Web Application Security Metrics

Metricon 2.0 (Boston)
08.07.2007

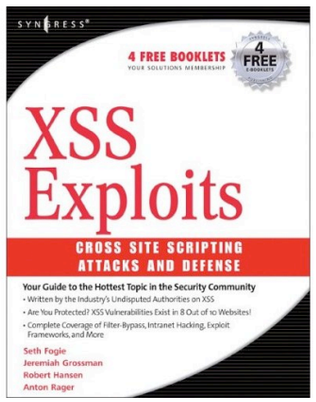
Jeremiah Grossman (Founder and CTO)

Jeremiah Grossman

- Founder and CTO of WhiteHat Security
- R&D and industry evangelism
- International conference speaker
- Co-Author of XSS Attacks
- Web Application Security Consortium Co-founder
- Former Yahoo! information security officer

WASC Member

2007 InfoWorld CTO 25



YAHOO!®

WhiteHat SECURITY

Target #1: Layer 7

128 million websites

Many are mission-critical and gateways to highly sensitive customer and corporate information

These websites are accessible by over 1 billion people

Everyone is a Target

Hacked

Consequences of an insecure Website

- Loss of business
- Damage to customer confidence and brand
- Regulatory fines
- Legal liability
- Financial costs of handling an incident

How a hacker can break-in: The Data

All data collected through vulnerability assessments performed by WhiteHat Security between January 2006 and August 2007

Includes hundreds of largest and most popular websites among the retail, financial services, IT, Pharma, Insurance, Education, social networking, and Healthcare verticals

Remote and external black-box assessment methodology - typically conducted weekly

WASC Threat Classification used as a baseline

We focus solely on custom web application vulnerabilities - no well-known issues

Collection process

WhiteHat Sentinel Service

Unlimited Assessments – customer controlled and expert managed - the ability to scan websites no matter how big or how often they change

Coverage – authenticated scans to identify technical vulnerabilities and custom testing to uncover business logical flaws

Virtually Eliminate False Positives – Operations Team verifies results and assigns the appropriate severity and threat rating

Development and QA – WhiteHat Satellite Appliance allows us to service intranet accessible systems remotely

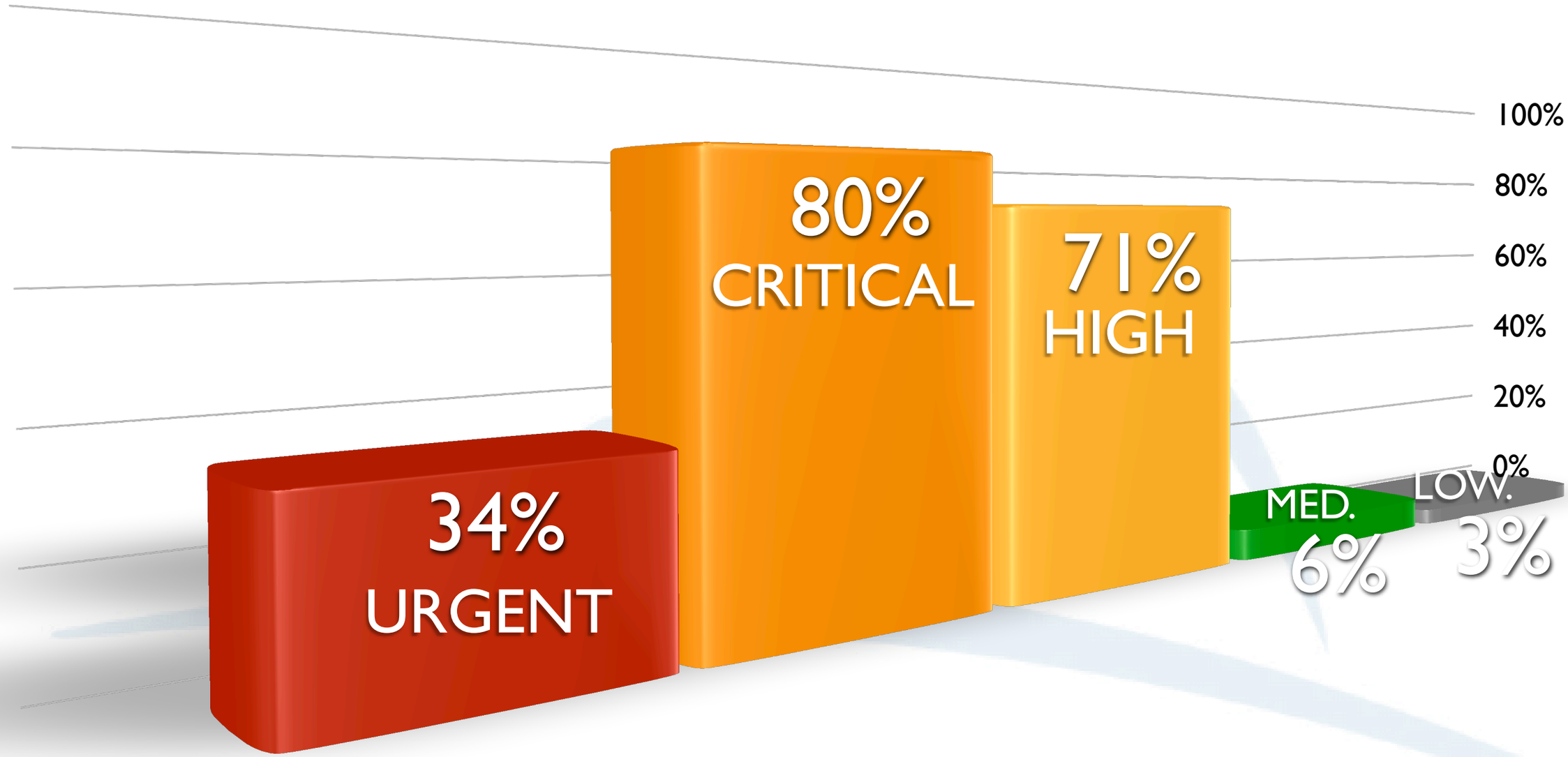
Improvement & Refinement – real-world scans enable fast and efficient updates

7 out of 10 Websites have serious Vulnerabilities

Not all websites have the same overall business value. Some websites are mission critical, while others are static "brochureware." Our dataset represents the most "important" and "secure" websites, conducting high-volume transactions or managing sensitive information.

But how bad is it really?

Likelihood that a website has a vulnerability, by severity

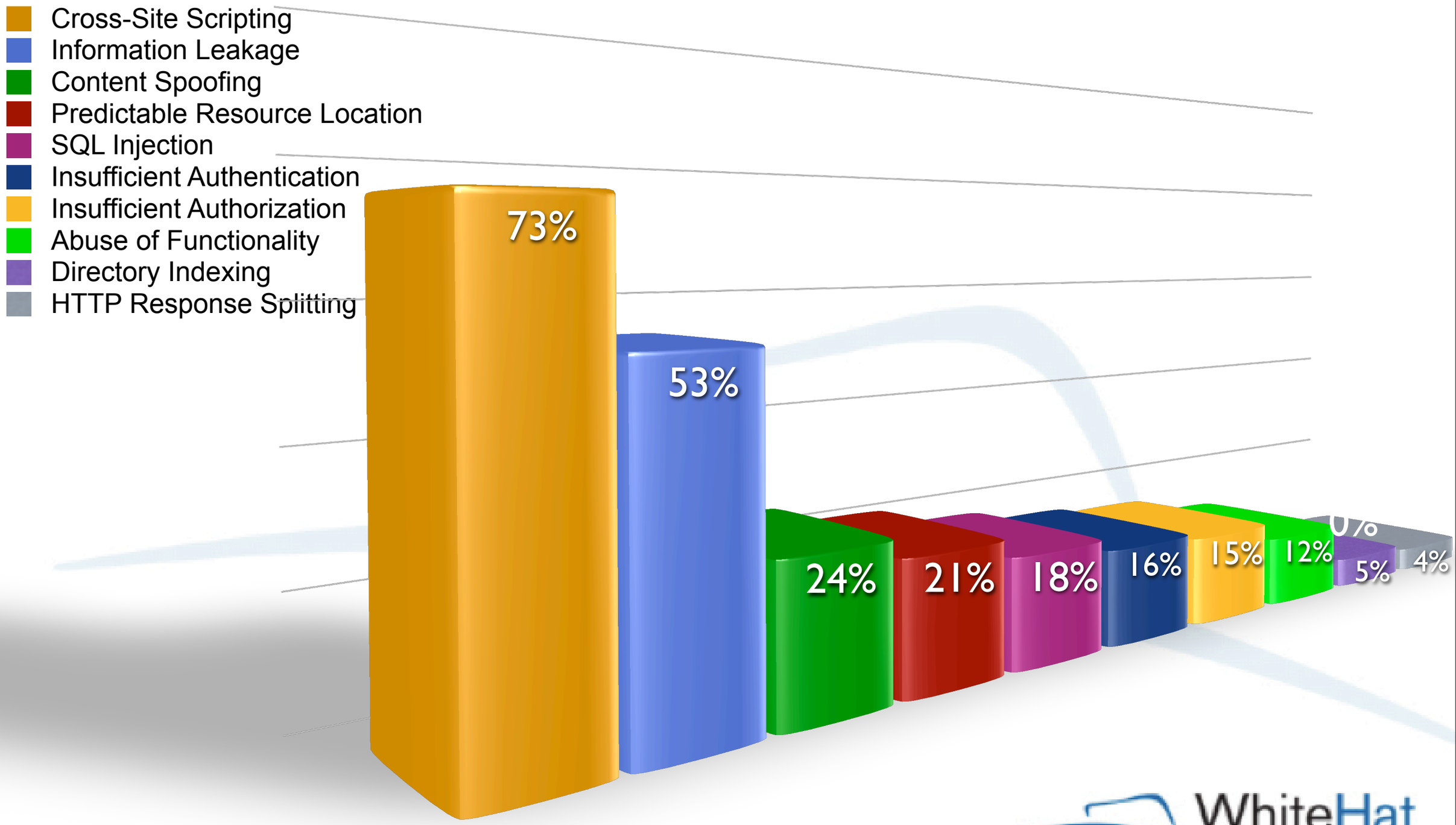


Websites with Urgent, Critical, or High severity issues technically would not pass PCI compliance

WhiteHat SECURITY

What's there: Top 10

Likelihood that a website has a vulnerability, by Class



WhiteHat SECURITY

What's not there

Obviously we're not going to find buffer overflows or format string issues in custom web applications

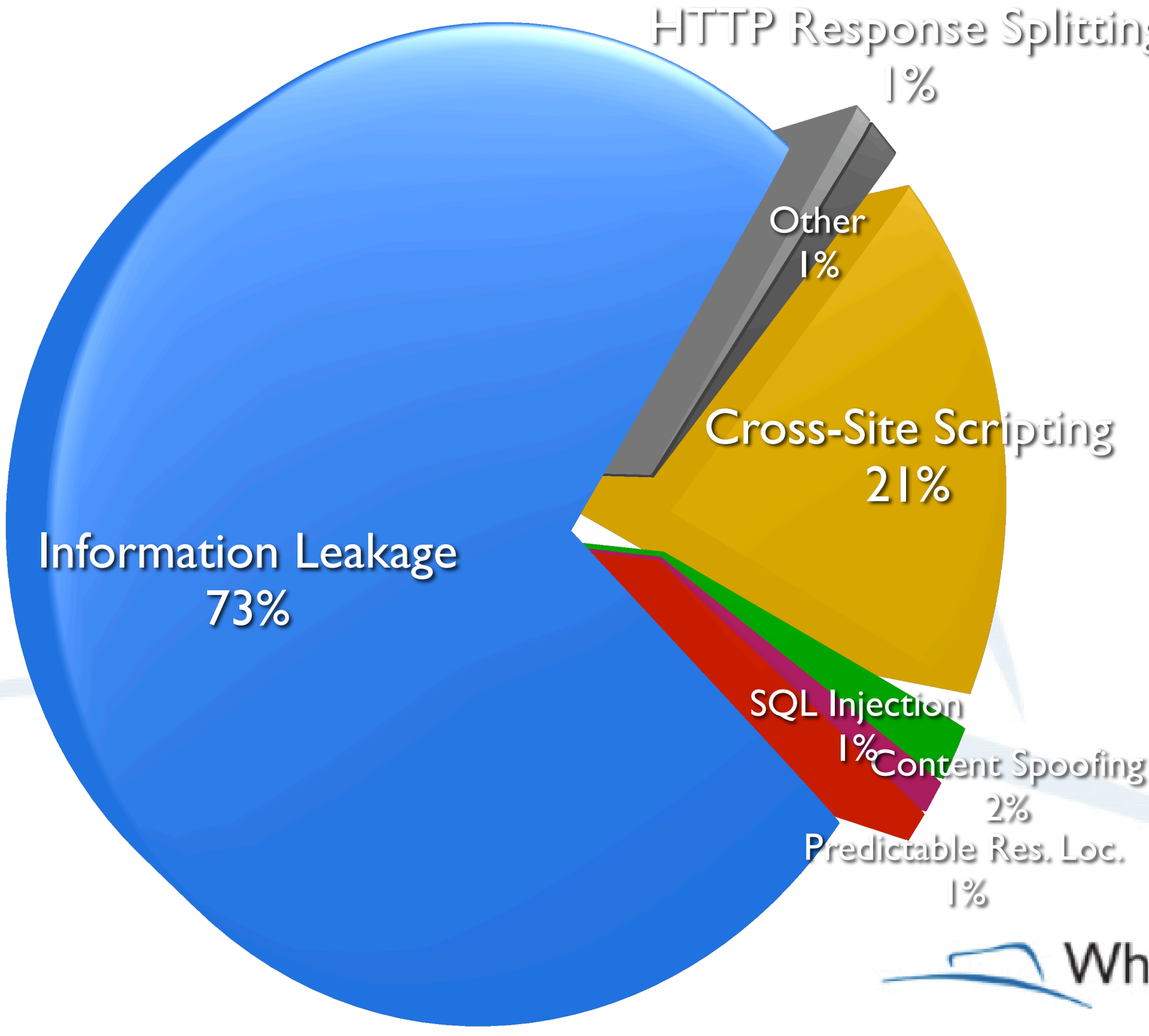
We're also not looking for the well-known PHP issues and the like

Cross-Site Request Forgery remains **very difficult** to scan for and we only report the most egregious cases identified by hand

We keep finding new and cool ways of performing XSS filter-evasions

HTTP Response Splitting pushed XPath Injection off the list

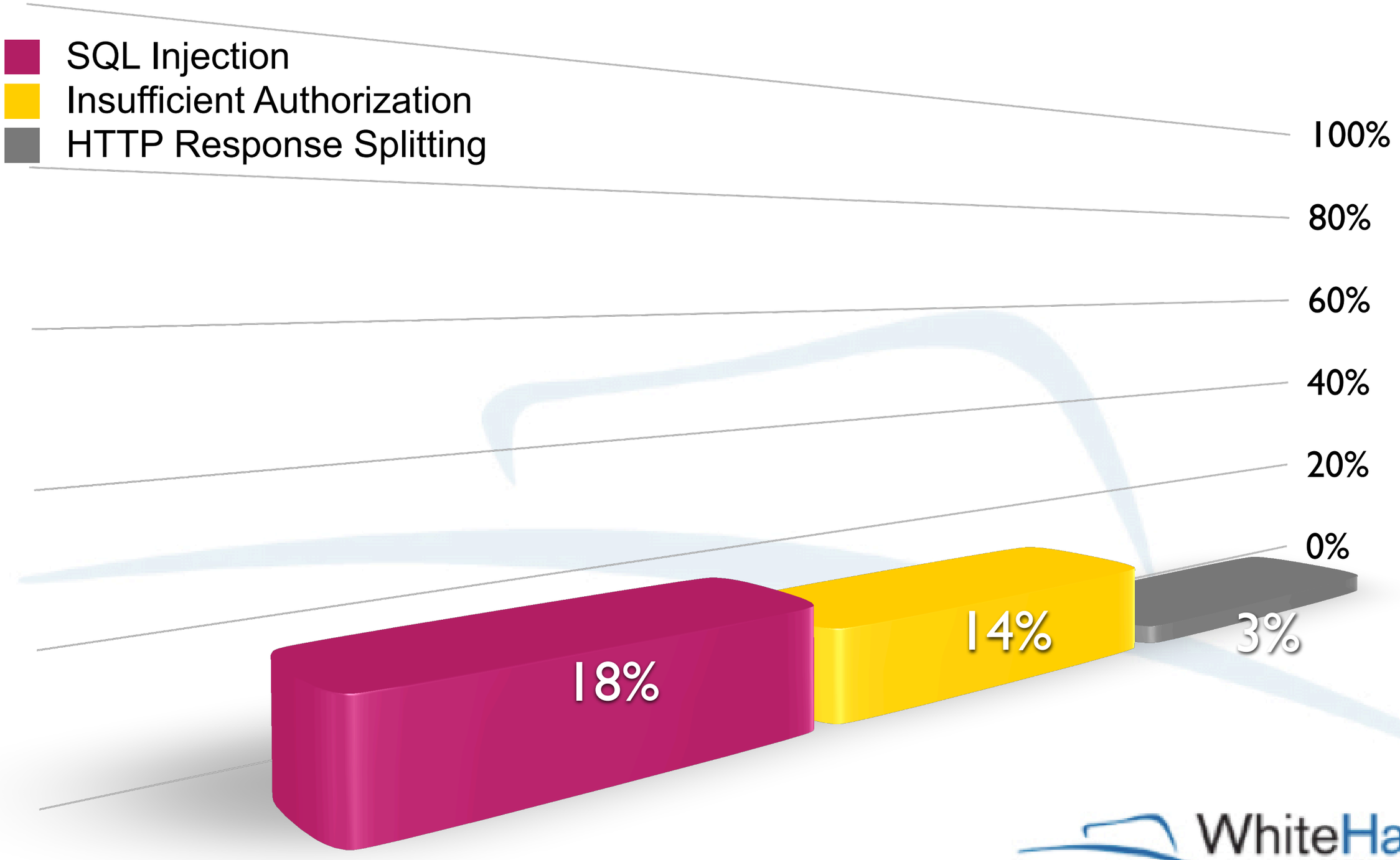
Overall vulnerability population



WhiteHat SECURITY

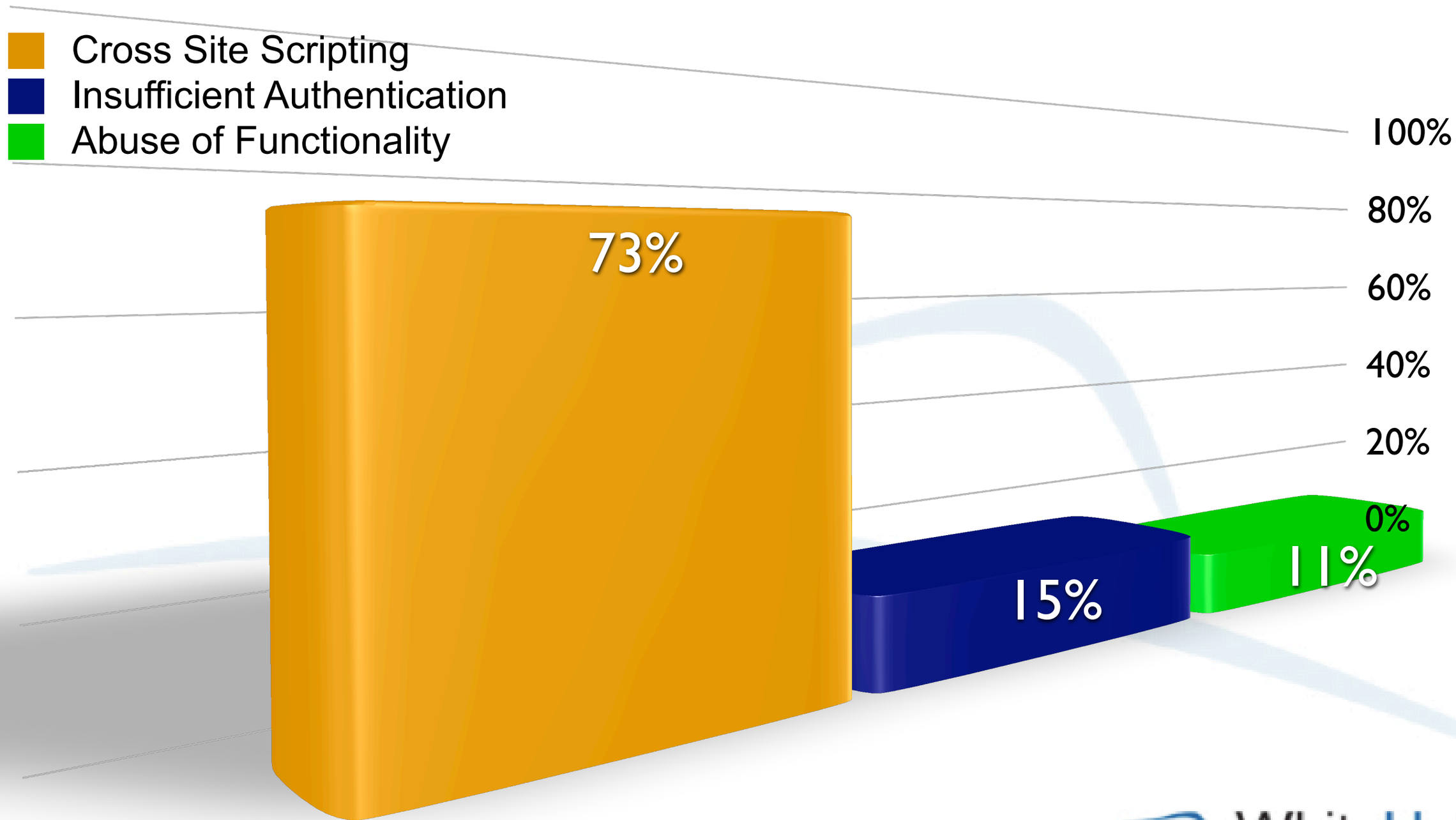
Urgent

Likelihood that a website has an "Urgent Severity" vulnerability, by class



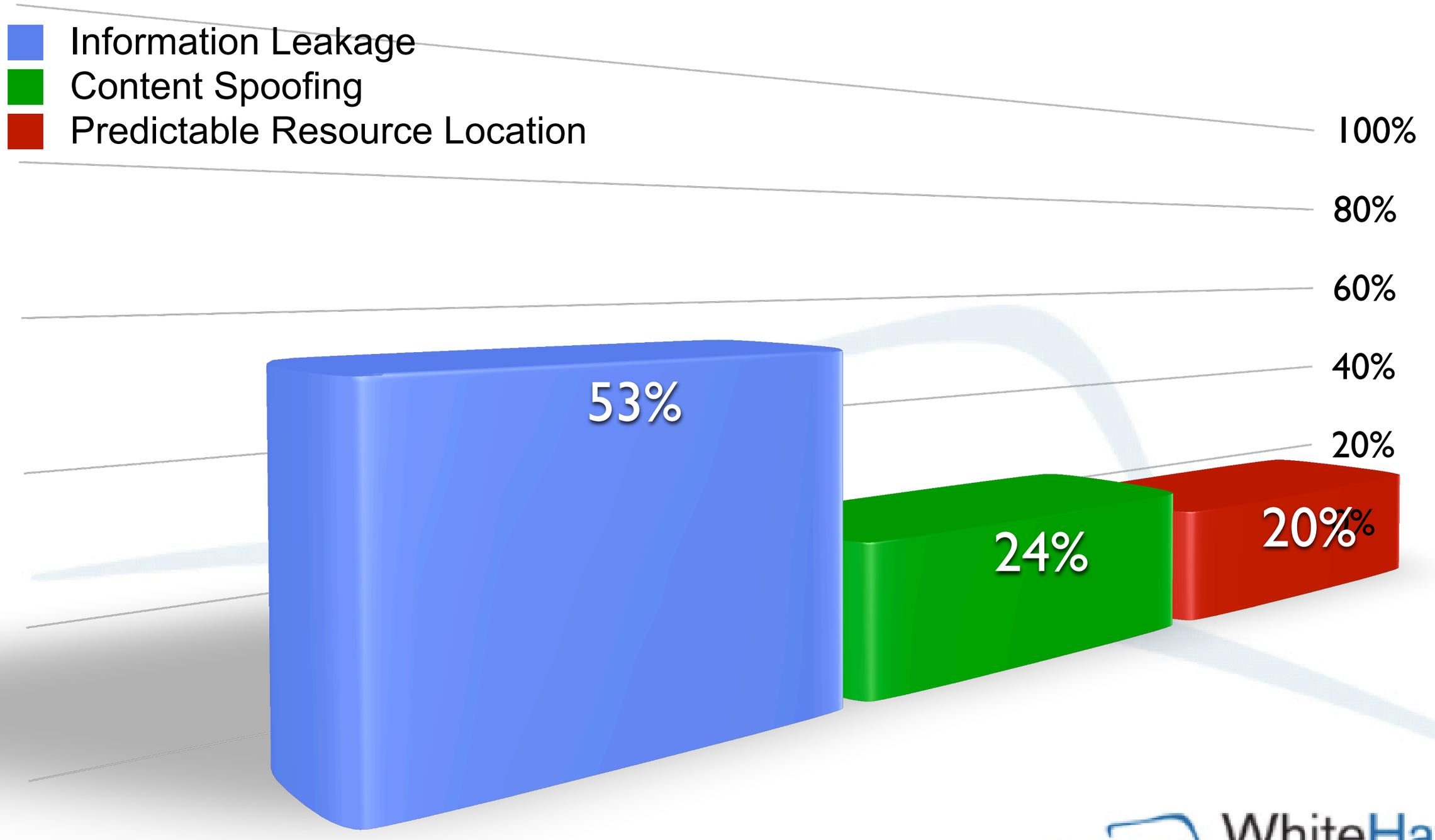
Critical

Likelihood that a website has a "Critical Severity" vulnerability, by class



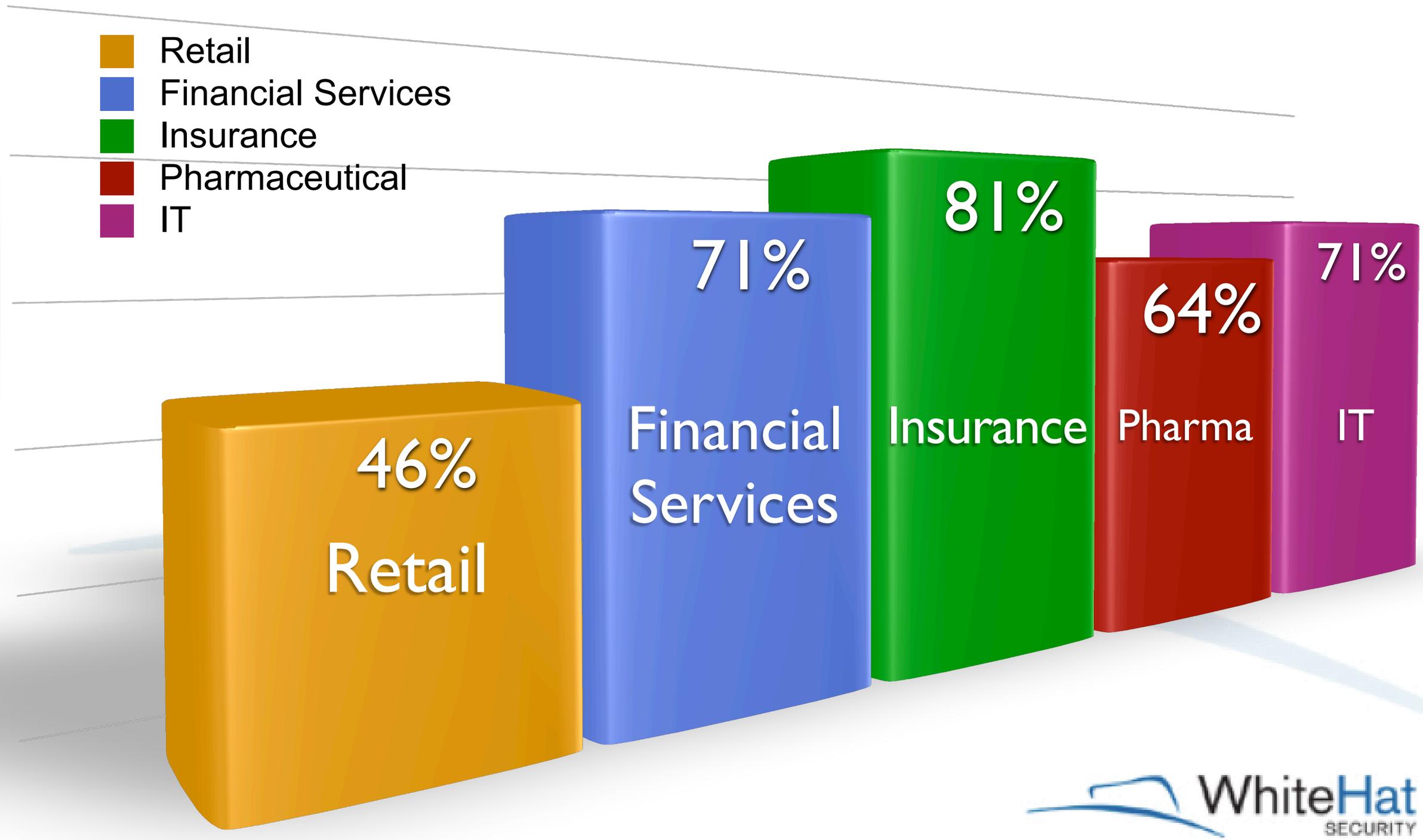
High

Likelihood that a website has a "High Severity" vulnerability, by class



Comparing industry verticals

Likelihood that a website in a particular has a vulnerability (at least 25 websites)



Top 3 by industry vertical

Retail
1) Cross Site Scripting
2) Information Leakage
3) Predictable Resource Location

Financial Services
1) Cross Site Scripting
2) Information Leakage
3) SQL Injection

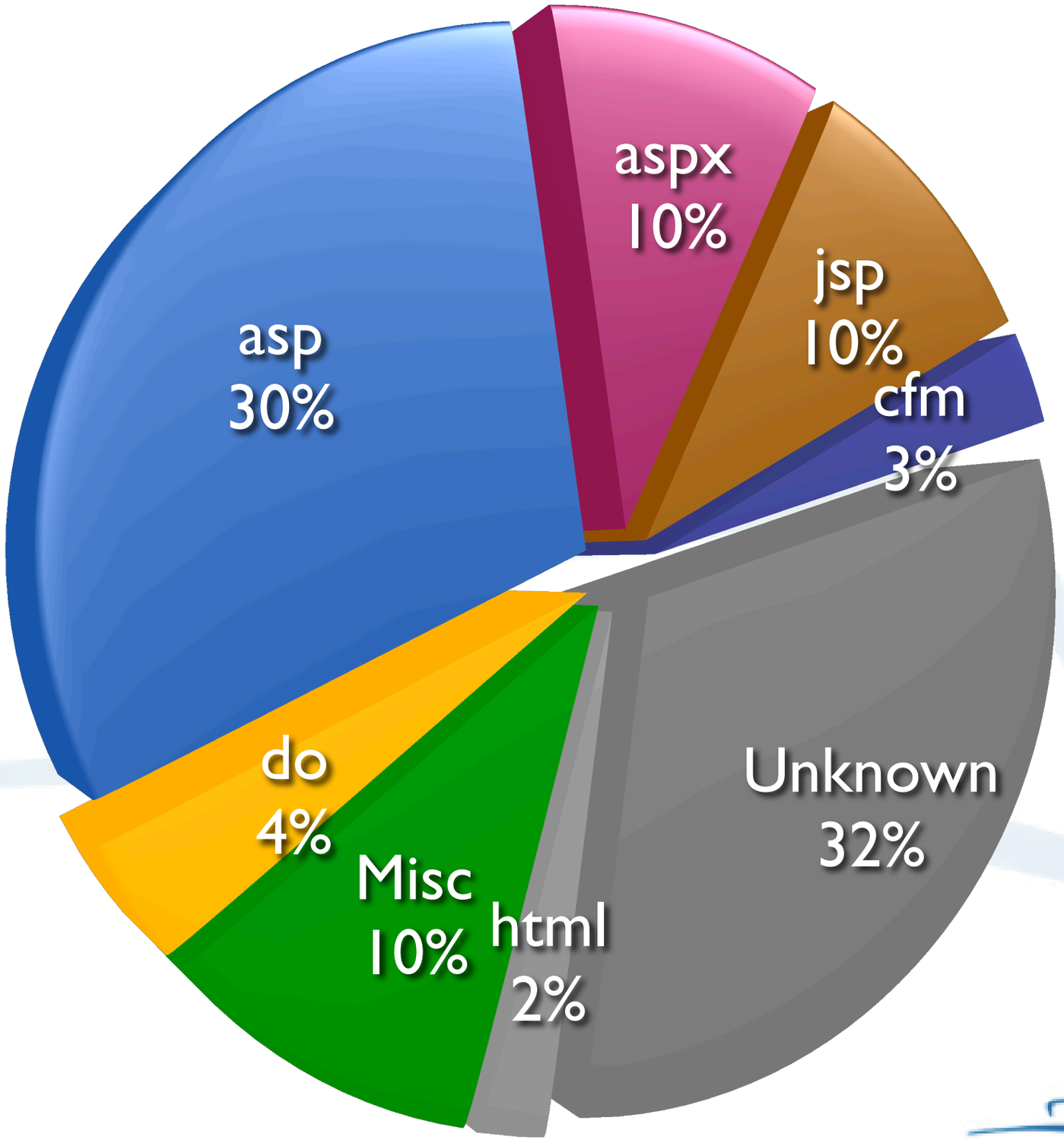
Insurance
1) Information Leakage
2) Insufficient Authentication
3) Cross Site Scripting

Pharmaceutical
1) Cross Site Scripting
2) Information Leakage
3) Content Spoofing

IT
1) Cross Site Scripting
2) Information Leakage
3) Insufficient Authentication

First pass at platform technology

Looking at the file extensions



Those that are more "secure" have:

- Use of modern development frameworks with security configs turned on (.NET, J2EE, Rails, etc.)
- At least some security involvement in the SDLC (awareness training, threat modeling, QA testing, etc.)
- Vulnerability remediation prioritized by severity/threat rating (High: 1 - 7 days, Medium: < 30 days, Low: Next Update)

WhiteHat SECURITY

Best Practices

Asset Tracking – Find your websites, assign a responsible party, and rate their importance to the business. Because you can't secure what you don't know you own.

Measure Security – Perform rigorous and on-going vulnerability assessments, preferably every week. Because you can't secure what you can't measure.

Development Frameworks – Provide programmers with software development tools enabling them to write code rapidly that also happens to be secure. Because, you can't mandate secure code, only help it.

Defense-in-Depth – Throw up as many roadblocks to attackers as possible. This includes custom error messages, Web application firewalls, security with obscurity, and so on. Because 8 in 10 websites are already insecure, no need to make it any easier.

WhiteHat SECURITY

Future Plans

- Flesh out vertical and technology comparisons
- Trend vulnerability increase/decrease over time and **re-open rate**
- Attack surface ratios of inputs to vulnerabilities
- Hackability!

Thank you

For more information visit:

http://www.whitehatsec.com/

Jeremiah Grossman (Founder and CTO)

jeremiah@whitehatsec.com