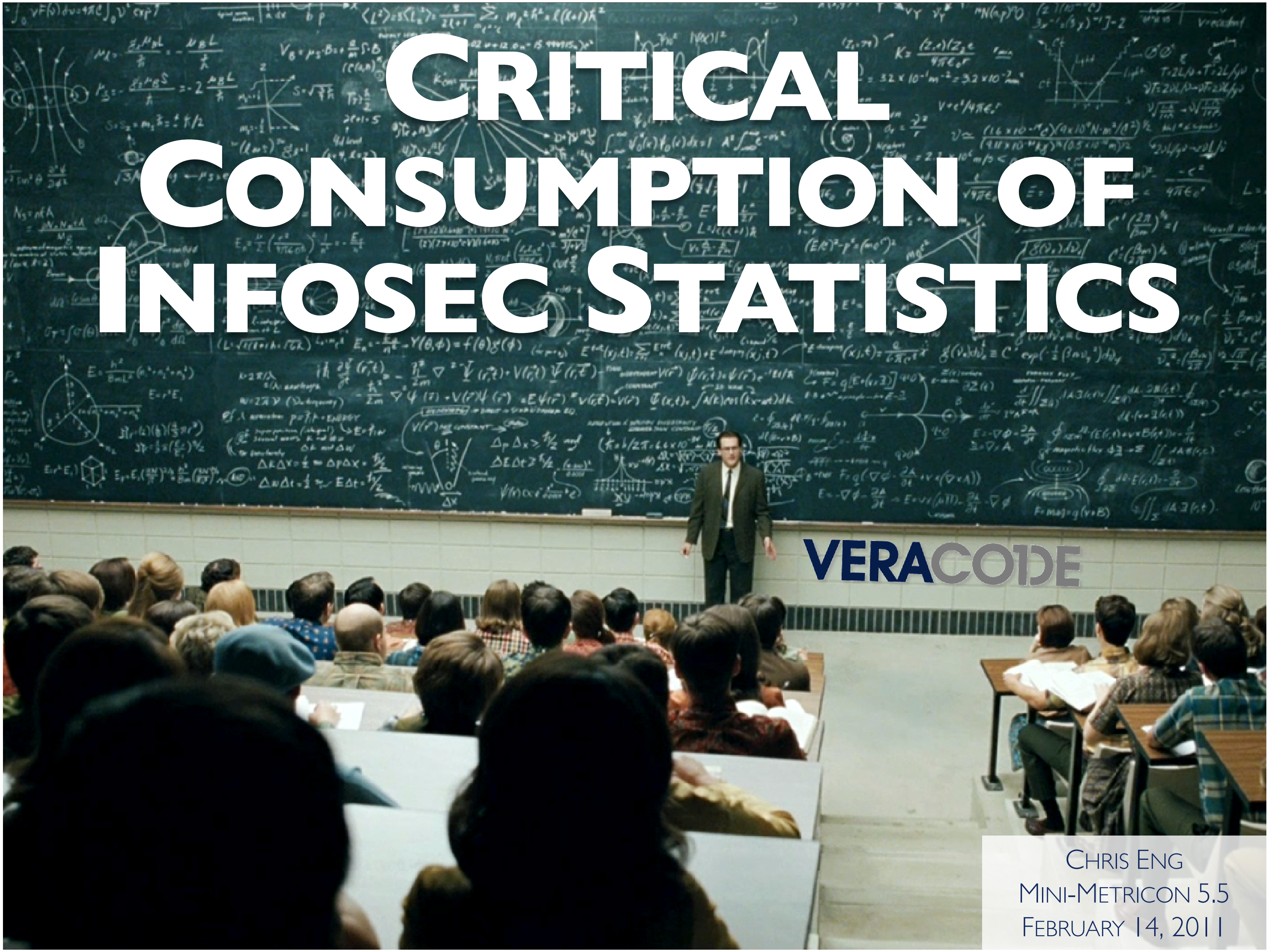


Critical Consumption of Infosec Statistics

VERACODE

Chris Eng

Mini-Metricon 5.5

February 14, 2011

Borderless security
Ernst & Young's 2010
Global Information Security Survey

SYMANTEC ENTERPRISE SECURITY

Symantec.
Confidence in a connected world.

Symantec Global Internet
Security Threat Report
Trends for 2009
Volume XV, Published April 2010

IBM X-Force® 2010
Mid-Year Trend and Risk Report
August 2010

WhiteHat SECURITY

WhiteHat Website Security Statistic Report

Fall 2010, 10th Edition – Industry Benchmarks

Executive Summary

"How are we doing?" That's the question on the mind of many executives and security practitioners whether they have recently implemented an application security program, or already have a well-established plan in place. The executives within these organizations want to know if the resources they have invested in source code reviews, threat modeling, developer training, security tools, etc. are making a measurable difference in reducing the risk of website compromise, which is by no means guaranteed. They want to know if their online business is truly more secure or less secure than industry peers. If above average, they may praise their team's efforts and promote their continued success. On the other hand, if the organization is a security laggard, this is cause for concern and action.

Every organization needs to know where it stands, especially against its adversaries. Verizon Business' 2010 Data Breach Investigations Report (DBIR), a study conducted in cooperation with the United States Secret Service, provides insight. The report analyzes over 141 confirmed data breaches from 2009 which resulted in the compromise of 143 million records. To be clear, this data set is restricted to incidents of a "data" breach, which is different than those only resulting in financial loss. Either way, the data is overwhelming. 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 – "websites" for short. The attack vector of choice was SQL Injection, typically a vulnerability that can't readily be "patched," and used to install customized malware.

As the Verizon DBIR describes, the majority of breach victims are targets of opportunity, as opposed to targets of choice. Directly in the crosshairs are the Financial Services, Hospitality, and Retail industries. Victimized organizations are selected because their security posture is weaker than others and the data they possess can be converted into cash, namely payment card data and intellectual property. As such, organizations are strongly encouraged to determine if they are similar potential targets of opportunity in these industries, have a relatively weak or unknown security posture, and the data they hold is similarly attractive. This is a key point because perfect security may not be necessary to avoid becoming another Verizon DBIR statistical data point.

There are of course many published examples in Web
victim was a target of choice. Curr
networks, more

2,000+ websites under management

Web security is a moving target and enterprises need timely information about the latest attack trends, how they can best defend their websites, and gain visibility into their vulnerability lifecycle. Through its Software-as-a-Service (SaaS) offering, WhiteHat Sentinel, WhiteHat Security is uniquely positioned to deliver the knowledge and solutions that organizations need to protect their brands, attain PCI compliance and avert costly breaches.

The WhiteHat Website Security Statistics Report provides a one-of-a-kind perspective on the state of website security and the issues that organizations must address to safely conduct business online. WhiteHat has been publishing the report, which highlights the top vulnerabilities, tracks vertical market trends and identifies new attack techniques, since 2006.

The WhiteHat Security ... rt presents
a St ... t website
... re than
... ement.
... pert analysis
... Hat's report is
... focus solely
... abilities in
... unique to an
... websites.

verizon

SOURCE: EXTERNAL
TYPE: ORGANIZED CRIMINAL GROUP
ORIGIN: EASTERN EUROPE
AGENT
CATEGORY: HACKING
TYPE: SQL INJECTION, USE OF STOLEN CREDENTIALS
PATH: WEB APPLICATION
ACTION
CATEGORY: MALWARE
TYPE: PACKET SNIFFER, BACKDOOR
PATH: INSTALLED/INJECTED BY REMOTE ATTACKER
ASSET
TYPE: DATABASE SERVER
DATA: PAYMENT CARD DATA; INTELLECTUAL PROPERTY
ATTRIBUTE
TYPE: INTEGRITY, CONFIDENTIALITY

BREACH
IONS REPORT
cooperation with the United States Secret Service.

Cisco 2010 Annual Security Report
Highlighting global security threats and trends

cisco.

VERACODE
VERACODE

VOLUME 2

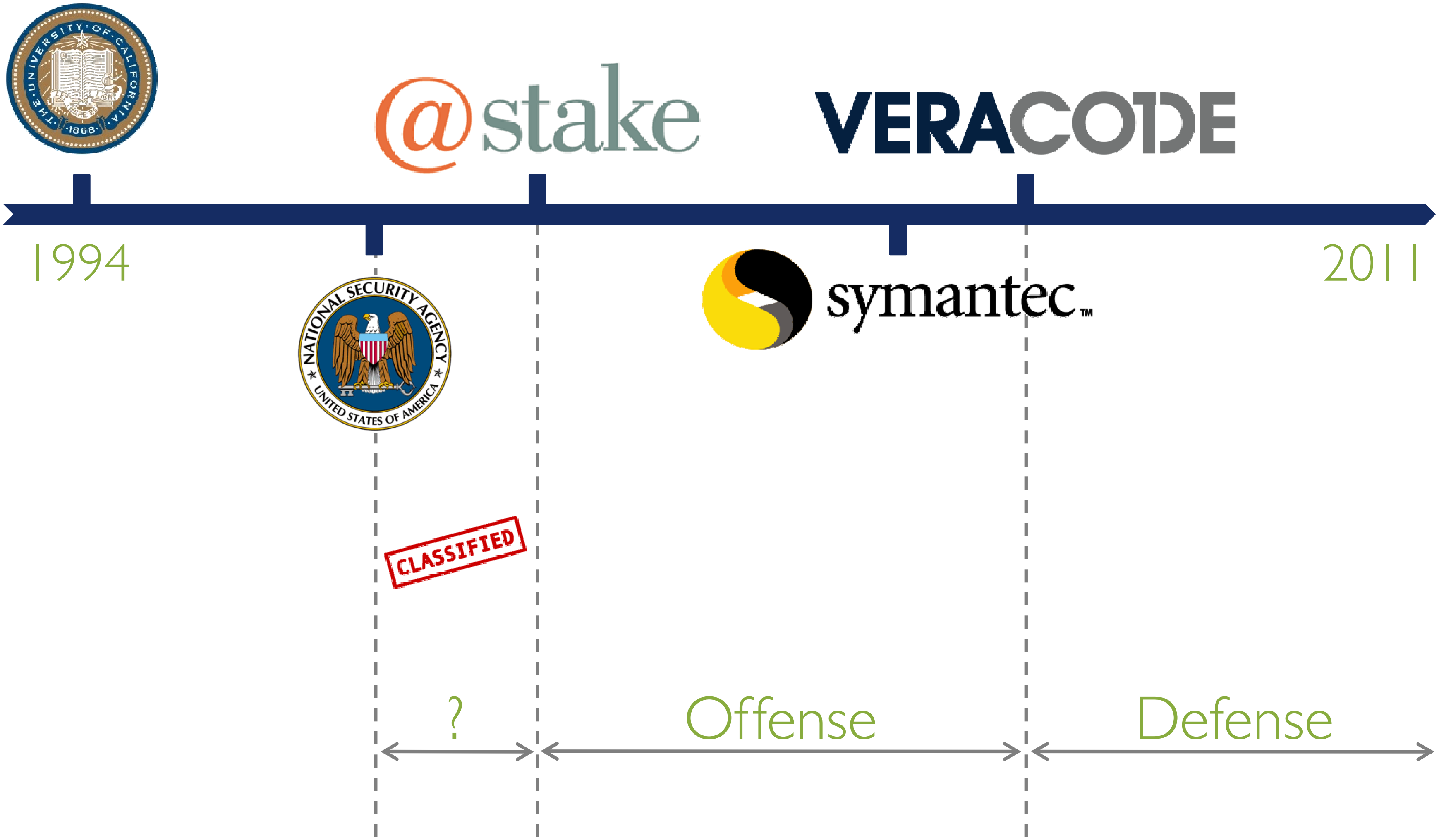
State of Software
Security Report

The Intractable Problem of Insecure Software
September 22, 2010

Global
Security Report
2011

Trustwave®
SpiderLabs®

Obligatory Bio



THE UNIVERSITY OF CALIFORNIA 1868
@stake
VERACODE
NATIONAL SECURITY AGENCY UNITED STATES OF AMERICA
symantec.
CLASSIFIED

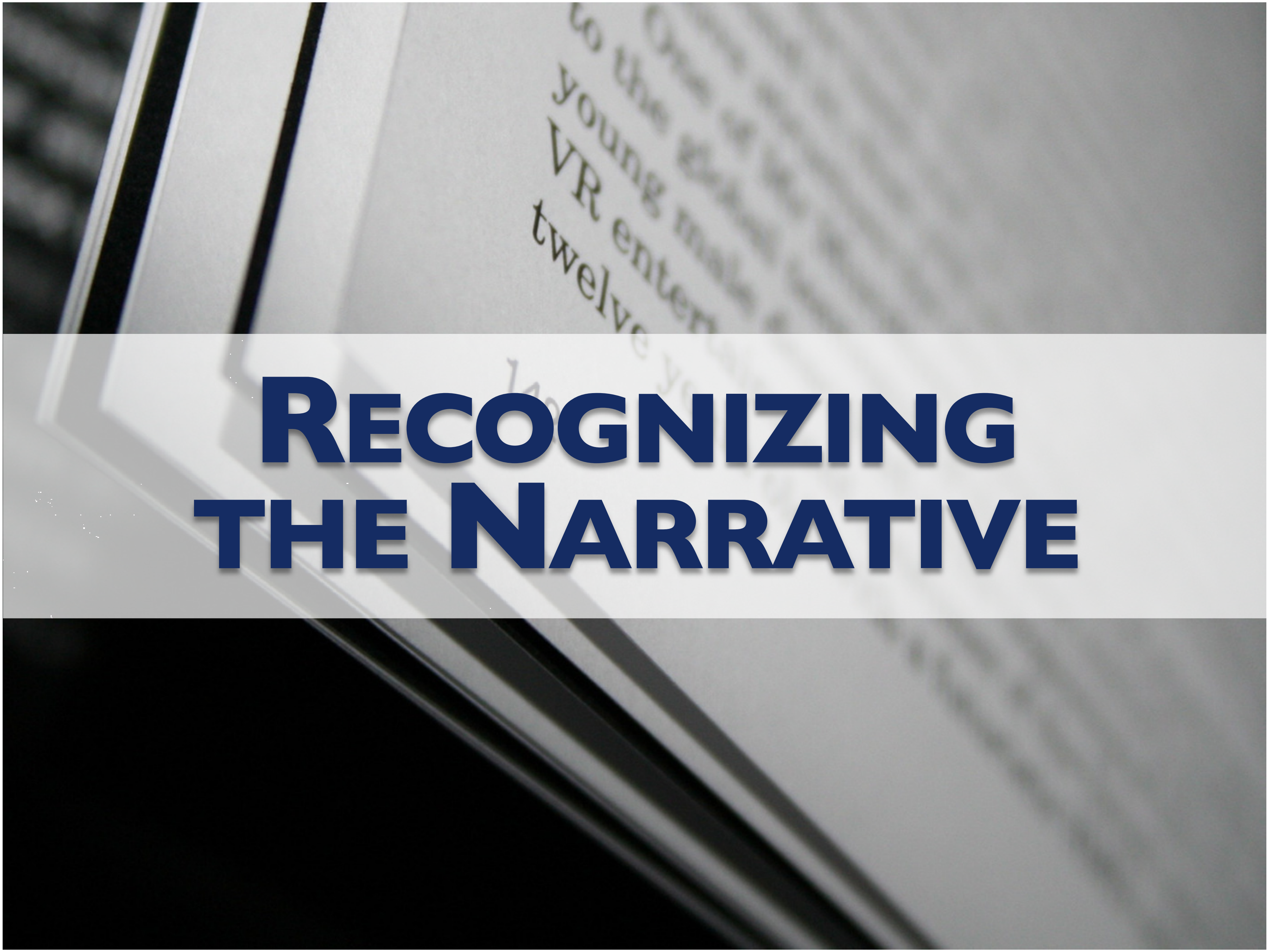
1994

2011

?

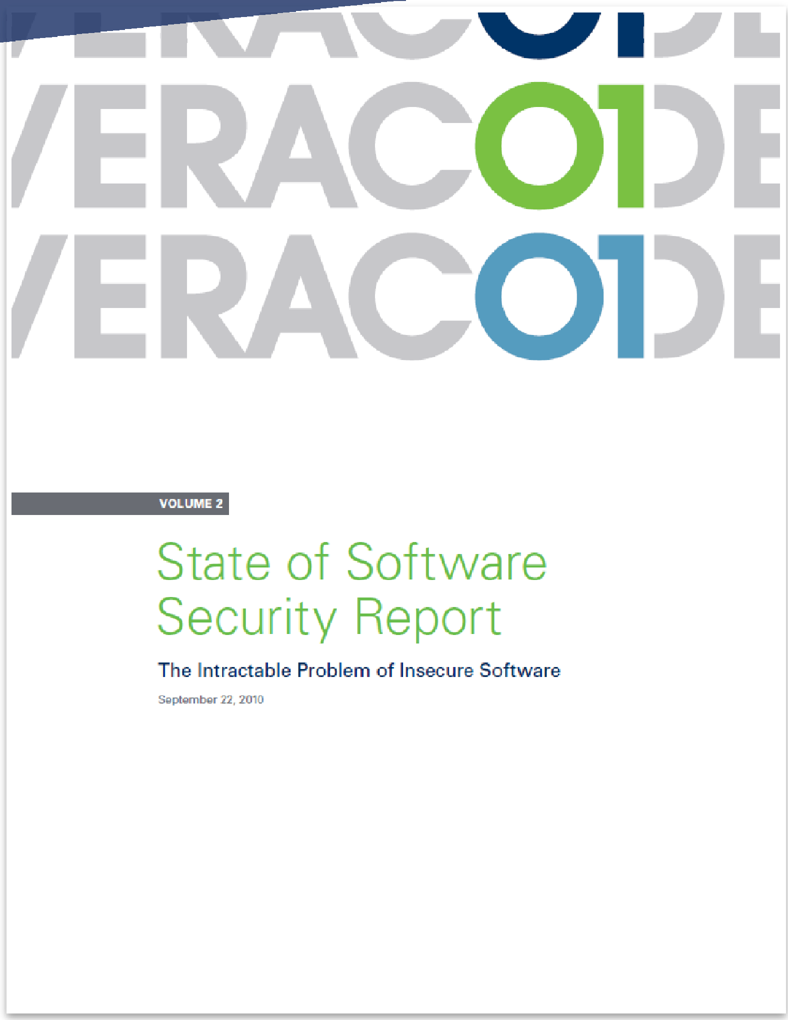
Offense

Defense



Recognizing the Narrative

Sample Narrative



VOLUME 2

State of Software Security Report

The Intractable Problem of Insecure Software

September 22, 2010

- More than half of all software failed to achieve acceptable level of security
- 3rd-party applications had lowest security quality
- No single method of testing is adequate

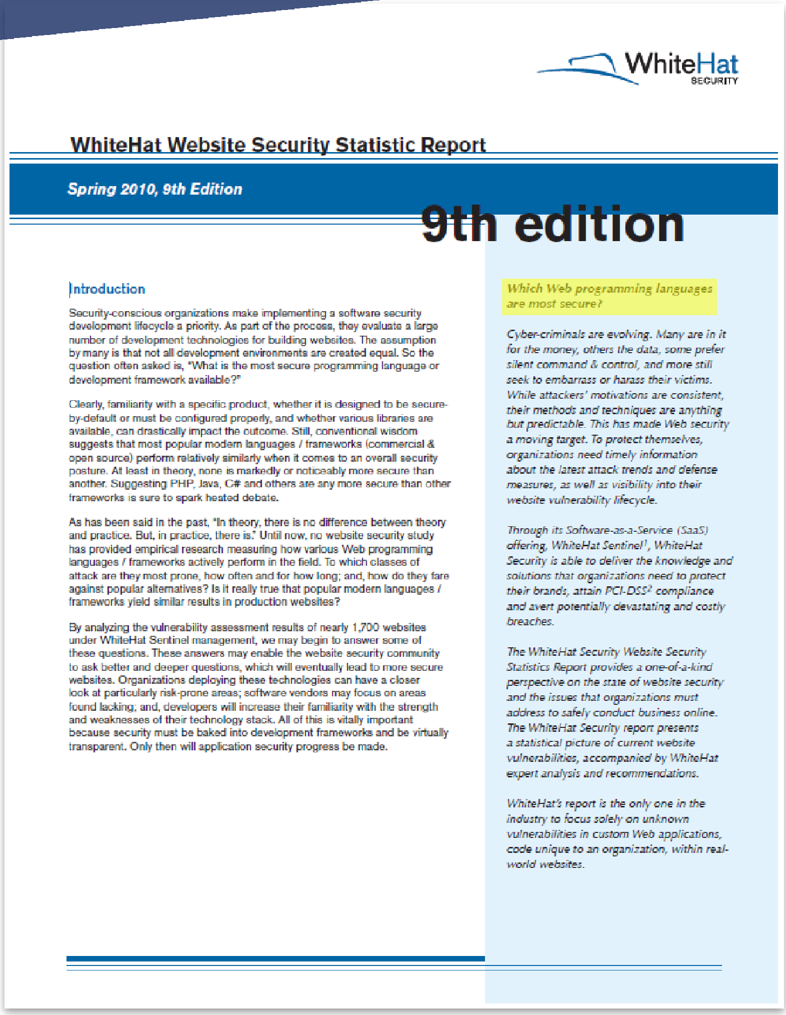
Sample Narrative

Global
Security Report
2011

Trustwave
SpiderLabs

- 2010 incident response investigations
- Attack vector evolution
- 11 strategic initiatives for 2011

Sample Narrative



WhiteHat SECURITY

WhiteHat Website Security Statistic Report

Spring 2010, 9th Edition

9th edition

Introduction

Security-conscious organizations make implementing a software security development lifecycle a priority. As part of the process, they evaluate a large number of development technologies for building websites. The assumption by many is that not all development environments are created equal. So the question often asked is, "What is the most secure programming language or development framework available?"

Clearly, familiarity with a specific product, whether it is designed to be secure-by-default or must be configured properly, and whether various libraries are available, can drastically impact the outcome. Still, conventional wisdom suggests that most popular modern languages / frameworks (commercial & open source) perform relatively similarly when it comes to an overall security posture. At least in theory, none is markedly or noticeably more secure than another. Suggesting PHP, Java, C# and others are any more secure than other frameworks is sure to spark heated debate.

As has been said in the past, "In theory, there is no difference between theory and practice. But, in practice, there is." Until now, no website security study has provided empirical research measuring how various Web programming languages / frameworks actively perform in the field. To which classes of attack are they most prone, how often and for how long; and, how do they fare against popular alternatives? Is it really true that popular modern languages / frameworks yield similar results in production websites?

By analyzing the vulnerability assessment results of nearly 1,700 websites under WhiteHat Sentinel management, we may begin to answer some of these questions. These answers may enable the website security community to ask better and deeper questions, which will eventually lead to more secure websites. Organizations deploying these technologies can have a closer look at particularly risk-prone areas; software vendors may focus on areas found lacking; and, developers will increase their familiarity with the strength and weaknesses of their technology stack. All of this is vitally important because security must be baked into development frameworks and be virtually transparent. Only then will application security progress be made.

Which Web programming languages are most secure?

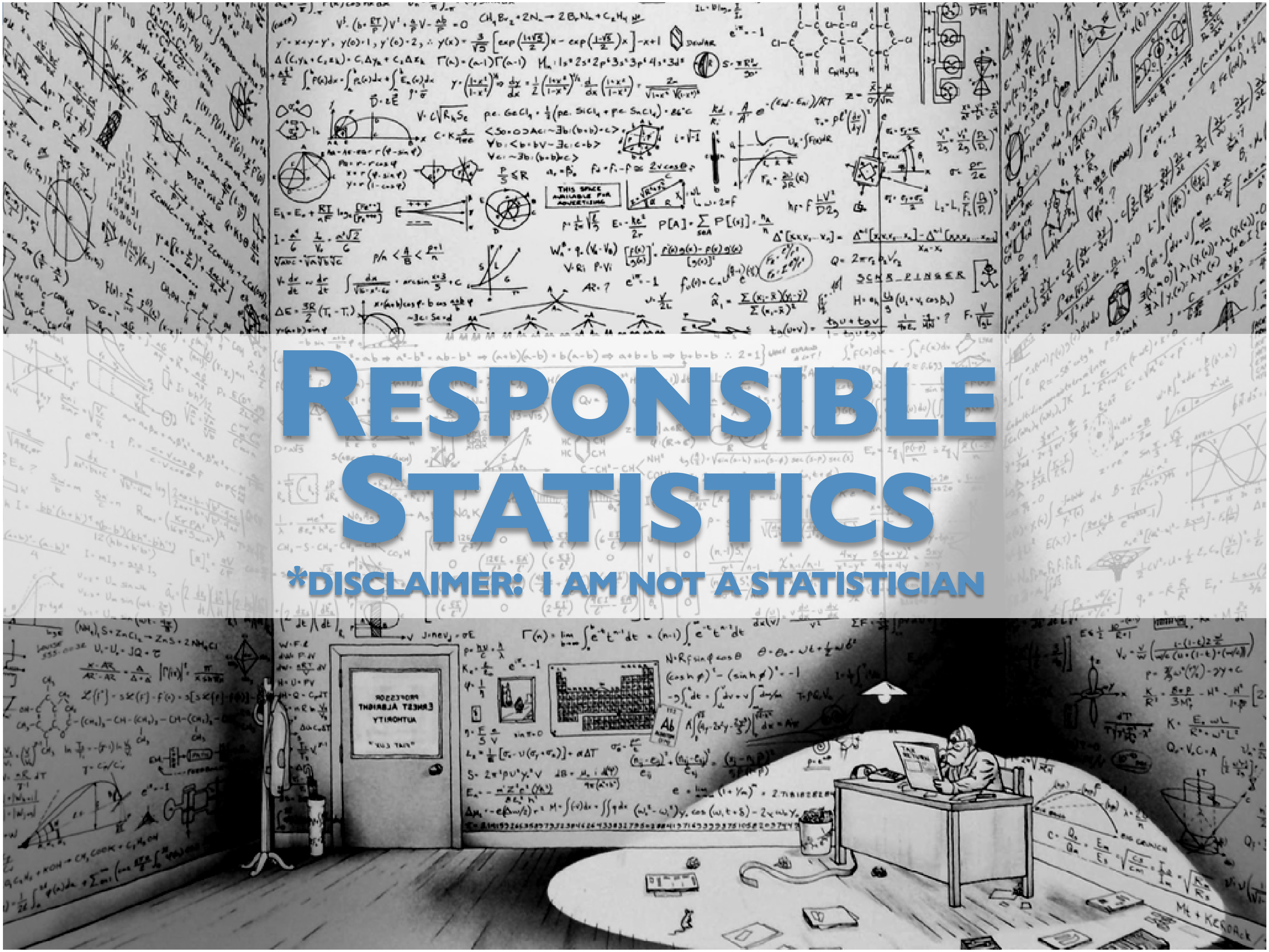
Cyber-criminals are evolving. Many are in it for the money, others the data, some prefer silent command & control, and more still seek to embarrass or harass their victims. While attackers' motivations are consistent, their methods and techniques are anything but predictable. This has made Web security a moving target. To protect themselves, organizations need timely information about the latest attack trends and defense measures, as well as visibility into their website vulnerability lifecycle.

Through its Software-as-a-Service (SaaS) offering, WhiteHat Sentinel[1], WhiteHat Security is able to deliver the knowledge and solutions that organizations need to protect their brands, attain PCI-DSS[2] compliance and avert potentially devastating and costly breaches.

The WhiteHat Security Website Security Statistics Report provides a one-of-a-kind perspective on the state of website security and the issues that organizations must address to safely conduct business online. The WhiteHat Security report presents a statistical picture of current website vulnerabilities, accompanied by WhiteHat expert analysis and recommendations.

WhiteHat's report is the only one in the industry to focus solely on unknown vulnerabilities in custom Web applications, code unique to an organization, within real-world websites.

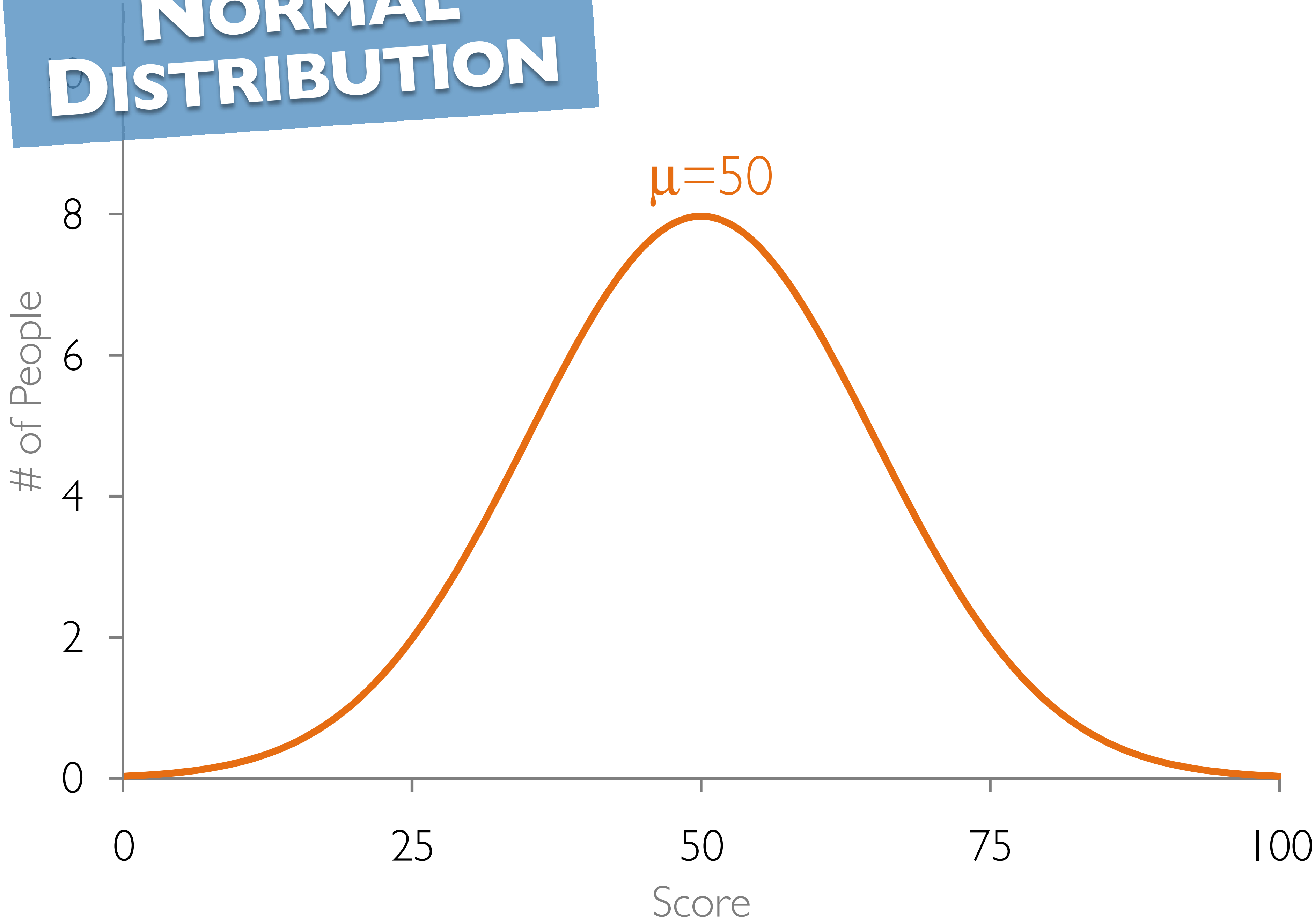
– Which web programming languages are most secure?



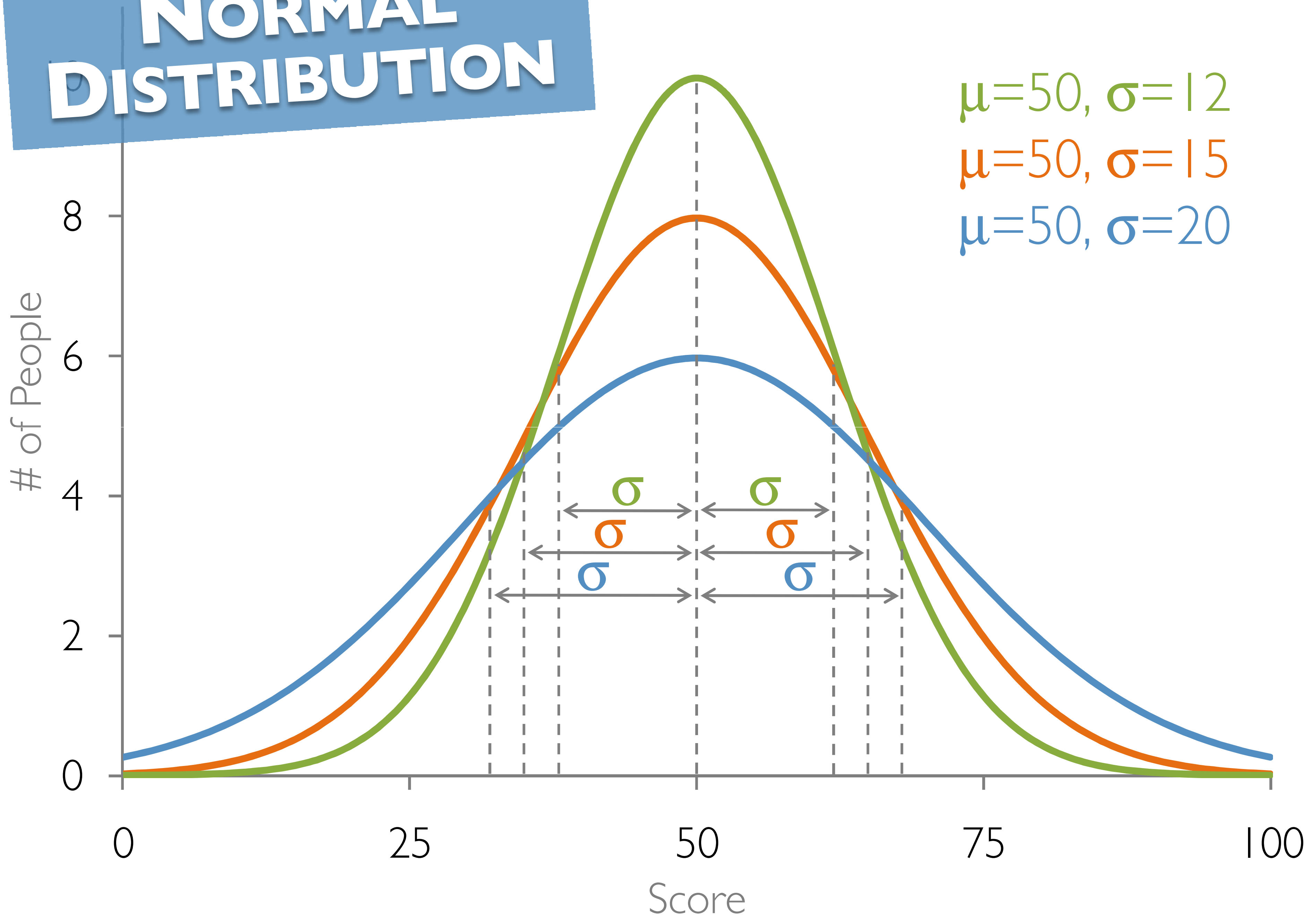
Responsible Statistics

*DISCLAIMER: I AM NOT A STATISTICIAN

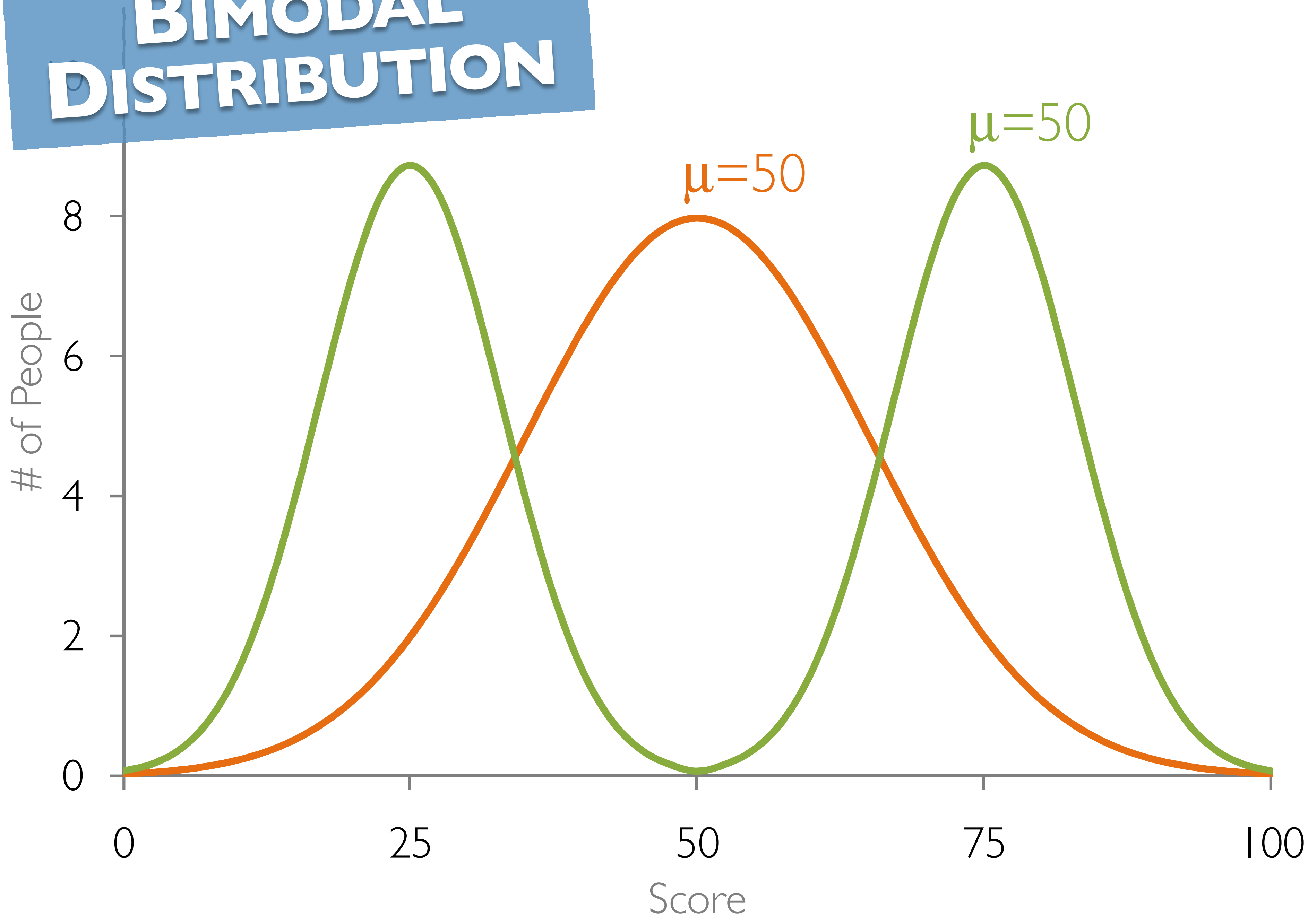
Normal Distribution



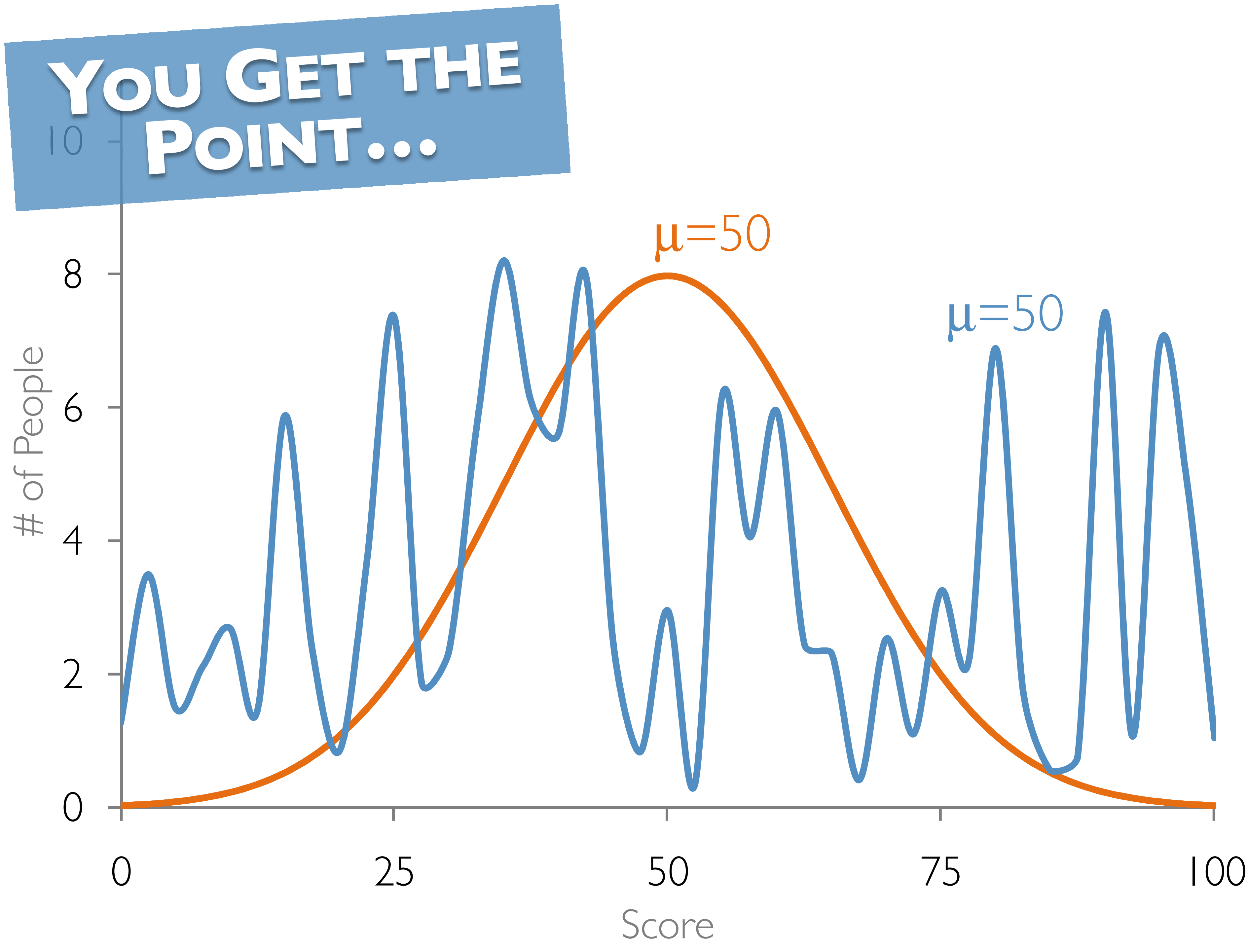
Normal Distribution



Bimodal Distribution



You Get the Point...



WhiteHat SECURITY

WhiteHat Website Security Statistic Report

Spring 2010, 9th Edition

9th edition

Introduction

Security-conscious organizations make implementing a software security development lifecycle a priority. As part of the process, they evaluate a large number of development technologies for building websites. The assumption by many is that not all development environments are created equal. So the question often asked is, "What is the most secure programming language or development framework available?"

Clearly, familiarity with a specific product, whether it is designed to be secure-by-default or must be configured properly, and whether various libraries are available, can drastically impact the outcome. Still, conventional wisdom suggests that most popular modern languages / frameworks (commercial & open source) perform relatively similarly when it comes to an overall security posture. At least in theory, none is markedly or noticeably more secure than another. Suggesting PHP, Java, C# and others are any more secure than other frameworks is sure to spark heated debate.

As has been said in the past, "In theory, there is no difference between theory and practice. But, in practice, there is." Until now, no website security study has provided empirical research measuring how various Web programming languages / frameworks actively perform in the field. To which classes of attack are they most prone, how often and for how long; and, how do they fare against popular alternatives? Is it really true that popular modern languages / frameworks yield similar results in production websites?

By analyzing the vulnerability assessment results of nearly 1,700 websites under WhiteHat Sentinel management, we may begin to answer some of these questions. These answers may enable the website security community to ask better and deeper questions, which will eventually lead to more secure websites. Organizations deploying these technologies can have a closer look at particularly risk-prone areas; software vendors may focus on areas found lacking; and, developers will increase their familiarity with the strength and weaknesses of their technology stack. All of this is vitally important because security must be baked into development frameworks and be virtually transparent. Only then will application security progress be made.

Which Web programming languages are most secure?

Cyber-criminals are evolving. Many are in it for the money, others the data, some prefer silent command & control, and more still seek to embarrass or harass their victims. While attackers' motivations are consistent, their methods and techniques are anything but predictable. This has made Web security a moving target. To protect themselves, organizations need timely information about the latest attack trends and defense measures, as well as visibility into their website vulnerability lifecycle.

Through its Software-as-a-Service (SaaS) offering, WhiteHat Sentinel[1], WhiteHat Security is able to deliver the knowledge and solutions that organizations need to protect their brands, attain PCI-DSS[2] compliance and avert potentially devastating and costly breaches.

The WhiteHat Security Website Security Statistics Report provides a one-of-a-kind perspective on the state of website security and the issues that organizations must address to safely conduct business online. The WhiteHat Security report presents a statistical picture of current website vulnerabilities, accompanied by WhiteHat expert analysis and recommendations.

WhiteHat's report is the only one in the industry to focus solely on unknown vulnerabilities in custom Web applications, code unique to an organization, within real-world websites.

"At an average of 44 days, SQL Injection vulnerabilities were fixed the fastest on Microsoft ASP Classic websites, just ahead of Perl (PL) at 45 days."

Global
Security Report
2011

Trustwave
SpiderLabs

> "Our analysis reveals that, on average, a lapse of 156 days occurred between an initial breach and detection of that incident."

	Average
C/C++	1.07
ColdFusion	8.90
Java	0.56
.NET	0.72

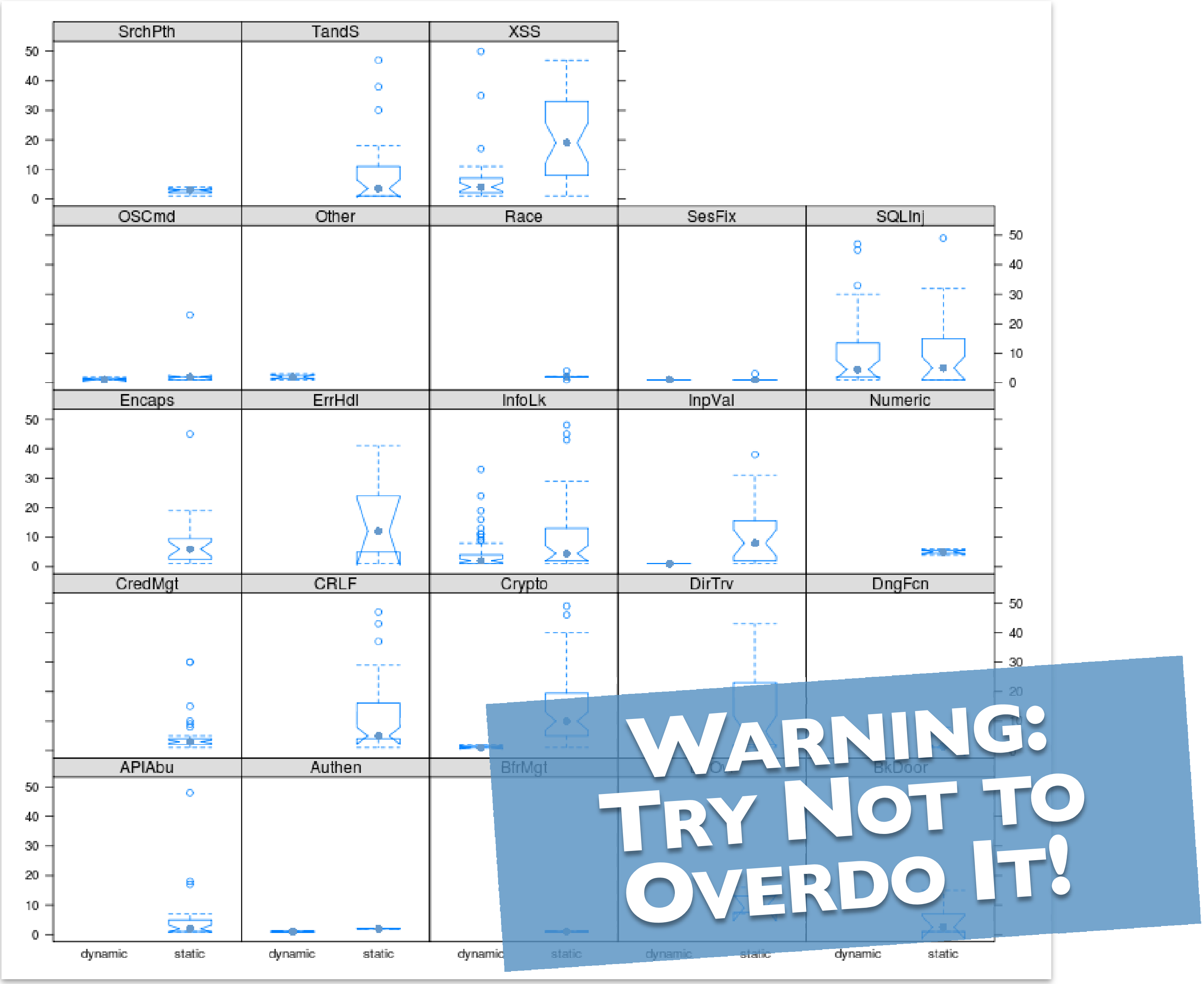
Flaws per KLOC, by language

Is This Useful to Anyone?

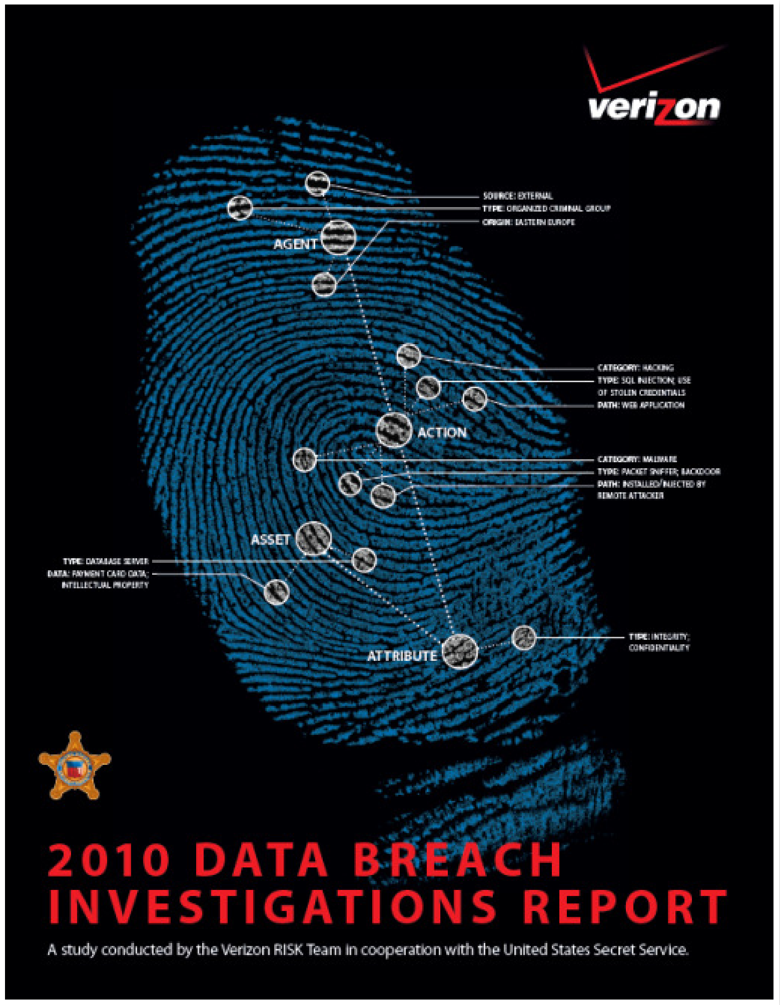
	Average	1st Q'tile	Median	3rd Q'tile
C/C++	1.07	0.01	0.03	0.13
ColdFusion	8.90	1.83	5.28	11.98
Java	0.56	0.01	0.03	0.16
.NET	0.72	0.01	0.04	0.16

Flaws per KLOC, by language

Anything Else
We Should
Show here?



WARNING:
TRY NOT TO
OVERDO IT!


verizon
SOURCE: EXTERNAL
TYPE: ORGANIZED CRIMINAL GROUP
ORIGIN: EASTERN EUROPE
AGENT
CATEGORY: HACKING
TYPE: SQL INJECTION; USE OF STOLEN CREDENTIALS
PATH: WEB APPLICATION
ACTION
CATEGORY: MALWARE
TYPE: PACKET SNIFFER; BACKDOOR
PATH: INSTALLED/INJECTED BY REMOTE ATTACKER
ASSET
TYPE: DATABASE SERVER
DATA: PAYMENT CARD DATA; INTELLECTUAL PROPERTY
ATTRIBUTE
TYPE: INTEGRITY; CONFIDENTIALITY

2010 DATA BREACH INVESTIGATIONS REPORT

A study conducted by the Verizon RISK Team in cooperation with the United States Secret Service.

> "The average number of records lost per breach was 1,381,183, the median a scant 1,082, and the standard deviation a whopping 11,283,151."

WhiteHat SECURITY

WhiteHat Website Security Statistic Report

Fall 2010, 10th Edition – Industry Benchmarks

2,000+ websites under management

Executive Summary

"How are we doing?" That's the question on the mind of many executives and security practitioners whether they have recently implemented an application security program, or already have a well-established plan in place. The executives within those organizations want to know if the resources they have invested in source code reviews, threat modeling, developer training, security tools, etc. are making a measurable difference in reducing the risk of website compromise, which is by no means guaranteed. They want to know if their online business is truly more secure or less secure than industry peers. If above average, they may praise their team's efforts and promote their continued success. On the other hand, if the organization is a security laggard, this is cause for concern and action.

Every organization needs to know where it stands, especially against its adversaries. Verizon Business' 2010 Data Breach Investigations Report (DBIR), a study conducted in cooperation with the United States Secret Service, provides insight. The report analyzes over 141 confirmed data breaches from 2009 which resulted in the compromise of 143 million records. To be clear, this data set is restricted to incidents of a "data" breach, which is different than those only resulting in financial loss. Either way, the data is overwhelming. 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 – "websites" for short. The attack vector of choice was SQL Injection, typically a vulnerability that can't readily be "patched," and used to install customized malware.

As the Verizon DBIR describes, the majority of breach victims are targets of opportunity, as opposed to targets of choice. Directly in the crosshairs are the Financial Services, Hospitality, and Retail industries. Victimized organizations are selected because their security posture is weaker than others and the data they possess can be converted into cash, namely payment card data and intellectual property. As such, organizations are strongly encouraged to determine if they are similar potential targets of opportunity in these industries, have a relatively weak or unknown security posture, and the data they hold is similarly attractive. This is a key point because perfect security may not be necessary to avoid becoming another Verizon DBIR statistical data point.

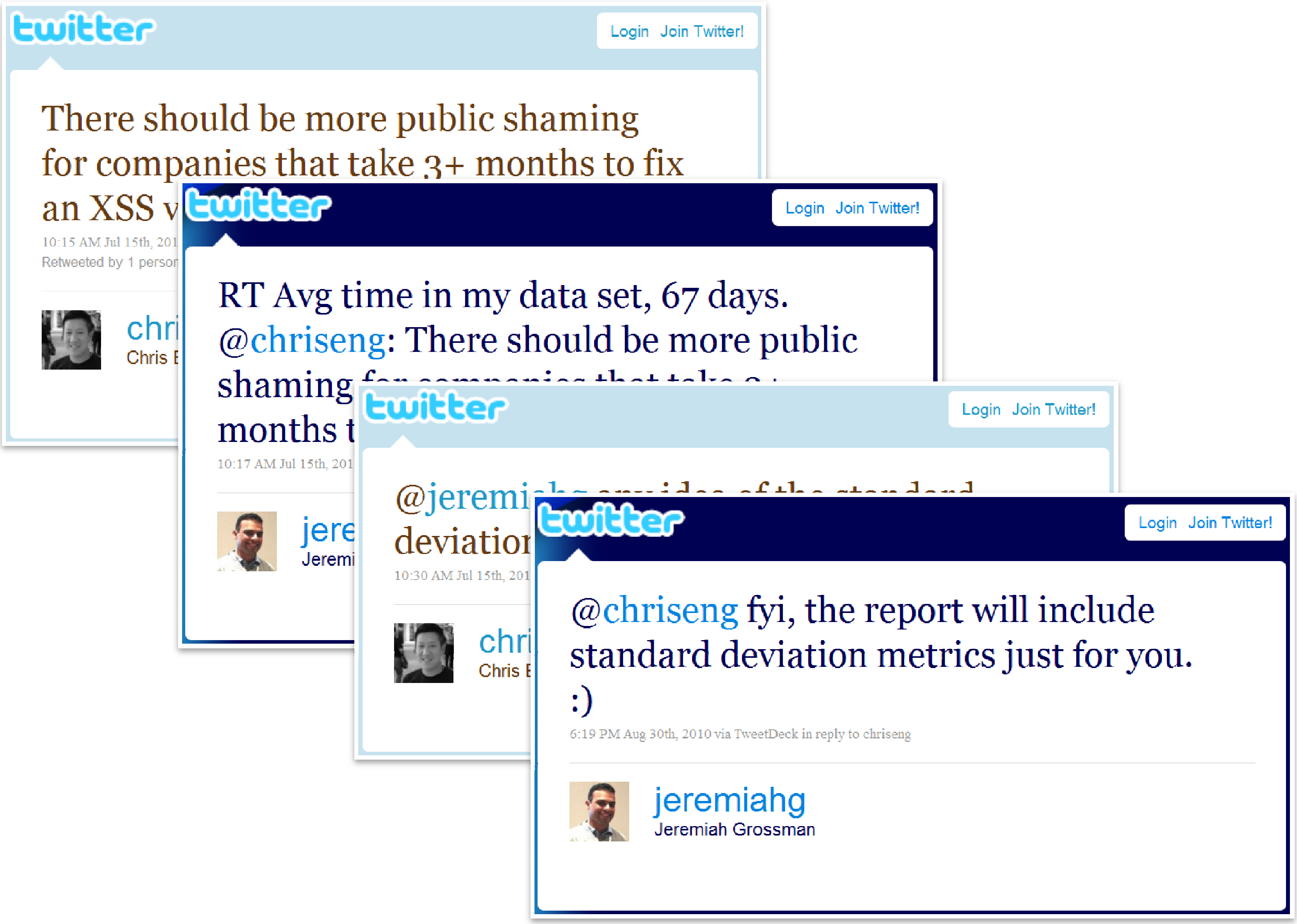
There are of course many published examples in Web security where the victim was a target of choice. Currently, Clickjacking attacks[1] targeting social networks, more specifically Facebook, are rampant. In these attacks, visitors are being tricked into posting unwanted messages to friends and installing malware. There has also been a rise in targeted Cross-Site Scripting attacks, including a notable incident involving Apache.org[2] in which passwords were compromised. Content Spoofing attacks have been aimed at Wired to spoof a Steve Jobs health scare[3]. Sears suffered a similar embarrassment[4] when a fake product listing appeared on the company's website. In an Insufficient Authorization incident involving Anthem Blue Cross Blue Shield, customers' personally identifiable information was exposed[5].

Web security is a moving target and enterprises need timely information about the latest attack trends, how they can best defend their websites, and gain visibility into their vulnerability lifecycle. Through its Software-as-a-Service (SaaS) offering, WhiteHat Sentinel, WhiteHat Security is uniquely positioned to deliver the knowledge and solutions that organizations need to protect their brands, attain PCI compliance and avert costly breaches.

The WhiteHat Website Security Statistics Report provides a one-of-a-kind perspective on the state of website security and the issues that organizations must address to safely conduct business online. WhiteHat has been publishing the report, which highlights the top vulnerabilities, tracks vertical market trends and identifies new attack techniques, since 2006.

The WhiteHat Security report presents a statistical picture of current website vulnerabilities among the more than 2,000 websites under management, accompanied by WhiteHat expert analysis and recommendations. WhiteHat's report is the only one in the industry to focus solely on previously unknown vulnerabilities in custom Web applications, code unique to an organization, within real-world websites.

"Over the last year we determined that the average website had nearly 13 serious vulnerabilities with a standard deviation (**σ**) of 29.11, meaning that most websites had between 0 and 42."



There should be more public shaming for companies that take 3+ months to fix an XSS v

10:15 AM Jul 15th, 201

Retweeted by 1 person

chri

Chris E

RT Avg time in my data set, 67 days. @chriseng: There should be more public shaming for companies that take 3+ months t

10:17 AM Jul 15th, 201

jere

Jeremi

@jeremiahg any idea of the standard deviation

10:30 AM Jul 15th, 201

chri

Chris E

@chriseng fyi, the report will include standard deviation metrics just for you. :)

6:19 PM Aug 30th, 2010 via TweetDeck in reply to chriseng

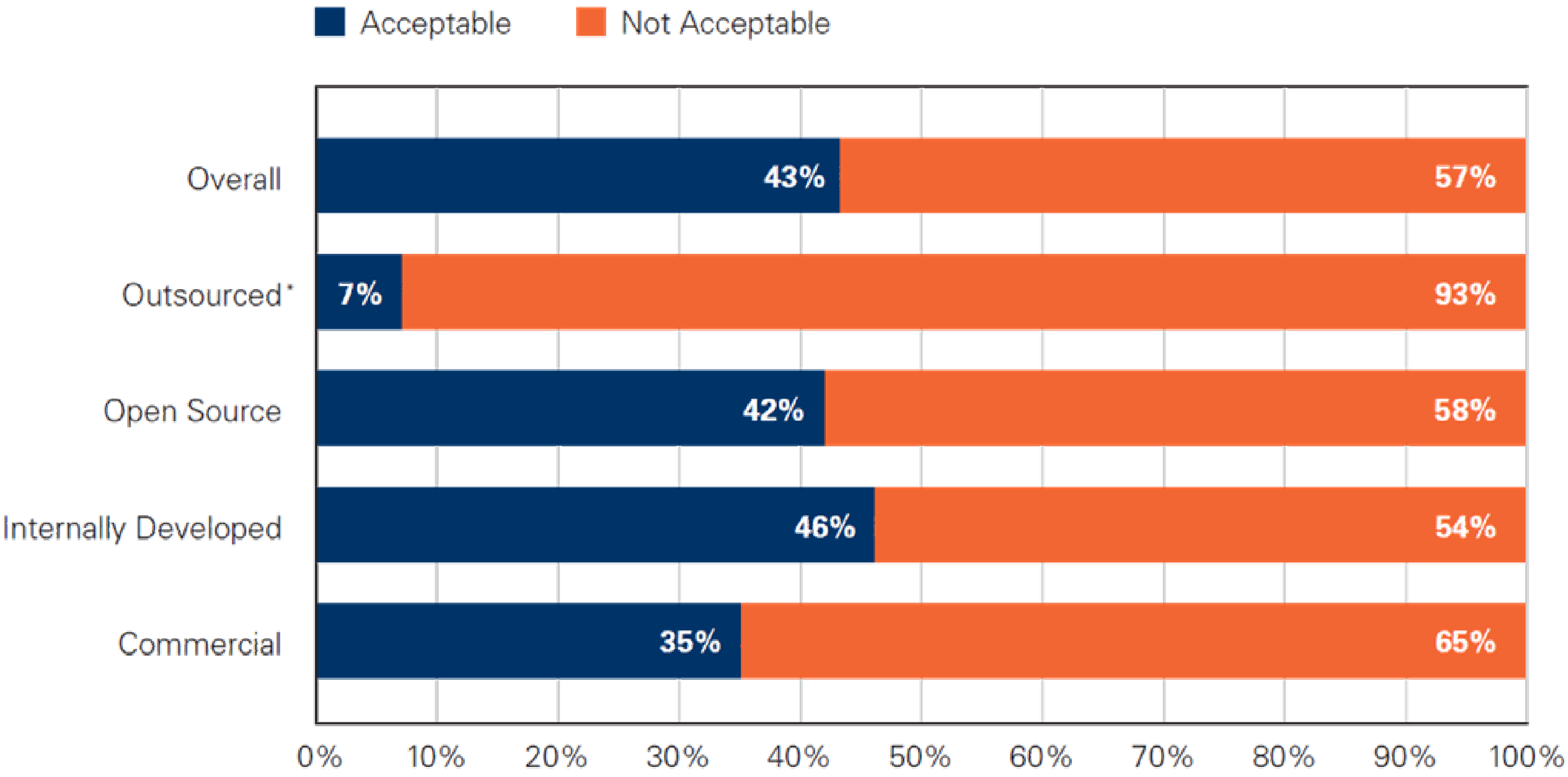
jeremiahg

Jeremiah Grossman

SAMPLE SIZE

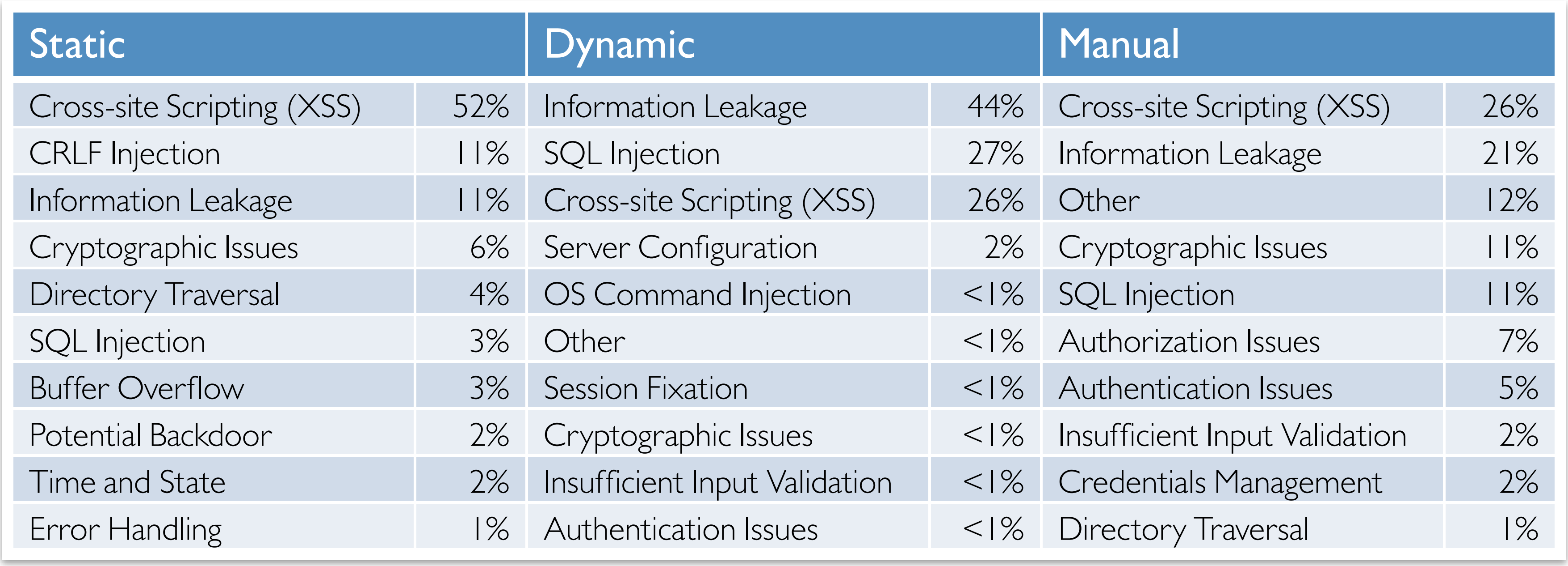
Power analysis can be used to determine the ***"statistically significant"*** sample size required to ensure the probability of error is acceptably low for a particular hypothesis.

To Asterisk or Not To Asterisk



Storytelling via Omission

*or, trying to figure out what’s missing and why



Static		Dynamic		Manual	
Cross-site Scripting (XSS)	52%	Information Leakage	44%	Cross-site Scripting (XSS)	26%
CRLF Injection	11%	SQL Injection	27%	Information Leakage	21%
Information Leakage	11%	Cross-site Scripting (XSS)	26%	Other	12%
Cryptographic Issues	6%	Server Configuration	2%	Cryptographic Issues	11%
Directory Traversal	4%	OS Command Injection	<1%	SQL Injection	11%
SQL Injection	3%	Other	<1%	Authorization Issues	7%
Buffer Overflow	3%	Session Fixation	<1%	Authentication Issues	5%
Potential Backdoor	2%	Cryptographic Issues	<1%	Insufficient Input Validation	2%
Time and State	2%	Insufficient Input Validation	<1%	Credentials Management	2%
Error Handling	1%	Authentication Issues	<1%	Directory Traversal	1%

Top 10 Flaw Categories by Analysis Type

Static		Dynamic		Manual	
Cross-site Scripting (XSS)	52%	Information Leakage	44%	Cross-site Scripting (XSS)	26%
CRLF Injection	11%	SQL Injection	27%	Information Leakage	21%
Information Leakage	11%	Cross-site Scripting (XSS)	26%	Other	12%
Cryptographic Issues	6%	Server Configuration	2%	Cryptographic Issues	11%
Directory Traversal	4%	OS Command Injection	<1%	SQL Injection	11%
SQL Injection	3%	Other	<1%	Authorization Issues	7%
Buffer Overflow	3%	Session Fixation	<1%	Authentication Issues	5%
Potential Backdoor	2%	Cryptographic Issues	<1%	Insufficient Input Validation	2%
Time and State	2%	Insufficient Input Validation	<1%	Credentials Management	2%
Error Handling	1%	Authentication Issues	<1%	Directory Traversal	1%
Numeric Errors	1%			Session Fixation	1%
Untrusted Search Path	1%			Time and State	1%
Credentials Management	1%			CRLF Injection	<1%
Encapsulation	1%			Server Configuration	<1%
API Abuse	1%			Deployment Configuration	<1%
Buffer Management Errors	<1%			Numeric Errors	<1%
Insufficient Input Validation	<1%			Potential Backdoor	<1%
OS Command Injection	<1%				
Race Conditions	<1%				
Dangerous Functions	<1%				
…6 more categories…	<1%				

Top 10 Vulnerabilities by Analysis Type

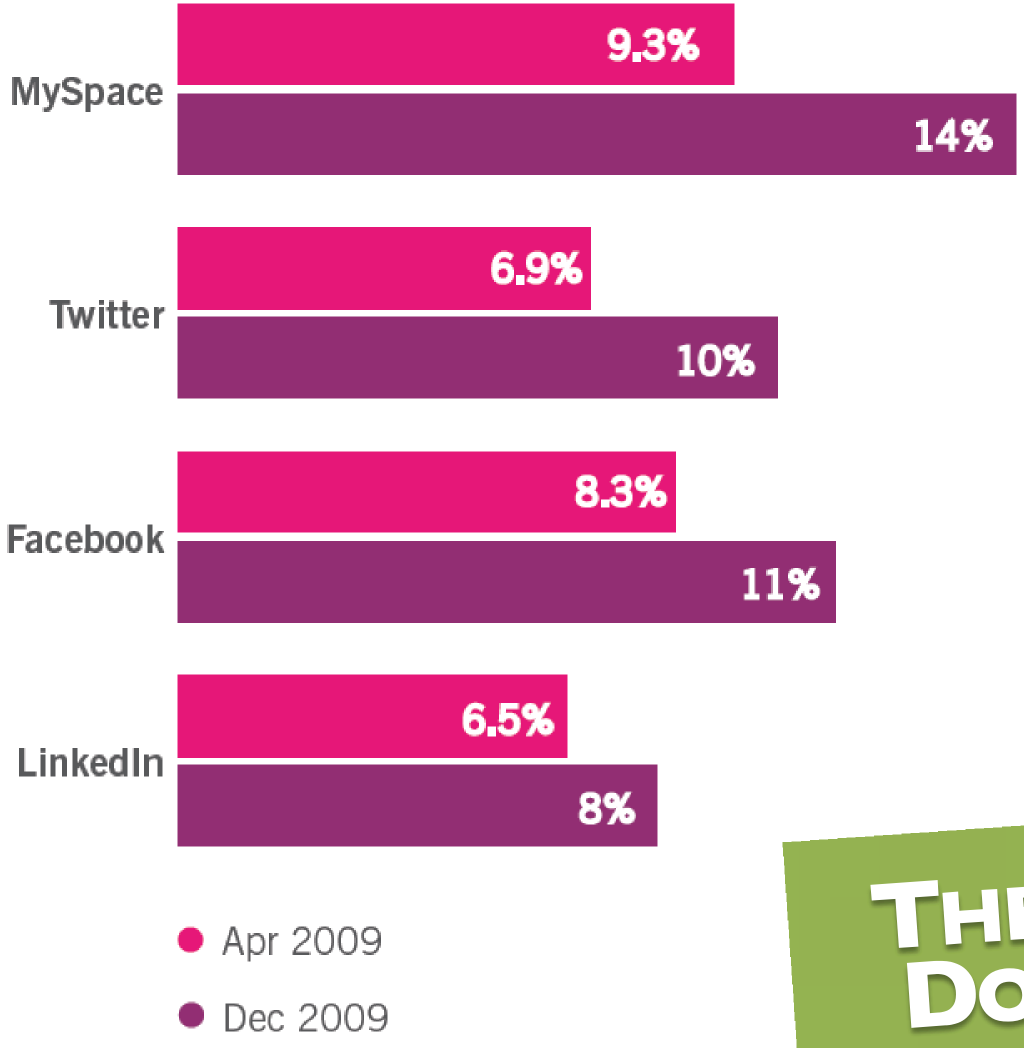
How Does This Alter Your Interpretation?

Flaw Category	Static	Dynamic
Cross-Site Scripting (XSS)	308.39	13.22
CRLF Injection	206.85	0
Cryptographic Issues	43.44	0.05
Information Leakage	43.38	5.71
SQL Injection	17.63	10.32
Directory Traversal	12.66	0
Potential Backdoor	10.82	0
Time and State	4.49	0
Encapsulation	3.95	0
Credentials Management	3.57	0
Insufficient Input Validation	3.18	0.01
API Abuse	1.73	0
Error Handling	1.42	0
Buffer Overflow	0.52	0
OS Command Injection	0.41	0.05
Numeric Errors	0.19	0
Untrusted Search Path	0.16	0
Dangerous Functions	0.15	0
Race Conditions	0.14	0
Session Fixation	0.09	[illegible]
Authentication Issues	0.08	0.03
Buffer Management Errors	0.03	0
Other	0	0.08

20% of Web Apps Scanned w/Both Static and Dynamic

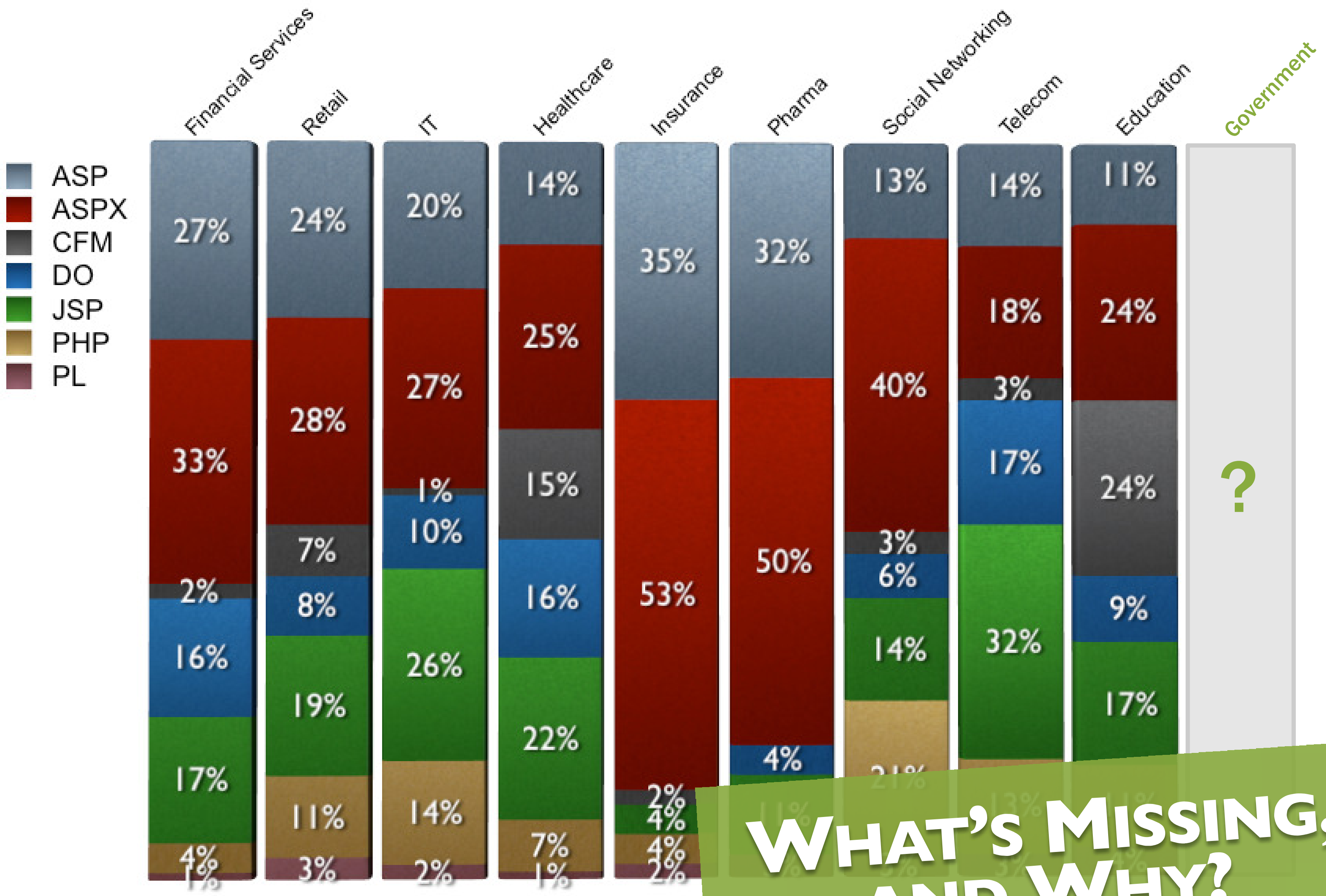
What Unwanted Assumptions Might Result?

Firms citing malware as their number one concern with social networks



THESE NUMBERS DON'T ADD UP

SOURCE: SOPHOS SECURITY THREAT REPORT 2010

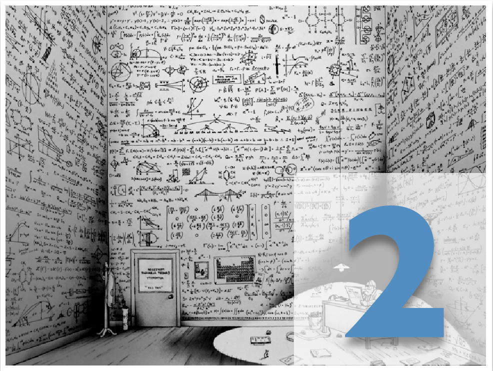


WHAT'S MISSING, AND WHY?

SOURCE: WHITEHAT WEBSITE SECURITY STATISTICS REPORT, 9TH EDITION, MAY 2010

Conclusions

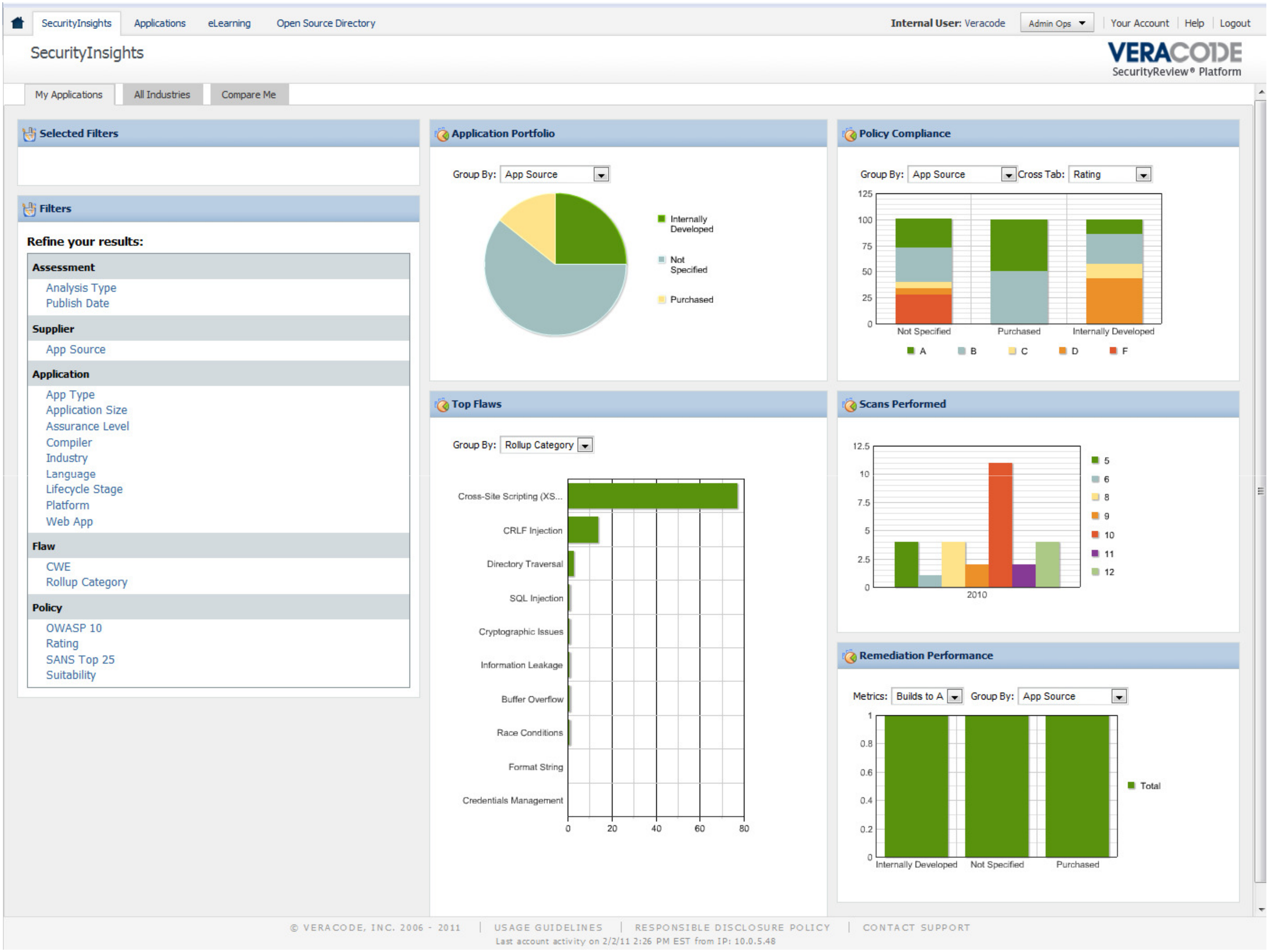
Identify the narrative(s)



Look for “responsible” use of statistics

Consider what’s not being shown

Encourage Data Sharing



Internal User: Veracode

SecurityInsights

Refine your results:

Assessment
- Analysis Type
- Publish Date

Supplier
- App Source

Application
- App Type
- Application Size
- Assurance Level
- Compiler
- Industry
- Language
- Lifecycle Stage
- Platform
- Web App

Flaw
- CWE
- Rollup Category

Policy
- OWASP 10
- Rating
- SANS Top 25
- Suitability

Application Portfolio — Group By: App Source

Policy Compliance — Group By: App Source Cross Tab: Rating

Top Flaws — Group By: Rollup Category

Remediation Performance — Metrics: Builds to A Group By: App Source

© VERACODE, INC. 2006 - 2011 | USAGE GUIDELINES | RESPONSIBLE DISCLOSURE POLICY | CONTACT SUPPORT

Last account activity on 2/2/11 2:26 PM EST from IP: 10.0.5.48

More Resources

- Sign up for FREE access to Veracode Analytics!
  - http://info.veracode.com/veracode-analytics
- Whitepapers, webcasts, and other resources
  - http://veracode.com/resources
- Veracode ZeroDay Labs Blog
  - http://veracode.com/blog
- Contact info
  - Email: ceng@veracode.com
  - Twitter: @chriseng

QUESTIONS?