

2,000 Websites Later

*Which Web Programming Languages
are Most Secure?*

Jeremiah Grossman
Founder & Chief Technology Officer

- WhiteHat Security Founder & Chief Technology Officer
- 2010 RSA Security Bloggers Award (Best Corporate Blog)
- InfoWorld's CTO Top 25 (2007)
- 5th most popular “Jeremiah” according to Google
- Brazilian Jiu-Jitsu Brown Belt
- Narcissistic Vulnerability Pimp
- Former Yahoo! information security officer



me.

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 AWARDS
2009
FINALIST
Honored in the U.S.
MAGAZINE

**USA
TODAY**

CIO

eWEEK.com

Forbes
com

San Francisco Chronicle

darkREADING
RISKY BUSINESS

PCWORLD

BANK TECHNOLOGY NEWS
btn

abcNEWS

InfoWorld

COMPUTERWORLD

internetnews.com®

CSO
The Resource for Security Executives

NETWORKWORLD

AP Associated Press

InformationWeek

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

Attacker Targeting



Fully Targeted (APT?)

- Customize their own tools
- Focused on business logic
- Profit or goal driven (\$\$\$)

WhiteHat Sentinel
PREMIUM EDITION

Directed Opportunistic

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

WhiteHat Sentinel
STANDARD EDITION

Random Opportunistic

- Fully automated scripts
- Unauthenticated scans
- Targets chosen indiscriminately

WhiteHat Sentinel
BASELINE EDITION

Evolution of Expectations

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, without spending a zillion hours doing it?

Vulnerability Overlap

What's a website?

Websites, which may be a collection of multiple web servers and hostnames, often utilize more than one programming language or framework. As such, a single website may contain vulnerabilities with multiple different extensions.



Data Overview

- **1,659 total websites**
- **24,286 verified custom web application vulnerabilities**
- Data collected from January 1, 2006 to March 25, 2010
- Vast majority of websites assessed for vulnerabilities weekly
- Vulnerabilities classified according to WASC Threat Classification, the most comprehensive listing of Web application vulnerabilities
- Vulnerability severity naming convention aligns with PCI-DSS
- Contrasted and compared ASP Classic, .NET, Cold Fusion, Struts, Java Server Pages, PHP, and Perl.

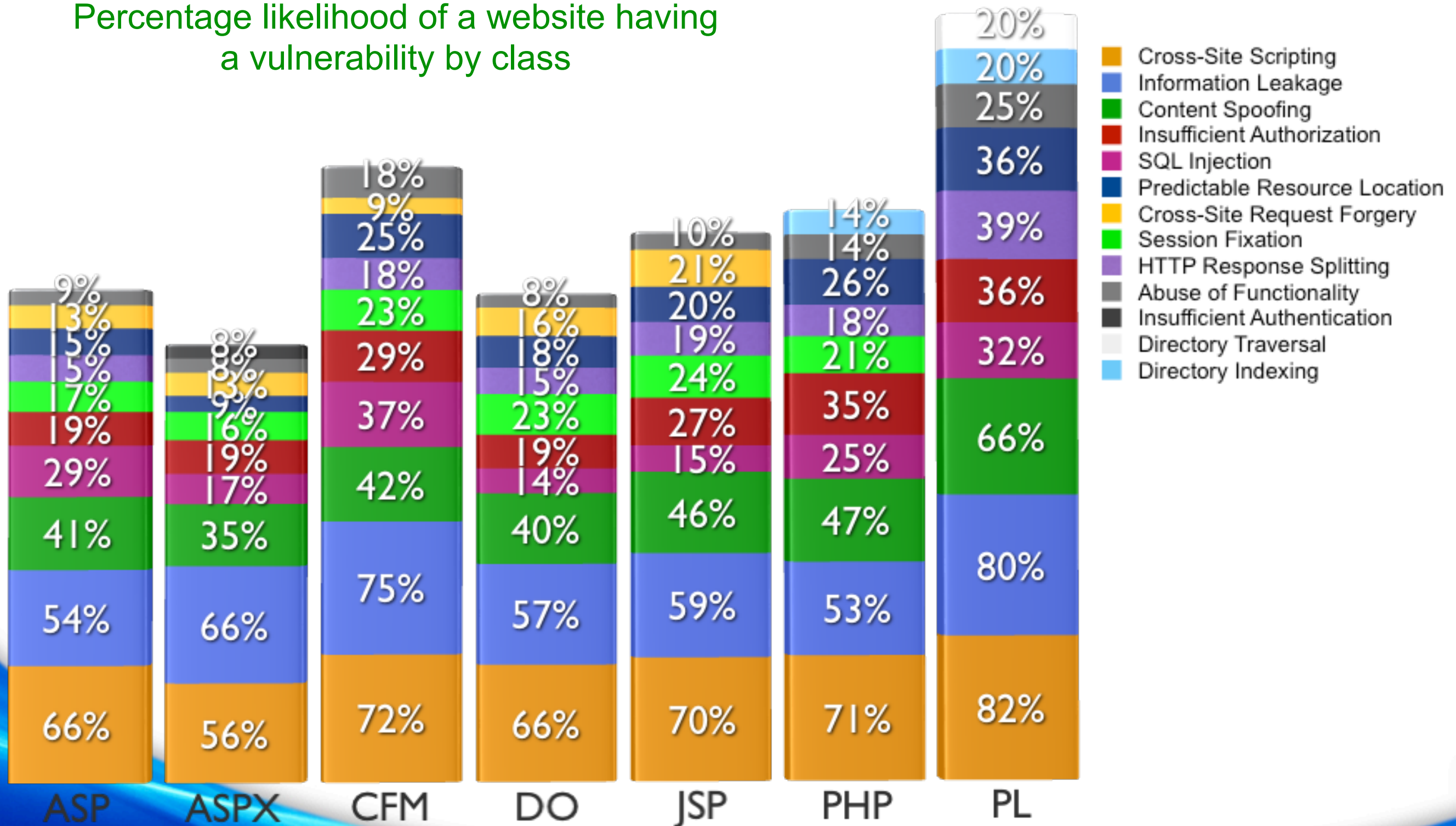
| | ASP | ASPX | CFM | DO | JSP | PHP | PL |
|---|------|------|------|------|------|------|-------|
| Average # of inputs (attack surface) per website | 470 | 484 | 457 | 569 | 919 | 352 | 588 |
| Average ratio of vulnerability count / number of inputs | 8.7% | 6.2% | 8.4% | 6.3% | 9.8% | 8.1% | 11.6% |

Key Findings

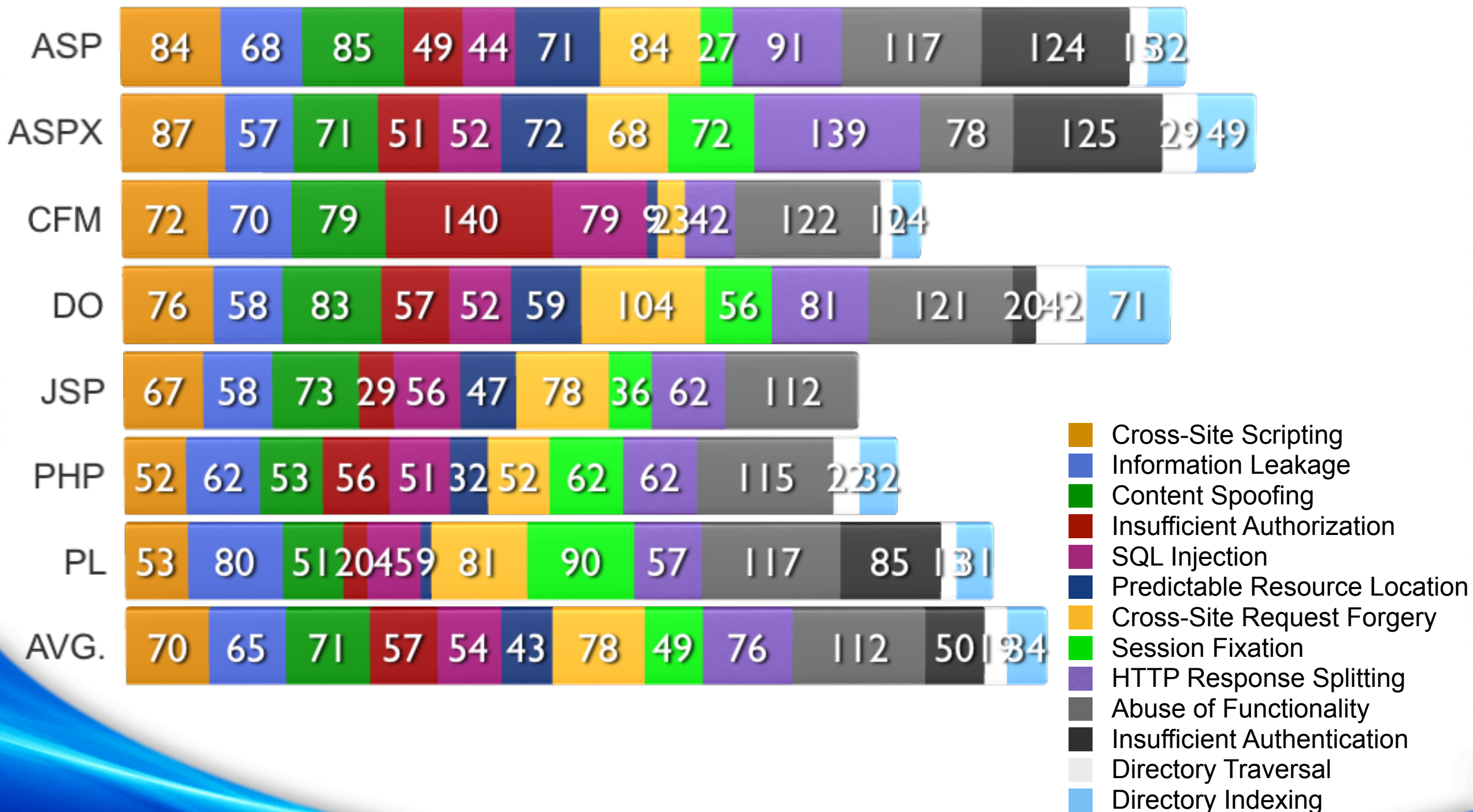
| | ASP | ASPX | CFM | DO | JSP | PHP | PL |
|---|-----|------|------|------|------|------|------|
| Websites <u>having had</u> at least one serious* vulnerability | 74% | 73% | 86% | 77% | 80% | 80% | 88% |
| Websites <u>currently with</u> at least one serious* vulnerability | 57% | 58% | 54% | 56% | 59% | 63% | 75% |
| Avg. # of serious* vulnerabilities per website during the WhiteHat Sentinel assessment lifetime | 25 | 18.7 | 34.3 | 19.9 | 25.8 | 26.6 | 44.8 |
| Avg. # of serious* severity unresolved vulnerabilities per website | 8.9 | 6.2 | 8.6 | 5.5 | 9.6 | 8.3 | 11.8 |

Top Ten Classes of Attack

Percentage likelihood of a website having a vulnerability by class



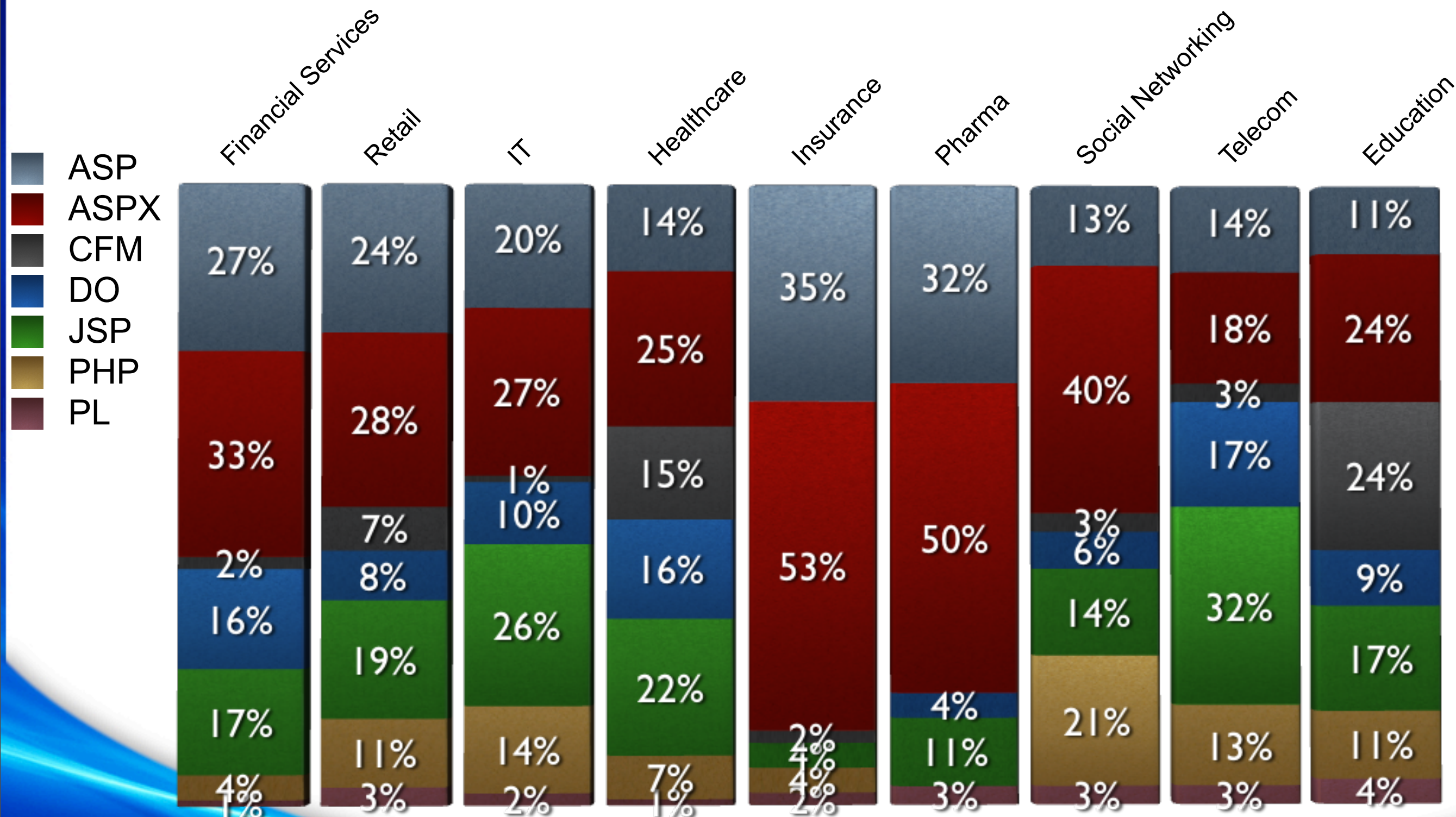
Time-to-Fix (Days)



Resolution Rates by Severity

| Class of Attack | Severity | ASP | ASPX | CFM | DO | JSP | PHP | PL |
|-----------------------------|----------|------|------|------|-----|-----|-----|------|
| SQL Injection | Urgent | 70% | 72% | 66% | 79% | 58% | 70% | 71% |
| Insufficient Authorization | Urgent | 21% | 45% | 46% | 20% | 25% | 18% | 10% |
| Directory Traversal | Urgent | 43% | 20% | 67% | 0% | 33% | 32% | 16% |
| Cross Site Scripting | Urgent | 100% | 0% | 100% | 0% | 0% | 50% | 0% |
| Cross-Site Scripting | Critical | 51% | 57% | 50% | 51% | 52% | 66% | 54% |
| Cross-Site Request Forgery | Critical | 18% | 34% | 17% | 27% | 39% | 57% | 27% |
| Session Fixation | Critical | 19% | 18% | 0% | 36% | 50% | 50% | 100% |
| Abuse of Functionality | Critical | 76% | 23% | 82% | 38% | 57% | 59% | 97% |
| Insufficient Authentication | Critical | 55% | 37% | 0% | 33% | 71% | 0% | 100% |
| Information Leakage | High | 32% | 34% | 57% | 49% | 45% | 39% | 29% |
| Content Spoofing | High | 31% | 30% | 43% | 37% | 44% | 46% | 69% |
| Predictable Resource Loc. | High | 29% | 64% | 85% | 64% | 53% | 56% | 29% |
| HTTP Response Splitting | High | 28% | 24% | 33% | 10% | 36% | 42% | 35% |
| Directory Indexing | High | 33% | 56% | 40% | 25% | 27% | 33% | 18% |
| TOTAL | | 65% | 67% | 75% | 72% | 63% | 69% | 74% |

Technology in Use



Lessons & Observations

You can't secure what you don't know you own – Inventory Web applications to gain visibility into what data is at risk and where attackers can exploit the money or data transacted.

Assign a champion – Designate someone who can own and drive data security and is strongly empowered to direct numerous teams for support. Without accountability, security, and compliance, will suffer.

Don't wait for developers to take charge of security – Deploy shielding technologies to mitigate the risk of vulnerable Web applications.

Questions?

I was not in your threat model.

1:53 PM Apr 28th via TweetDeck

Retweeted by 1 person



jeremiahg
Jeremiah Grossman

Jeremiah Grossman

Founder & Chief Technology Officer

Blog: <http://jeremiahgrossman.blogspot.com/>

Twitter: <http://twitter.com/jeremiahg>

Email: jeremiah@whitehatsec.com