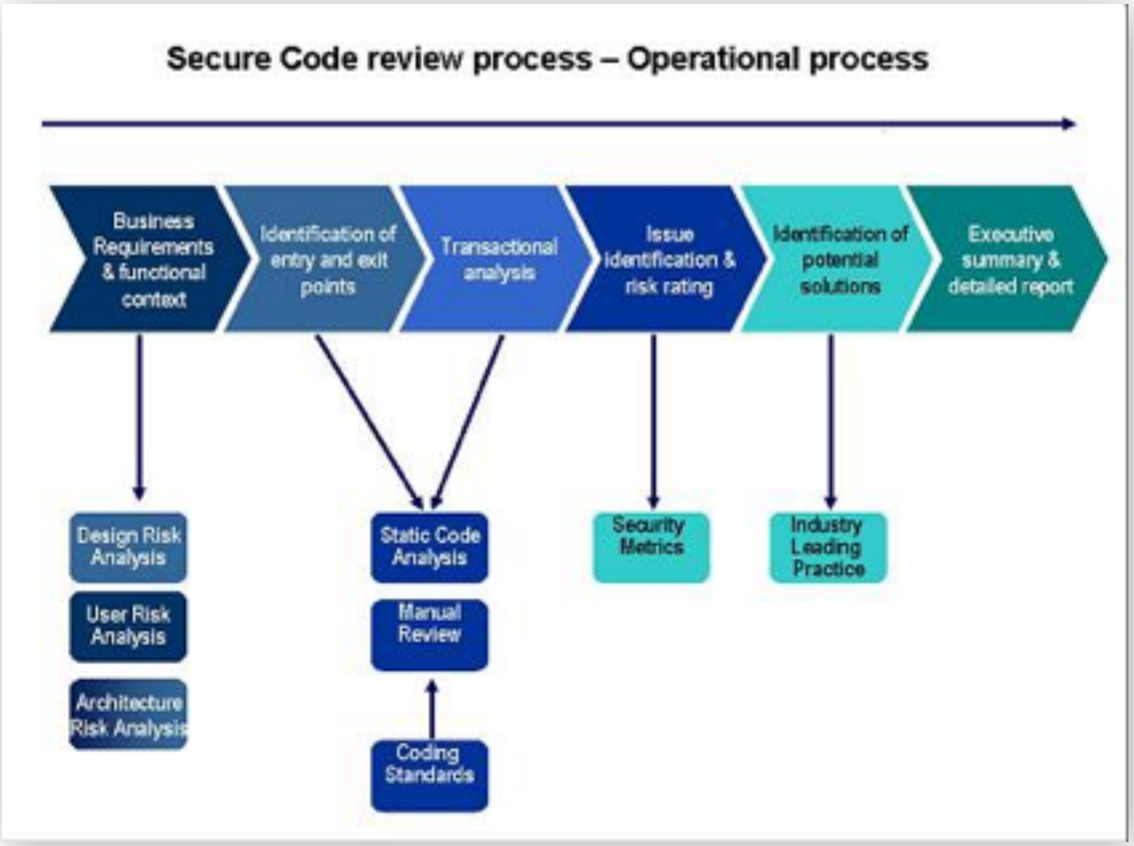
Organizations that depend on software to work (and that's pretty much everybody these days) need software that doesn't leak millions of identity records, call election results into question, gin up huge legal liabilities, or allow secrets to fall into the wrong hands. The only way to make software trustworthy is to build security in. In short, everyone who relies on software needs the BSIMM.

What makes BSIMM so special?

We built the BSIMM entirely from observations we made by studying (42) real software security initiatives. The BSIMM does not tell you what you should do; instead, it tells you what everyone else is actually doing. This approach stands in sharp contrast to "faith-based" approaches to software security.

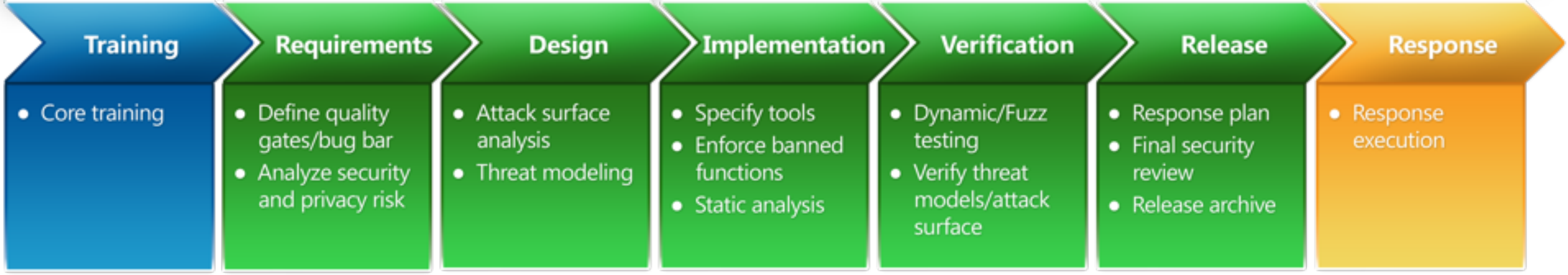
Who did you study?

BSIMM3 describes the software security initiatives at forty-two well-known companies. The full public list of participants is here. All



Secure Code review process – Operational process

Business Requirements & functional context
Identification of entry and exit points
Transactional analysis
Issue identification & risk rating
Identification of potential solutions
Executive summary & detailed report
Design Risk Analysis
User Risk Analysis
Architecture Risk Analysis
Static Code Analysis
Manual Review
Coding Standards
Security Metrics
Industry Leading Practice



Training

- Core training

Requirements

- Define quality gates/bug bar
- Analyze security and privacy risk

Design

- Attack surface analysis
- Threat modeling

Implementation

- Specify tools
- Enforce banned functions
- Static analysis

Verification

- Dynamic/Fuzz testing
- Verify threat models/attack surface

Release

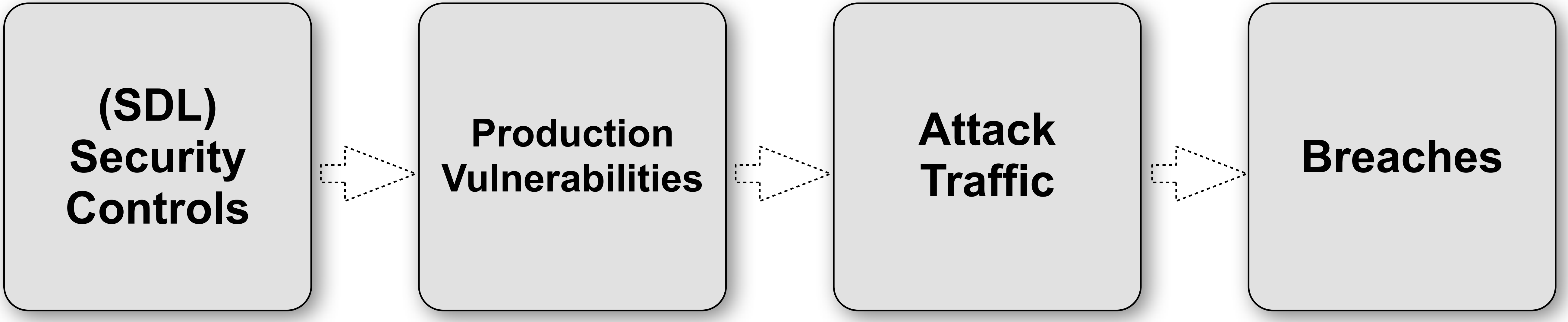
- Response plan
- Final security review
- Release archive

Response

- Response execution

Little-to-No Supporting Data.

Connect the Dots...



(SDL) Security Controls

Production Vulnerabilities

Attack Traffic

Breaches

BSIMM

WhiteHat Security

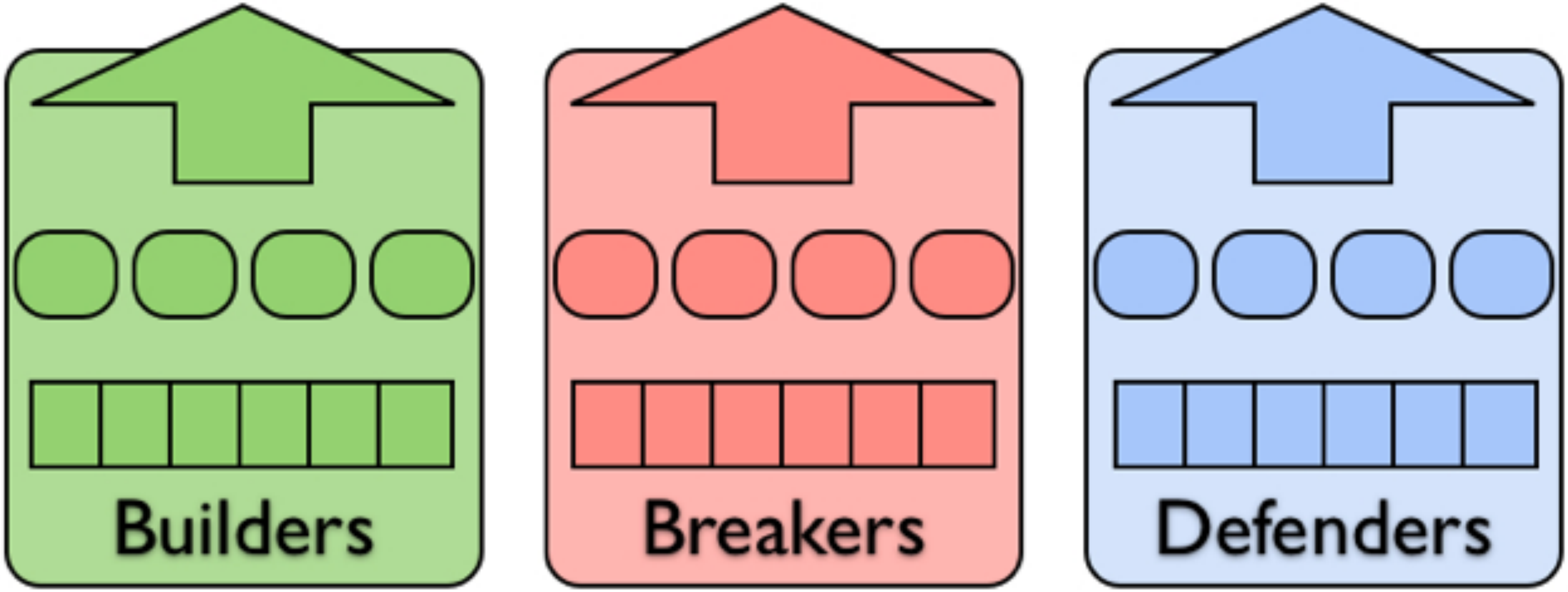
Akamai
IBM

Verizon DBIR
Trustwave

Then we'll start getting some real answers about how to product secure-enough.

The biggest problem in application security today…

The need for qualified people.



Builders

Gary McGraw (CTO, Cigital) says roughly 2% of all programmers should be software security pros, or “Builders” in our case. Gary, through a project called BSIMM, arrived at 2% by surveying dozens of software security programs among large companies and measuring what they do.

Worldwide programmer population: 17 million

We’ll need <u>340,000</u> “Builders”

Breakers

We’ll use a ratio of 1 “breaker” per to 100 websites. This ratio comes from internal metrics at WhiteHat Security generated from assessment conducted over the last 8 years and encompassing more than 5,000 websites.

“Important” (SSL) website population: 1.2 million

Out of 550 million total websites that should be assessed continuously for vulnerabilities.

We’ll need 12,000 “Breakers”

Defenders

No idea how to begin to estimate the Defender need, but it’ll be in the tens of thousands at least. Considering the vast number of website assets that must be protected, the 1 billion online users who someone needs to ensure are playing nice, and monitoring the serious volume of Web traffic they generate.

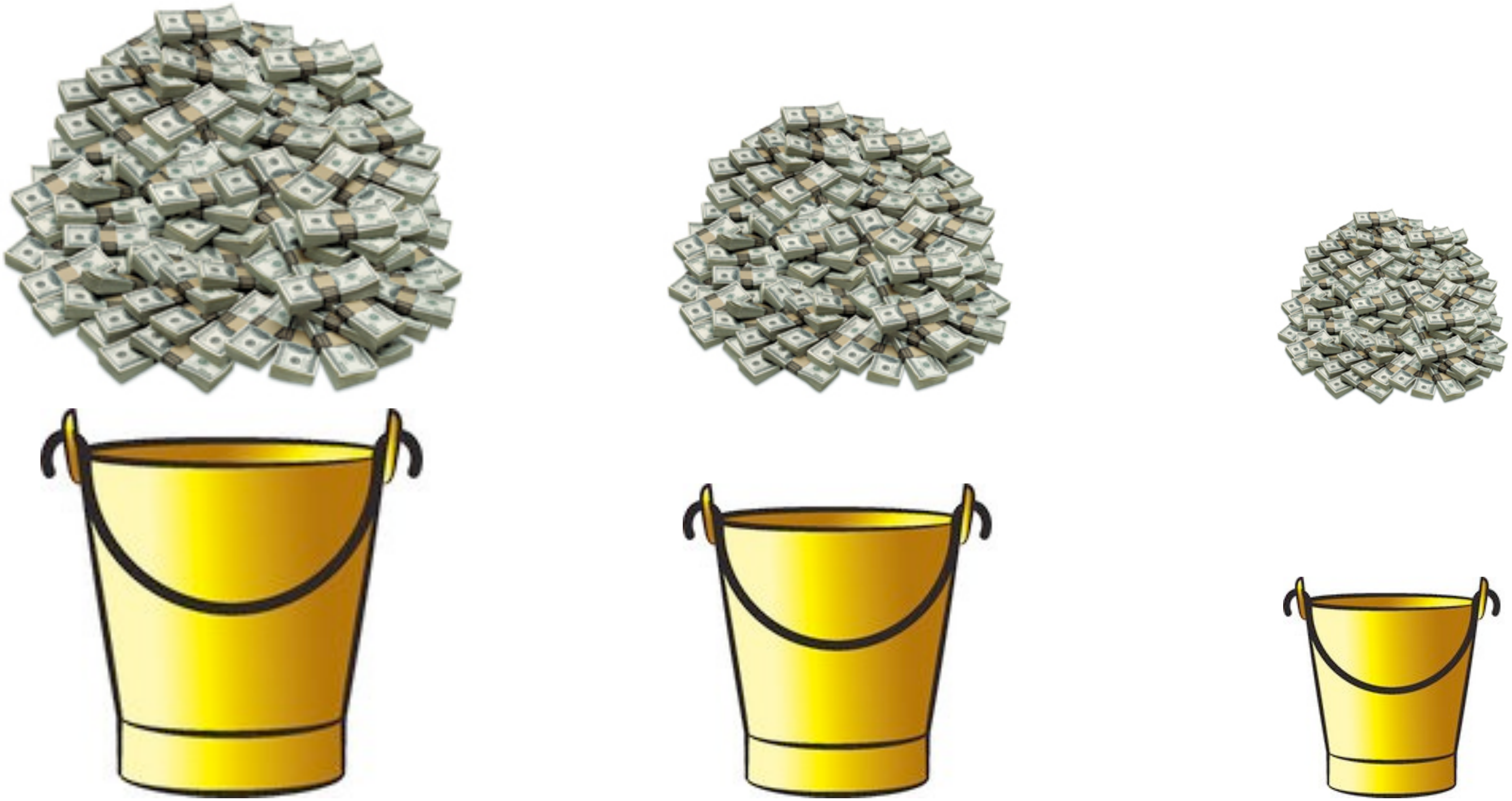
?

Why Do Breaches

(*and vulnerabilities*)

Continue to Happen?

Typical IT Budget Allocation



Applications

Software, development, CRM, ERP, etc.

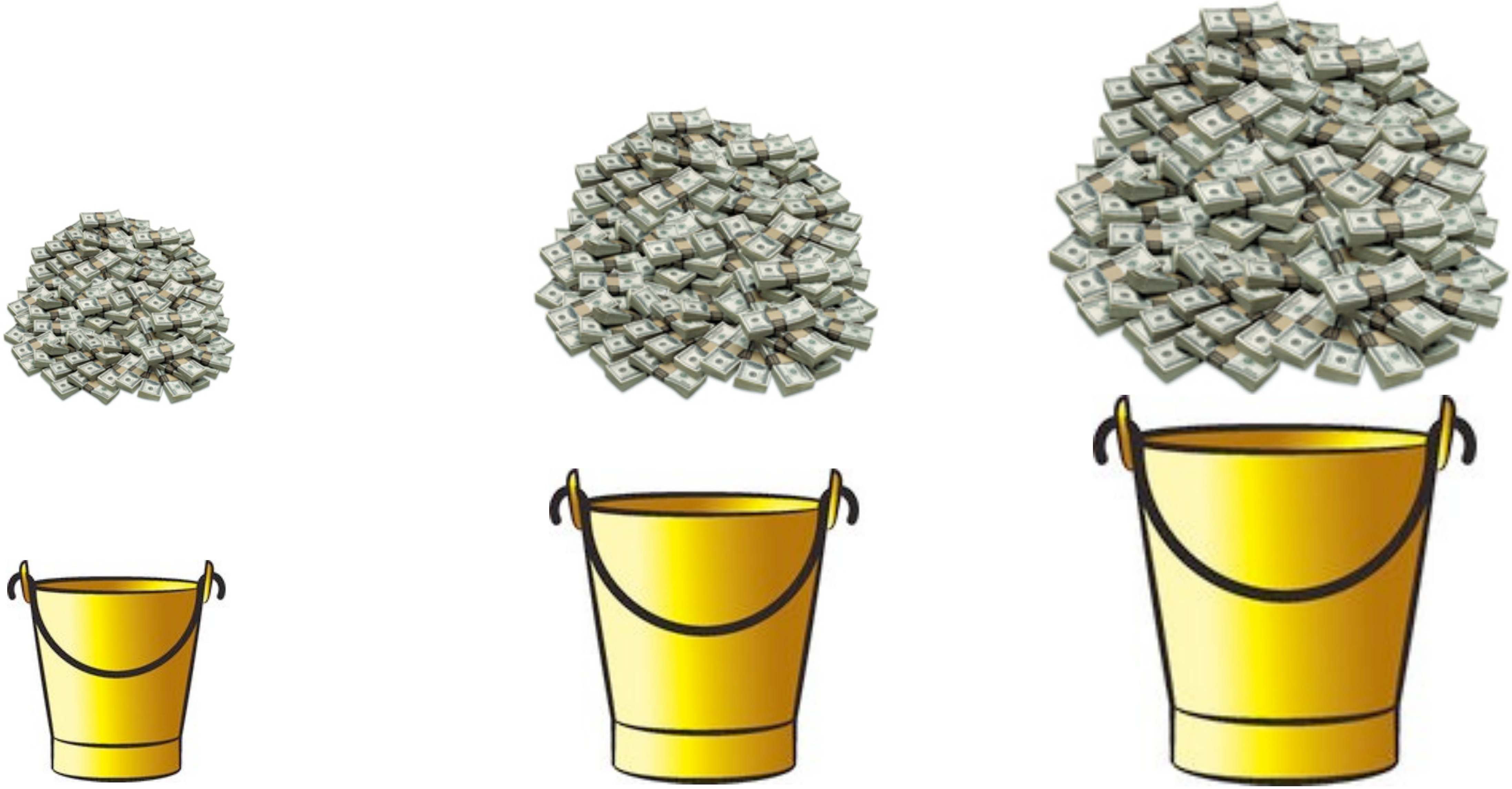
Host

Servers, desktops, laptops, etc.

Network

Routers, switches, network admins, etc.

Typical IT Security Budget



Applications

Software architecture, trainings,testing, etc.

Host

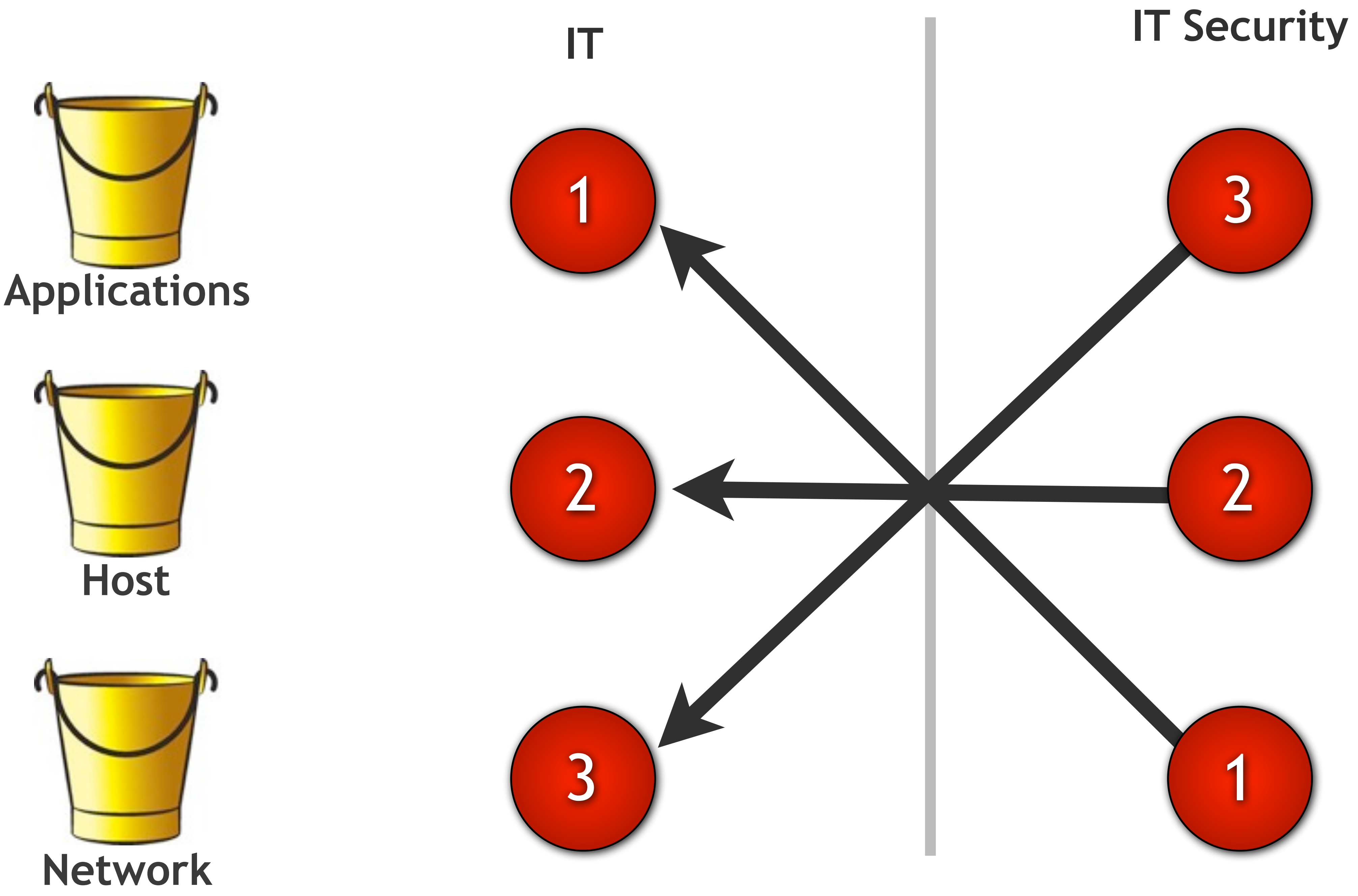
Vulnerability management, system config,patching, etc.

Network

Firewalls, Network IDS, SSL, monitoring, etc.

Budget Prioritization

The biggest line item in [non-security] spending **SHOULD** match the biggest line item in security.



Survey [2010] of IT pros and C-level executives from 450 Fortune 1000 companies (FishNet Security)...

*"Nearly 70% [of those surveyed] say **mobile computing** is the biggest threat to security today, closely followed by **social networks** (68%), and **cloud computing platforms** (35%). Around 65% rank mobile computing the top threat in the next two years, and 62% say cloud computing will be the biggest threat, bumping social networks."*

The report goes on to say...

*"45% say **firewalls** are their priority security purchase, followed by **antivirus** (39%), and **authentication** (31%) and **anti-malware tools** (31%)."*

http://www.darkreading.com/security-services/167801101/security/perimeter-security/227300116/index.html

Big Picture

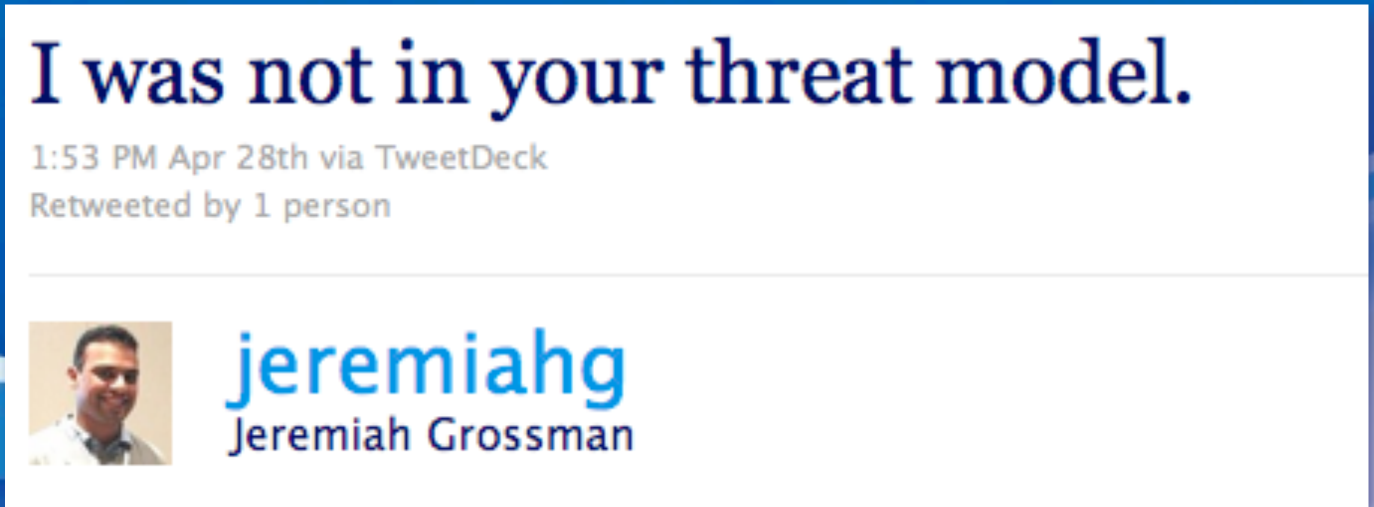
"Market-sizing estimates for network security range anywhere from $5-8bn, whereas our calculation for the aggregate application security market is about $444m. Despite the spending boost on application security mandated by the Payment Card Industry Data Security Standards (PCI-DSS), it's still not commensurate with the demonstrated level of risk."

The Application Security Spectrum (The 451 Group)

"...we expect this revenue will grow at a CAGR of 23% to reach $1bn by 2014."

Thank You!

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



I was not in your threat model.

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

jeremiahg
Jeremiah Grossman