# WhiteHat Website Security Statistic Report

***Fall 2009, 8th Edition***

**8th edition**

## Introduction

The fact that the vast majority of websites, including those considered most business critical, are riddled with vulnerabilities is common knowledge to regular readers of this report. Essentially, every other industry report available unanimously agrees Web applications represent the #1 avenue of attack. Unfortunately, what is not well-known is exactly what are the most efficient steps to measurably improve the security posture of an existing website, or one soon to be built. Ironically, there is no shortage of security best-practice recommendations, despite a dearth of metrics to justify the investment. So, enterprises are left to guess, and hope their actions actually decrease the likelihood and impact of an incident.

WhiteHat Security would like to continue its long track record of bringing meaningful metrics to the fore and shedding new light on "what works." We believe the data gathered by WhiteHat Security contains valuable lessons from those that are "more secure" than rest. In this report we have introduced a new section, Zero-Vulnerability, which is a first-look at various websites which do not currently or have never had serious issues. The goal of this new section is to begin exploring what differences they may have, if any, from those sites which do – have vulnerabilities. What can they teach us about the best-practices they use and how outcomes are affected? Does implementing certain controls equally affect all vulnerabilities in the same way, on the same timeline, or are the results less consistent?

We can make no claim to answer all these questions immediately in this edition of the report, but there are some very interesting observations already. From this point forward, we will continue the process of peeling back the layers so we can ask better questions, and field questions from our readership, and questions from our customers. We are confident that over time new ways of understanding, prioritizing, and addressing Web application security issues will be made readily apparent.

## Data Collection Process

Built on a Software-as-a-Service (SaaS) – technology platform, WhiteHat Sentinel combines advanced proprietary scanning technology with expert website security analysis, to enable customers to identify, prioritize, manage and remediate vulnerabilities as they occur. WhiteHat Sentinel focuses solely on previously unknown vulnerabilities in custom Web applications-- code unique to an organization, on real-world websites (see Figure 1 next page ). Unique to WhiteHat Security, every vulnerability discovered by any WhiteHat Sentinel Service is verified and prioritized, virtually eliminating false-positives and radically simplifying remediation.

*Web security is a moving target and enterprises need timely information about the latest attack trends, how they can best defend their websites, and 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 ten vulnerabilities, tracks vertical market trends and identifies new attack techniques, since 2006.*

*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.*

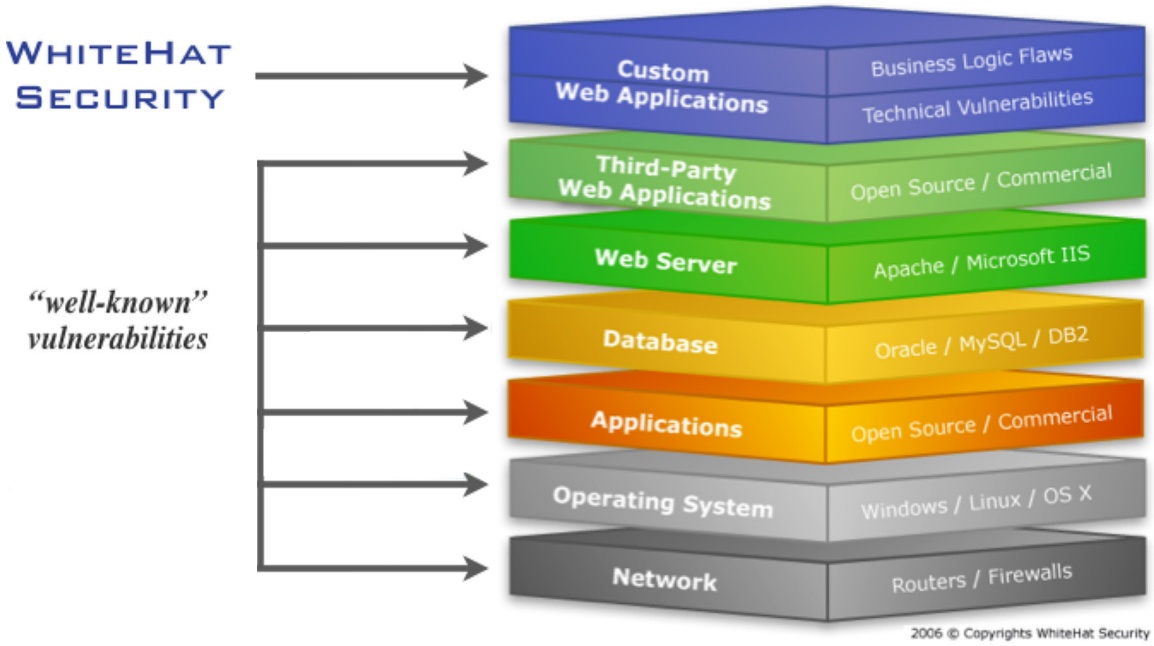


**Figure 1.**

WhiteHat Sentinel was built to scale massively with the capability to assess hundreds, even thousands of the largest and most complex websites simultaneously. The technology was also built specifically to run in both QA/development and production environments to ensure maximum coverage with no performance impact. The websites covered by WhiteHat Sentinel likely representing the most "important" and "secure" sites on the Web, owned by enterprises that are serious about their security.

WhiteHat Sentinel offers three different levels of service (Premium, Standard, and Baseline) to match the level of security assurance required by the organization. http://www.whitehatsec.com/home/services/selection.html And, WhiteHat Sentinel exceeds PCI 6.6 and 11.3.2 requirements for Web application scanning.

## Most Advanced Scanning Technology Available

- "Production safe" scanning process - Non-invasive testing methodology with less performance impact than a single user
- Years of battlefield testing - Proven track record of identifying more vulnerabilities than any commercial scanner
- Unparalleled accuracy - False-positives are virtually eliminated by the WhiteHat Security Operations Team
- Seamless support for Web 2.0 technology - modern websites using JavaScript, Macromedia Flash, AJAX, Java Applets, or ActiveX
- Authenticated scans - Patented automated login technology for complete website mapping
- Thorough coverage - custom tests analyze every Web form, business process, and authentication/authorization component

## Data Overview

- Data collected from **January 1, 2006 to October 1, 2009**
- **1,364 (32% ↑) total websites**
- **22,776 (4,888 ↑) verified** (custom web application) vulnerabilities
- Vast majority of websites assessed for vulnerabilities weekly
- Vulnerabilities classified according to WASC Threat Classification, the most comprehensive listing of Web application vulnerabilities (see Figure 2 on the following page)

- Vulnerability severity naming convention aligns with PCI-DSS
- Average # of links spidered per website: 766*
- Average # of inputs (attack surface) per website: 246
- Average ratio of vulnerability count / number of inputs: 2.14%
- Websites with responding with at least one X-FRAME-OPTIONS[1] (anti-clickjacking) header: 1
- Websites with responding with at least one httpOnly[2] (anti-XSS cookie stealing) header: 150

* *WhiteHat Sentinel seeks to identify all of a websites externally available attack surface, which may or may not require spidering all of its available links.*

| Technical Vulnerabilities | Business Logic Flaws |
|---|---|
| **Command Execution**<br>– Buffer Overflow<br>– Format String Attack<br>– LDAP Injection<br>– OS Commanding<br>– SQL Injection<br>– SSI Injection<br>– XPath Injection<br><br>**Information Disclosure**<br>– Directory Indexing<br>– Information Leakage<br>– Path Traversal<br>– Predictable Resource Location<br><br>**Client-Side**<br>– Content Spoofing<br>– Cross-site Scripting (XSS)<br>– HTTP Response Splitting | **Authentication**<br>– Brute Force<br>– Insufficient Authentication<br>– Weak Password Recovery<br>– Validation<br>– Cross-Site Request Forgery<br><br>**Authorization**<br>– Credential/Session Prediction<br>– Insufficient Authorization<br>– Insufficient Session Expiration<br>– Session Fixation<br><br>**Logical Attacks**<br>– Abuse of Functionality<br>– Denial of Service<br>– Insufficient Anti-automation<br>– Insufficient Process Validation |

**Figure 2. WASC Threat Classification**

## Key Findings

- **83% of websites have had at least one serious* vulnerability**
- **64% of websites currently have at least one serious* vulnerability**
- 61% vulnerability resolution-rate with 8,902 unresolved issues remaining
- Average # of serious* vulnerabilities per website during the WhiteHat Sentinel assessment lifetime: 16.7
- Average # of serious* severity unresolved vulnerabilities per website: 6.5
- The vulnerability characteristics of websites currently without any serious* issues were nearly identical to those that did, with the exception that they had about half as many to begin with.
- Vulnerability time-to-fix metrics are beginning to fluctuate, both lengthening or shortening depending on class, yet still require weeks to months to resolve.
- Vulnerability resolution percentages are nudging higher across the range, particularly within the Cross-Site Scripting and SQL Injection classes.
- Social Networking and Education vertical websites most likely to have serious* severity issues (86% and 83% respectively)
- Most previously established metrics, such as the WhiteHat Security Top Ten, have remained largely static indicating a representative data sampling.

* *Serious vulnerabilities are those of HIGH, CRITICAL, or URGENT severity as defined by PCI-DSS naming conventions. Exploitation could lead to significant and direct business impact.*

**When interpreting the results there are several factors that should be considered that influence the results:**

- Websites range from highly complex and interactive with large attack surfaces to static brochureware.
- Vulnerabilities are counted by unique Web application and class of attack. If there are five parameters in a single Web application (/foo/webapp.cgi), three of which are vulnerable to SQL Injection, it is counted as one vulnerability (not three).
- "Best practice" findings are not included in the report. For example, if a website mixes SSL content with non-SSL on the same Web page, while this may be considered a business policy violation, it must be taken on a case-by-case basis. Only issues that can be directly and remotely exploitable are included.
- Vulnerability assessment processes are incremental and ongoing, the frequency of which is customer-driven and as such should not automatically be considered "complete." The vast majority of WhiteHat Sentinel customers have their sites assessed weekly.
- New attack techniques are constantly being researched to uncover previously unknown vulnerabilities. This makes it best to view the data as a best-case scenario. Likewise, assessments may be conducted in different forms of authenticated states (i.e. user, admin, etc.).
- Websites may be covered by different WhiteHat Sentinel service levels (Premium (PE), Standard (SE), Baseline (BE)) offering varying degrees of testing comprehensiveness, but all include verification. PE covers all technical vulnerabilities and business logic flaws identified by the WASC 24 (and some beyond). SE focuses primarily on the technical vulnerabilities. BE bundles critical technical security checks into a production-safe, fully-automated service.

## Vulnerability Prevalence by Severity

In order for organizations to take appropriate action, each website vulnerability must be independently evaluated for business criticality. For example, not all Cross-Site Scripting or SQL Injection vulnerabilities are equal, making it necessary to consider its true "severity" for an individual organization. Using the Payment Card Industry Data Security Standard[3] (PCI-DSS) severity system (Urgent, Critical, High, Medium, Low) as a baseline, WhiteHat Security rates vulnerability severity by the potential business impact if the issue were to be exploited and does not rely solely on default scanner settings.


**Figure 3. Percentage likelihood of websites having a least one vulnerability (sorted by severity)**

## The Top Ten

The most prevalent issues are calculated by the percentage likelihood of a particular vulnerability class occurring within websites (Figure 4). This approach minimizes data skewing in website edge cases that are either highly secure or extremely risk-prone.

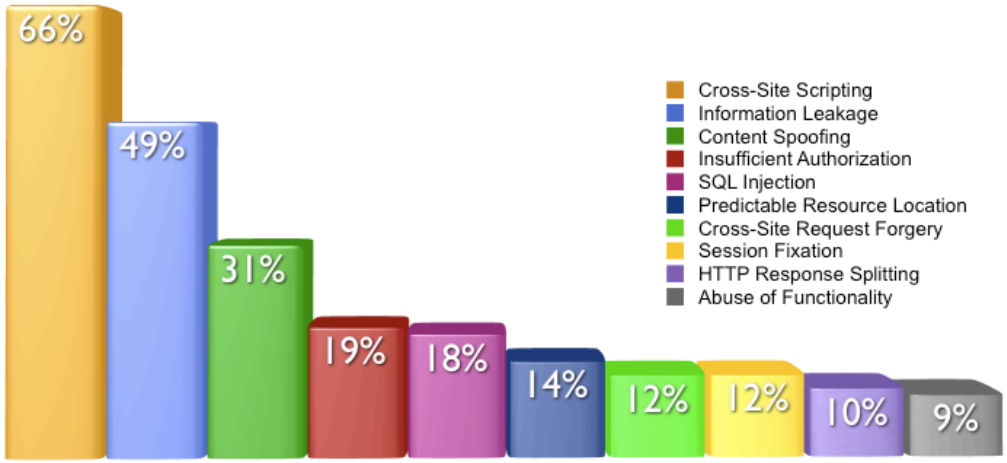


**Figure 4. Top 10 Vulnerability Classes (sorted by percentage likelihood)**

To supplement vulnerability likelihood statistics, the following graph (Figure 5) illustrates prevalence by class in the overall vulnerability population. Notice how greatly it differs from the Top Ten graph. The reason is that one website may possess hundreds of unique issues of a specific class, such as Cross-Site Scripting, Information Leakage, or Content Spoofing, while another website may not contain any.

It is our opinion that SQL Injection and Cross-Site Request Forgery are under-represented in the Top Ten. To protect against SQL Injection attacks, industry best-practices suggest verbose error messages should be disabled to increase the difficulty of its exploitation. This act also has the side effect of increasing the difficulty for scanning technology to identify open issues. Despite adherence to this practice, the vulnerability persists and can be exploited by worms leveraging Blind SQL Injection without the need to identify an issue first. "SQL Injection, eye of the storm" has additional detailed information. Cross-Site Request Forgery is under-represented because scanning technology industrywide is still extremely limited in its detection capability. Most serious issues are still found by hand as were the majority of CSRF vulnerabilities identified in this report.

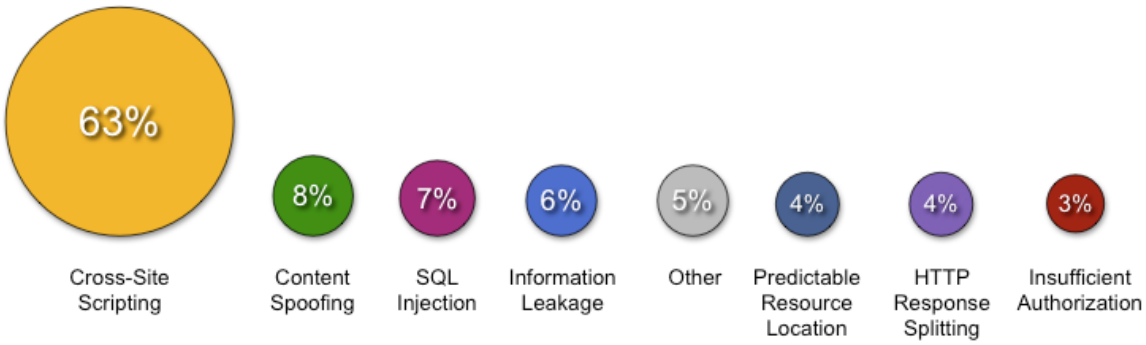


**Figure 5. Vulnerability Classes (sorted by overall class population)**

## Development Technology and Vulnerabilities

Table 1 provides insight into the types of technologies encountered during WhiteHat Sentinel vulnerability assessments and the associated vulnerability percentage breakdown. The statistics are not meant to establish which technology is more secure. For example, the under-representation of PHP likely means that this technology is not being utilized by those in the sample set relative to others. The large set of "unknown" are those without a file extension.

| URL Extension | % of websites | % of vulnerabilities |
|---|---|---|
| unknown | 62% | 39% |
| aspx | 23% | 9% |
| asp | 22% | 24% |
| xml | 11% | 2% |
| jsp | 10% | 8% |
| do | 6% | 3% |
| php | 6% | 3% |
| html | 5% | 2% |
| old | 3% | 1% |
| cfm | 3% | 4% |
| bak | 3% | 1% |

**Table 1.**

## Time-to-Fix

When website vulnerabilities are identified, there is a certain amount of time required for the issue to be resolved. Resolution could take the form of a software update, configuration change, Web application firewall rule, etc. Ideally the time to fix should be as short as possible because an open vulnerability represents an opportunity for hackers to exploit the website, but no remedy is instantaneous. To perform this analysis, we focused on vulnerabilities identified and resolved within the last twelve months between October 1, 2008 and October 1, 2009. The data was then sorted by the most common URGENT, CRITICAL, and HIGH severity issues.

**There are aspects worth noting that may bias the data:**

- *Should a vulnerability be resolved, it could take up to seven days before it is retested and confirmed closed by WhiteHat Sentinel, depending upon the customer's scan schedule. A customer can proactively use the auto-retest function to get real-time confirmation of a fix.*
- *Not all vulnerabilities identified within this period have been resolved, which means the time to fix measurements are likely to grow (See Table 2).*

Once vulnerabilities are identified it does not necessarily mean they are fixed quickly, or ever. It is interesting to analyze the types and severity of the vulnerabilities that do get fixed (or not) and in what volumes. Some organizations target the easier issues first to demonstrate their progress by vulnerability reduction. Others prioritize the high severity issues to reduce overall risk. Still, resources and security interest are not infinite so some issues will remain unresolved for extended periods of time. The reasons for this are diverse, but may include:

- *No one at the organization understands or is responsible for maintaining the code.*
- *Feature enhancements are prioritized ahead of security fixes.*
- *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 it*
- *No one at the organization knows about, understands, or respects the issue.*
- *Lack of budget to fix the issues*

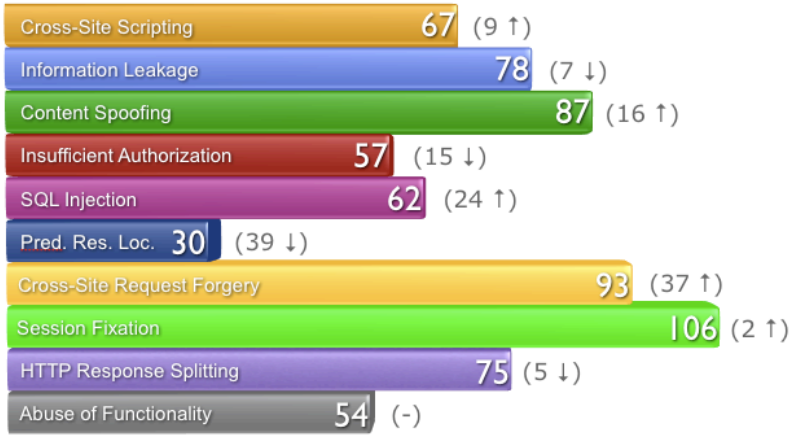


**Figure 6. Average number of days for vulnerabilities to be resolved**
** Up/down arrows indicate the increase or decrease since the last report.*

The time-to-fix metrics are still somewhat volatile. As we stated before, we expect the numbers to lengthen and become more representative as the percentage of resolved issues increases. What we can say with confidence is that IT Security and development organizations must coordinate when it comes to dealing with website vulnerabilities to close the time-to-fix gap.

| Class of Attack | % resolved | Δ | severity |
|---|---|---|---|
| Cross Site Scripting | 12% | 8 ↓ | urgent |
| Insufficient Authorization | 18% | 1 ↓ | urgent |
| SQL Injection | 40% | 10 ↑ | urgent |
| HTTP Response Splitting | 12% | 15 ↓ | urgent |
| Directory Traversal | 65% | 12 ↑ | urgent |
| Insufficient Authentication | 37% | 1 ↓ | critical |
| Cross-Site Scripting | 44% | 5 ↑ | critical |
| Abuse of Functionality | 14% | 14 ↓ | critical |
| Cross-Site Request Forgery | 39% | 6 ↓ | critical |
| Session Fixation | 31% | 10 ↑ | critical |
| Brute Force | 31% | 20 ↑ | high |
| Content Spoofing | 46% | 21 ↑ | high |
| HTTP Response Splitting | 32% | 2 ↑ | high |
| Information Leakage | 30% | 21 ↑ | high |
| Predictable Resource Location | 34% | 8 ↑ | high |

**Table 2. Percentage of vulnerabilities resolved (sorted by class & severity)**

## Comparing Industry Verticals

Figure 7 shows the percentage of websites with at least one Urgent, Critical, or High severity issue sorted by industry vertical. The majority of websites have these types of issues, which would likely preclude them from being classified as PCI-DSS compliant. Clearly no vertical is performing exceptionally well, but some are holding steady and achieving better results than others. The question is, why?

It is difficult to prove causation, but we have some correlation ideas. Battlefield testing, which occurs on those sites where significant functionality is ahead of the login screen (i.e. Retail). Meaning, the attackers are able to test their targets deeper and more often, which forces security improvement. Other swings could also be attributed to the addition of new websites into the sample set from that particular vertical, which have never undergone professional vulnerability assessment testing.

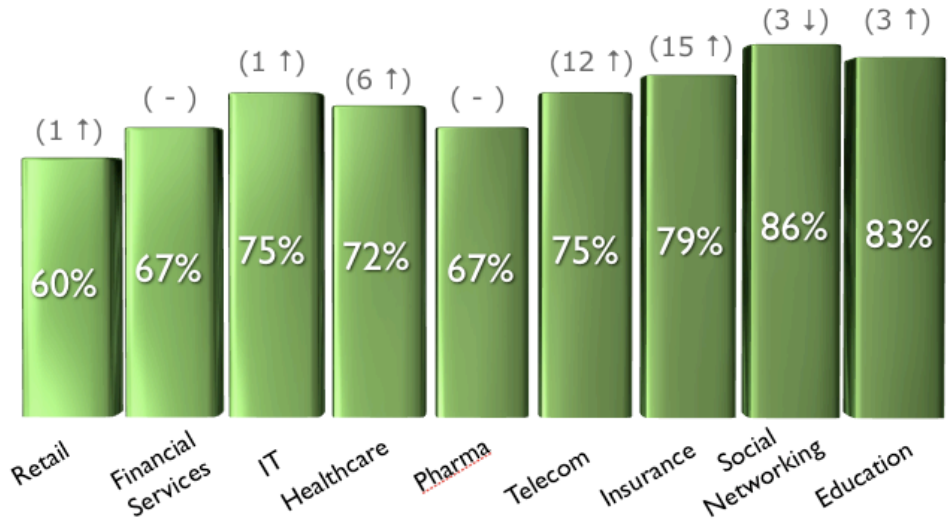


**Figure 7. Percentage of websites with an URGENT, CRITICAL or HIGH severity vulnerability sorted by industry vertical**

** Up/down arrows indicate the percentage increase or decrease since the last report.*

## INTRODUCING A NEW MATRIX SECTION – Zero-Vulnerability

- 485 total websites
- 17% of websites have never had a serious* vulnerability
- 36% of websites currently do not have an vulnerability
- 1,800 verified vulnerabilities
- Average # of serious* severity vulnerabilities per website during the WhiteHat Sentinel assessment lifetime: 3.7
- Average # of inputs (attack surface) per website: 244
- Average ratio of vulnerability count / number of inputs: 2.11%

* *Serious vulnerabilities are those of HIGH, CRITICAL, or URGENT severity as defined by PCI-DSS naming conventions. Exploitation could lead to significant and direct business impact.*

It should be noted that there may be a number of factors present which potentially artificially inflate the number of zero-vulnerability websites in the sample set. In future reports, we will try to isolate and measure these factors accordingly. These factors would include, but are not limited to:

- Brochureware websites, or those with very little functionality and attack surface.
- Websites where the bulk of the functionality cannot be exercised without proper authentication credentials which have not be supplied.
- WhiteHat Sentinel testing coverage differences between Premium Edition, Standard Edition, and Baseline Edition.
- WhiteHat Sentinel is being blocked at an IP address level by mitigating devices such as Web Application Firewall and Intrusion Prevention Systems.
- Customer WhiteHat Sentinel injection tests are being specifically blacklisted by application input-filters not properly resolving the issue.

**Zero-Vulnerability – Top 10 vulnerability classes (sorted by percentage likelihood)**

- Cross-Site Scripting (37.3%)
- Information Leakage (22.2%)
- Content Spoofing (10.7%)
- Predictable Resource Location (7.8%)
- SQL Injection (7.4%)
- Abuse of Functionality (4.3%)
- Insufficient Authorization (4.1%)
- Session Fixation (4.1%)
- Cross Site Request Forgery (3.7%)
- HTTP Response Splitting (3.1%)

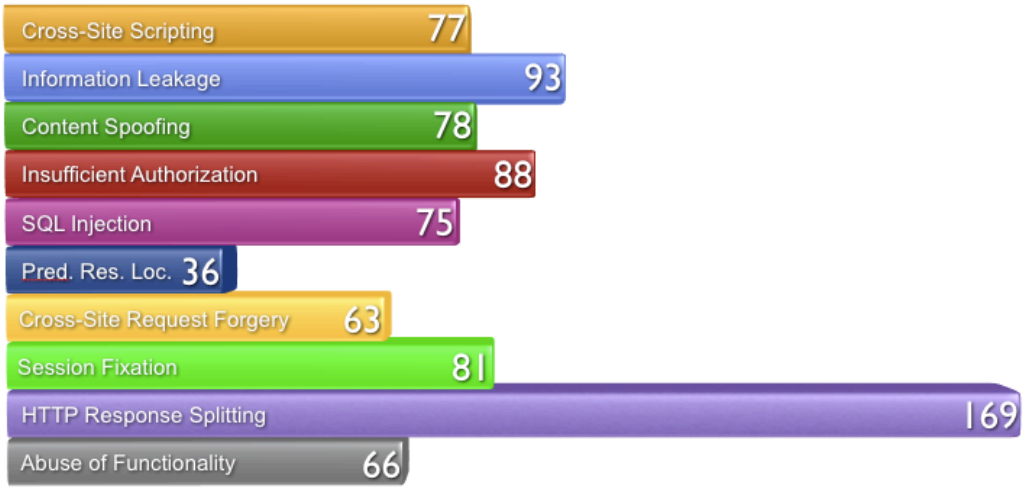


**Figure 8. Zero-Vulnerability – Average number of days for vulnerabilities to be resolved**

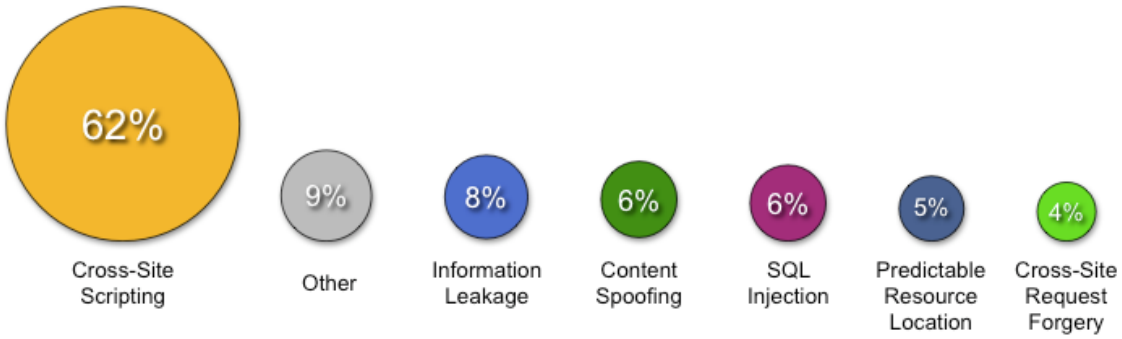


**Figure 9. Zero-Vulnerability Vulnerability Classes (sorted by overall class population)**

| URL Extension | % of websites | % of vulnerabilities |
|---|---|---|
| **unknown** | 33% | 33% |
| **aspx** | 7% | 10% |
| **asp** | 14% | 25% |
| **jsp** | 7% | 9% |
| **do** | 7% | 8% |
| **html** | 2% | 2% |
| **old** | 2% | 2% |
| **cfm** | 2% | 3% |

**Table 3. Zero-Vulnerability – Development Technology and Vulnerabilities**

**SSL-enabled websites**

- 44% (602) of websites are using SSL
- 81% of websites have had at least one serious* vulnerability
- 58% of websites currently have at least one serious* vulnerability
- 58% vulnerability resolution-rate among with 2,484 (out of 5,863 historical vulnerabilities) unresolved issues remaining
- Average # of serious* vulnerabilities per website during the WhiteHat Sentinel assessment lifetime: 9.7
- Average # of serious* severity unresolved vulnerabilities per website: 4.1

* *Serious vulnerabilities are those of HIGH, CRITICAL, or URGENT severity as defined by PCI-DSS naming conventions. Exploitation could lead to significant and direct business impact.*

## Conclusion

In the security industry, positive indicators are exceptionally rare. Hyped up doom and gloom headlines are the rule rather than the exception. In this case though, we have some good news to share. The good news is, as our statistics are showing, real progress of application security risk reduction can be made by organizations which truly desire to do so. Taking application security seriously is more than just spending more -- it is being strategic. With consistent outcome-based measurements and the implementation of incremental improvements to one's security controls, a dramatically increased security posture can be realized.

One thing to keep in mind is that we should not expect all security controls to yield the same outcomes, across all vulnerability classes in the same degrees, with the same investment, for every organization. We also should not expect each organization to be able to justify security investment with identical rationale as each has a different tolerance for risk. The best we can do is continue to reveal the lessons learned about how particular organizations do better than others. We'll continue to work with them to understand more deeply about what it is they are doing, identifying what is working (or not), and sharing the wisdom publicly.

## Glossary: The Top Ten Defined

### 1. Cross-Site Scripting (66% of websites)

Cross-site Scripting[4] (XSS) is easily the most prevalent website vulnerability. XSS has proven to be extremely hazardous to businesses and consumers in the form of either Web Worms[5], "Phishing with Superbait[6]" scams, Javascript malware-laced defacements, and malicious Web Widgets. The evolution of JavaScript malware, finding its way into more and more attackers toolboxes, has made finding and fixing this vulnerability more vital than ever.

### 2. Information Leakage (49% of websites)

Information Leakage[7] occurs when a website knowingly or unknowingly reveals sensitive information such as developer comments, user information, internal IP addresses, source code, software versions numbers, error messages/codes, etc., which may all aid in a targeted attack. While most of the time rated MEDIUM or LOW severity, several Information Leakage issues could be used in combination to compromise a website.

### 3. Content Spoofing (31% of websites)

Content Spoofing[8] is often used in phishing scams (or intelligence gathering) as a method of forcing a legitimate website to deliver or redirect users to bogus content. For example, users often receive a suspicious link that instructs them to confirm their user name and password information. Typically, phishing websites are hosted on look-alike domain names mimicking the content of the real site. In the case of Content spoofing phishing scams fake content is injected into the real website, making it very difficult, if not impossible, for users to detect the difference and therefore protect themselves.

### 4. Insufficient Authorization (19% of websites)

Insufficient Authorization[9] flaws are also typically found within the business logic of an application. Successful exploitation leads to an attacker being able to escalate his or her privileges, exercise unauthorized access, and potentially defraud the systems. For example, while logged-in as a normal user, an attacker could gain access to another user's data while still being logged-in under their current account.

### 5. SQL Injection (18% of websites)

SQL Injection[10] has been at the center of some of the largest credit card, identity theft incidents, and mass scale website compromises. Today's backend website databases store highly sensitive information, making them a natural, attractive target for malicious hackers. Names, addresses, phone numbers, passwords, birth dates, intellectual property, trade secrets, encryption keys and often much more could be vulnerable to theft. With a few well-placed quotes, semi-colons and commands entered into a standard Web browser entire databases could fall into the wrong hands.

### 6. Predictable Resource Location[11] (PRL) (14% of websites)

Over time, many pages on a website become unlinked, orphaned, and forgotten--especially on websites experiencing a high rate of content and/or code updates. These Web pages sometimes contain payment logs, software backups, post dated press releases, debug messages, source code – nothing or everything. Normally the only mechanism protecting the sensitive information within is the predictability of the URL. Automated scanners have become adept at uncovering these files by generating thousands of guesses.

### 7. Cross-Site Request Forgery (12% of websites)

Cross-Site Request Forgery[12] (aka Session Riding, Web Trojan, Confused Deputy, etc.) allow an attacker to force an unsuspecting user's browser to make a Web request they didn't intend. For example, the attacker could force a user to compromise their own banking, eCommerce or other website accounts invisibly without their knowledge. Since the forged request is coming the legitimate user, even when they are logged-in, the website will accept it as being the intent of that user.

### 8. Session Fixation (12% of websites)

Session Fixation is an attack technique that forces a user's session ID to an explicit value. Depending on the functionality of the target web site, a number of techniques can be utilized to "fix" the session ID value. Once the victim user authenticates in with the fixed session value, the attacker can them leverage it because the knowledge of the value.

### 9. HTTP Response Splitting (10% of websites)

HTTP Response Splitting[13] is an attack technique in which a single request is sent to the website in such a way that the response may appear to look like two. Depending on the network architecture of the website or the behavior of a users Web browser, the "second" HTTP response that's under the control of the attacker can be used to poison cache servers, deface Web pages, perform session fixation, etc.

### 10. Abuse of Functionality[14] (9% of websites)

As stated by the WASC Threat Classification "Abuse of Functionality is an attack technique that uses a website's own features and functionality to consume, defraud, or circumvent access controls mechanisms. Some functionality of a website, possibly even security features, may be abused to cause unexpected behavior. When a piece of functionality is open to abuse, an attacker could potentially annoy other users or perhaps defraud the system entirely."

## References


1 *IE8 Security Part VII: ClickJacking Defenses*
*http://blogs.msdn.com/ie/archive/2009/01/27/ie8-security-part-vii-clickjacking-defenses.aspx*

2 *OWASP HTTPOnly*
*http://www.owasp.org/index.php/HTTPOnly*

3 *PCI Data Security Standard*
*https://www.pcisecuritystandards.org/*

4 *Cross-Site Scripting*
*http://www.webappsec.org/projects/threat/classes/cross-site_scripting.shtml*

5 *Cross Site Scripting Worms and Viruses*
*http://www.whitehatsec.com/home/assets/WP5CSS0607.pdf*

6 *Phishing with Superbait*
*http://www.whitehatsec.com/home/assets/presentations/phishing_superbait.pdf*

7 *Information Leakage*
*http://www.webappsec.org/projects/threat/classes/information_leakage.shtml*

8 *Content Spoofing*
*http://www.webappsec.org/projects/threat/classes/content_spoofing.shtml*

9 *Insufficient Authorization*
*http://www.webappsec.org/projects/threat/classes/insufficient_authorization.shtml*

10 *SQL Injection*
*http://www.webappsec.org/projects/threat/classes/sql_injection.shtml*

11 *Predictable Resource Location*
*http://www.webappsec.org/projects/threat/classes/predictable_resource_location.shtml*

12 *Cross-Site Request Forgery*
*http://en.wikipedia.org/wiki/Cross-site_request_forgery*

13 *HTTP Response Splitting*
*http://www.webappsec.org/projects/threat/classes/http_response_splitting.shtml*

14 *Abuse of Functionality*
*http://www.webappsec.org/projects/threat/classes/abuse_of_functionality.shtml*

## The WhiteHat Sentinel Service – Website Risk Management

**WhiteHat Sentinel is the most accurate, complete and cost-effective website vulnerability management solution available. It delivers the flexibility, simplicity and manageability that organizations need to take control of website security and prevent Web attacks. WhiteHat Sentinel is built on a Software-as-a-Service (SaaS) platform designed from the ground up to scale massively, support the largest enterprises and offer the most compelling business efficiencies, lowering your overall cost of ownership.**

**Cost-effective Website Vulnerability Management –** As organizations struggle to maintain a strong security posture with shrinking resources, WhiteHat Sentinel has become the solution of choice for total website security at any budget level. The entire Sentinel product family is subscription-based. So, no matter how often you run your application assessments, whether it's once a week or once a month, your costs remain the same.

**Accurate –** WhiteHat Sentinel delivers the most accurate and customized website vulnerability information available– rated by both threat and severity ratings – via its unique assessment methodology. Built on the most comprehensive knowledgebase in Web application security, WhiteHat Sentinel verifies all vulnerabilities, virtually eliminating false positives. So, even with limited resources, the remediation process will be sped up by seeing only real, actionable vulnerabilities, saving both time and money, dramatically limiting exposure to attacks.

**Timely –** WhiteHat Sentinel was specifically designed to excel in rapidly-changing threat environments and dramatically narrow the window of risk by providing assessments on your schedule. Whether it's a quarterly compliance audit, new product roll-out, or weekly business-as-usual site updates, WhiteHat Sentinel can begin assessing your websites at the touch of a button.

**Complete –** WhiteHAT Sentinel was built to scale to assess hundreds, even thousands of the largest and most complex websites simultaneously. This scalability of both the methodology and the technology enables WhiteHat to streamline the process of website security. WhiteHat Sentinel was built specifically to run in both QA/development and production environments to ensure maximum coverage with no performance impact. And, WhiteHat Sentinel exceeds PCI 6.6 and 11.3.2 requirements for Web application scanning.

**Simplified Management –** WhiteHat Sentinel is turnkey – no hardware or scanning software to install requiring time-intensive configuration and management. WhiteHat Sentinel provides a comprehensive assessment, plus prioritization recommendations based on threat and severity levels, to better arm security professionals with the knowledge needed to secure an organization's data. WhiteHat Sentinel also provides a Web services API to directly integrate Sentinel vulnerability data with industry-standard bug tracking systems, or SIMs or other systems allowing you to work within your existing framework. With WhiteHat, you focus on the most important aspect of website security – fixing vulnerabilities and limiting risk.

**About WhiteHat Security, Inc.**

Headquartered in Santa Clara, California, WhiteHat Security is the leading provider of website security solutions that protect critical data, ensure compliance and narrow the window of risk. WhiteHat Sentinel, the company's flagship product family, is the most accurate, complete and cost-effective website vulnerability management solution available. It delivers the flexibility, simplicity and manageability that organizations need to take control of website security and prevent Web attacks. Furthermore, WhiteHat Sentinel enables automated mitigation of website vulnerabilities via integration with Web application firewalls. To learn more about WhiteHat Security, please visit our website at www.whitehatsec.com.

WhiteHat SECURITY
WhiteHat Security, Inc. | 3003 Bunker Hill Lane | Santa Clara, CA 95054
408.343.8300 | www.whitehatsec.com