

WhiteHat Website Security Statistics Report

August 2008
Jeremiah Grossman
Founder and CTO, WhiteHat Security

Introduction

The Web layer is the number one target for malicious online attacks. Why? Simply put, because that is where the money is¹. In the span of just a few years, Web hacking has evolved from exploration and experimentation to exploitation and monetization.

The advent of this trend can be marked by the benign Samy Worm², which compromised over one million MySpace profiles in 2005, and was motivated entirely by one man's curiosity. Today, sophisticated mass SQL Injection attacks have infected over 1.5 million Web pages worldwide³ in the last year alone, including those belonging to the Department of Homeland Security, the United Nations, and Sony, among others. Infected Web pages will foist malware upon their visitor's computers, which in turn may cause the URLs to be blacklisted,⁴ resulting in massive loss of online traffic and revenue - not to mention the costs of the cleanup effort. These days no website is considered too small or insignificant to be targeted, because just about every website can be exploited for illicit financial gain.

Cyber-criminals are eager to break into websites to access social security numbers, credit card numbers, bank account details, customer lists, early quarterly earnings reports, as well as the email addresses they hold. To reduce the risk of financial losses⁵, brand damage⁶, theft of intellectual property⁷, legal liability⁸ and fines,⁹ this data must be rigorously protected. Yet, even if this type of sensitive data is not stored on the website or is well protected, the bad guys may not need it all to profit because their real targets are the website visitors themselves. For example, the ever popular drive-by-download¹⁰ attacks are designed to exploit a user's browser and infect them with malware as soon as they arrive. The user from that point forward can be phished, passwords stolen, or even used as part of a botnet that sends out spam to infect other machines – all as a result of visiting a legitimate Web page.

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 life-cycle. Through its Software-as-a-Service (SaaS) offering, WhiteHat Sentinel¹¹, WhiteHat Security is uniquely positioned to deliver the knowledge organizations need to protect their brands, attain PCI compliance and avert costly breaches.

WhiteHat customers use Sentinel, an annual subscription service, to assess and manage their website vulnerability status. Each week, WhiteHat Sentinel assesses hundreds of public-facing and pre-production websites for vulnerabilities using a consistent and repeatable three-phase process:

1. *A proprietary automated scanning technology identifies technical vulnerabilities such as Cross-Site Scripting, SQL Injection, and many others.*
2. *WhiteHat Security Engineers create customized tests for each website to uncover business logic flaws including Insufficient Authorization, Abuse of Functionality, etc.*
3. *All results are verified to remove false-positives.*

This one-of-a-kind perspective gives WhiteHat an unparalleled view into the state of website security across vertical markets and different attack vectors, in companies of all sizes. So, whether an organization is currently starting a website security program, or has been assessing their sites for years, this report provides insight into the breadth of vulnerabilities, overall industry health, and other issues that can help focus a plan of attack and raise awareness of nascent attack trends.

It is important to note that the websites WhiteHat Sentinel manages likely represent the most "important" and "secure" websites on the Web - those conducting high-volume transactions and regulating sensitive information across the retail, finance, insurance, healthcare, and IT industries. With access to an enormous sampling of vulnerabilities in custom Web applications, we're able to publish which issues are the most prevalent on an aggregate basis. It is also important to understand the differentiation between the data contained within this document and the statistics presented in reports by Symantec¹², Mitre (CVE)¹³, IBM X-Force¹⁴, SANS¹⁵, and others. Those reports track publicly disclosed vulnerabilities in commercial and open source software. WhiteHat focuses solely on previously unknown vulnerabilities in custom Web applications, code unique to an organization, on real-world websites (Figure 1).

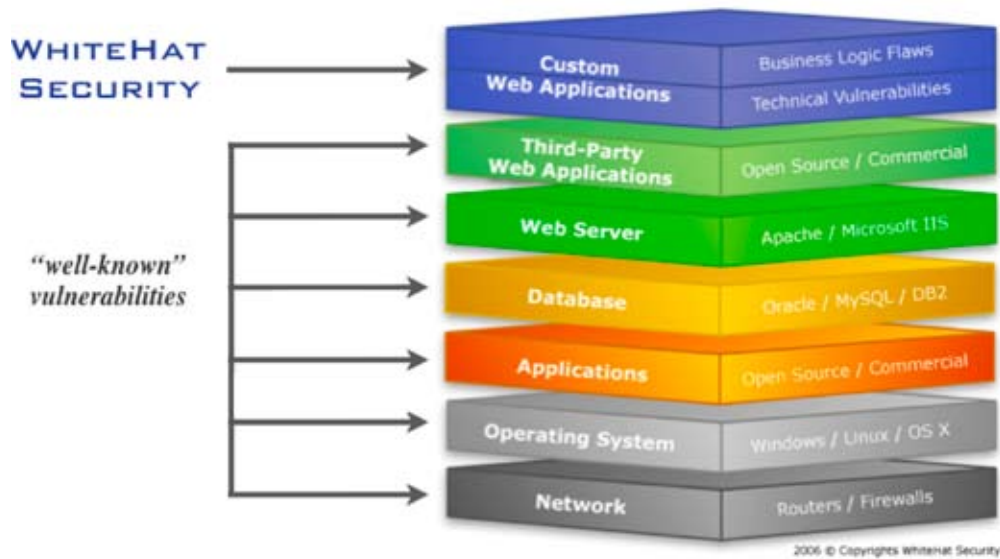


Figure 1. Software vulnerability stack

After two years of reporting on the industry, for the first time we see a positive trend--the majority of vulnerabilities discovered have been resolved. This is significant because it demonstrates that a consistent, methodical Web application security program does in fact help organizations become more secure. Consistency is important, because as the data will show, new attack techniques are constantly being tested in the wild and only a regular assessment approach will identify these new threats. PCI DSS 6.6 is also placing pressure on application security practitioners to intensify their efforts.

For a detailed description of the vulnerabilities described in this report, please see the glossary on page 11.

Data Overview

- 687 total websites
- Vast majority of websites assessed for vulnerabilities weekly
- Vulnerabilities classified according to WASC Threat Classification¹⁶
- Vulnerability severity naming convention aligns with PCI-DSS
- Obtained between January 1, 2006 and July 31, 2008

Key Findings

- Total identified vulnerabilities (open & closed): 11,234
- Current open vulnerabilities: 3,541 (66% resolved)
- 82% of assessed websites have had at least one issue
- 61% of assessed websites currently have issues of HIGH, CRITICAL, or URGENT severity
- Average of five open vulnerabilities per website
- Simple forms of SQL Injection and Cross Site Scripting (XSS) are being replaced with encoded attacks bypassing many defense measures.
- Cross Site Request Forgery (CSRF) has broken into the top 10
- Time-to-fix measurements, while still lengthy, have significantly improved.

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

- *The mix of websites includes the highly complex and interactive with a large attack surface to the static brochure-ware.*
- *Vulnerabilities are organized and counted based upon a unique Web application and class of attack. For example, if there are five possible 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).*
- *Vulnerabilities do not include “best-practices” findings. 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. As an example, the lack of encrypted passwords or data storage on the system are not considered vulnerabilities for the purpose of this report. Only issues that can be directly exploited remotely 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.” However, the vast majority of WhiteHat Sentinel customers do assess their sites on a weekly basis. When interpreting the data it is best to keep in mind that new attack vectors are always being researched by attackers, making it best to view the data as a best-case scenario based on the most up-to-date information available.*

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 Standard¹⁷ (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 settings. It should also be noted that according to the PCI-DSS, any websites with URGENT, CRITICAL, or HIGH severity issues cannot be considered compliant.

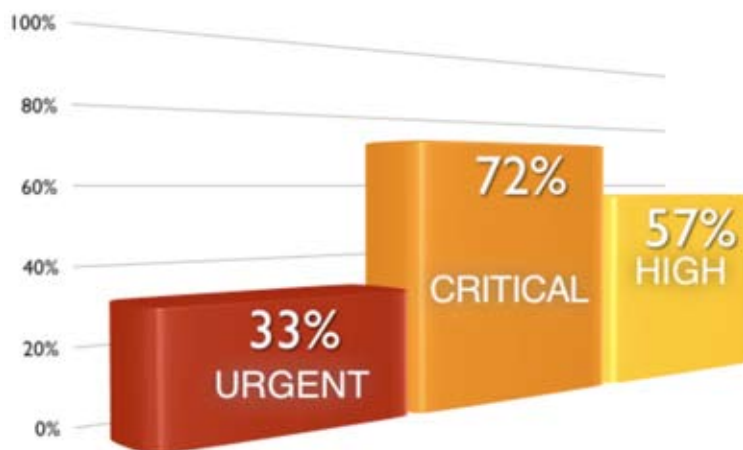


Figure 2. Percentage likelihood of websites having a least one vulnerability (sorted by severity)

While overall vulnerability counts are beginning to decline, the likelihood of websites having at least one issue of a specific severity has remained constant when compared to previous reports. Having fewer or more of each severity will not shift these figures. While having dozens of Urgent, Critical, or High severity issues makes it easier for an attacker to achieve a successful data compromise, finding and exploiting a single issue is all that is required. The closer these numbers approach to zero, the better, yet they remain unchanged.

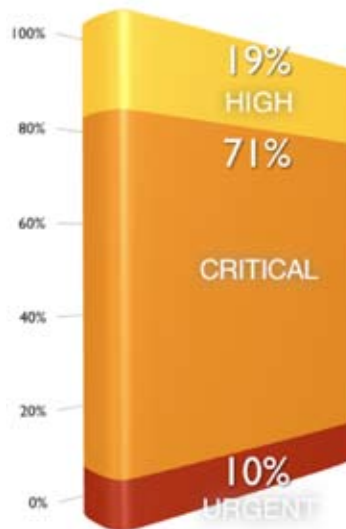


Figure 3. Percentage of vulnerabilities (sorted by severity)

As expected, the breakdown of vulnerabilities by severity has remained largely static. These numbers will track closer to industry research rather than the actual security of websites because when new attack techniques are publicly disclosed, they will automatically be identified by WhiteHat Sentinel in existing websites, even when their code has remained unchanged. However, entirely new classes of attack are rare, especially those that affect a large percentage of existing websites. It is more common that existing attack techniques are improved upon, increasing their severity in certain edge cases that must be considered on a site by site basis.

The Top Ten

WhiteHat Security determines the most prevalent issues by calculating 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.

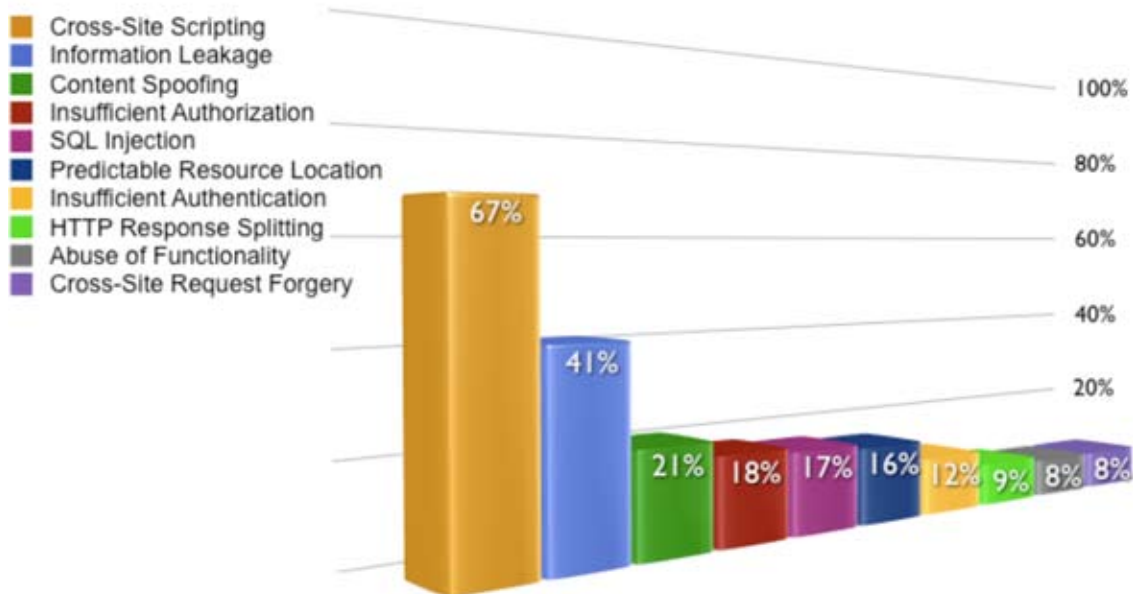


Figure 4. Top 10 vulnerability classes (sorted by percentage likelihood)

Since the last Website Security Statistics Report, the Top Ten graph has seen a few notable changes, while other areas have remained static. Most changes can be attributed to: customers resolving reported issues; increased use of modern development frameworks with native security protections built-in; and, WhiteHat's ever-evolving and improving testing methodology, an inherent benefit of a SaaS platform. Areas remain static for a variety of reasons. Typically, the cause is lack of awareness or apathy due to large numbers of outstanding issues. In other cases, remediation solutions conflict with business requirements, or the risk is deemed acceptable by the business owner (and security is put on the back burner in the interest of feature releases and the like).

As predicted, Cross-Site Request Forgery (CSRF) has managed to crack into the Top Ten by replacing Directory Indexing. All statistics surrounding CSRF can be deceptive though, including all reports issued worldwide, because identifying this issue reliably by purely automated means remains elusive - this despite vendor claims to the contrary. WhiteHat has been making steady gains in automatically detecting CSRF, but most are found through manual custom testing by WhiteHat's Security Operations Team (the same for all researchers and pen-testers globally). WhiteHat posits that a better estimate of CSRF's prevalence is similar to that of Cross-Site Scripting (XSS), appearing in approximately 75% of the world's websites.

While customers are remediating SQL Injection, XSS, and HTTP Response Splitting issues as effectively as they can, achieving 100% effectiveness has proven difficult. At the same time, new attack techniques are emerging that bypass many existing defense measures. For example, as Arian Evans (WhiteHat's Directory of Operations) detailed at BlackHat USA 2008 in his presentation entitled "Encoded, Layered and Transcoded Syntax Attacks,"¹⁸ these types of attacks are taking advantage of internationalized Web-based software.

Business Logic Flaws have remained steady in the Top Ten, demonstrating that these workflow flaws are still overlooked at many organizations, which include Insufficient Authorization, Insufficient Authentication, Abuse of Functionality, and Content Spoofing. While not at the top of the list when calculating raw numbers, these flaws are still highly prevalent across websites and can lead directly to business loss through non-sophisticated attacks. Jeremiah Grossman (WhiteHat's Chief Technology Officer) and Trey Ford (WhiteHat's Directory of Solutions Architecture), described many methods of these attacks during their BlackHat USA 2008 presentation entitled "Get Rich or Die Trying - Making Money on the Web, the Black Hat Way."¹⁹

To supplement vulnerability likelihood statistics, the following graph (Figure 5) illustrates prevalence in the overall vulnerability population. Notice how greatly the two graphs differ. The reason is that one website may possess hundreds of unique issues of a specific class, such as Cross-Site Scripting, SQL Injection, or Information Leakage, while another website may not contain any.

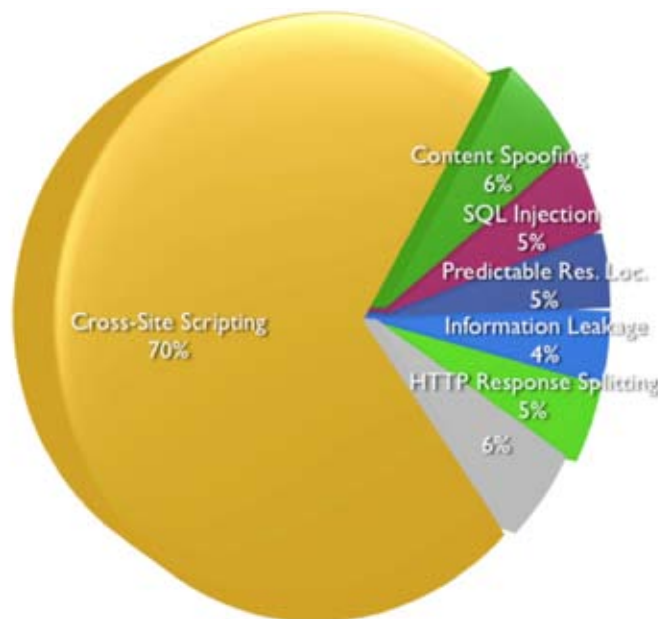


Figure 5. Vulnerability classes (stored by class)

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, probably a large supply of servlet containers. In future reports, we plan to offer likelihood of vulnerability numbers when using specific technologies.

URL Extension	% of websites	% of vulnerabilities
unknown	55%	38%
asp	27%	26%
aspx	22%	10%
jsp	8%	7%
xml	7%	1%
do	7%	4%
php	5%	2%
html	4%	2%
old	4%	1%
dll	4%	1%
cfm	3%	3%

Table 1.

Attack Surface and Number of Vulnerabilities

Application inputs are areas where arbitrary data is received, potentially leaving the software open to attack (attack surface). Application inputs include, but are not limited to, query and POST data parameter names/values, cookies, files paths/names, and so on. WhiteHat is constantly improving its ability to accurately and comprehensively identify application input points via Sentinel technology. This number comes directly from spidering all of a website’s Web pages while maintaining a logged-in state. It is possible, but not confirmed through our data, that a correlation exists between the numbers of application inputs and the number of overall vulnerabilities.

Average number of inputs per website: 290

Average ratio of vulnerability count / number of inputs: 1%

Time to Fix

When website vulnerabilities are identified, there is always 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. While the issue is being handled, an organization has four options:

- *Take the website down*
- *Revert to an older version of the website/code (if it is secure)*
- *Stay up while exposed*
- *Virtually patch the issue with a Web Application Firewall (i.e. WhiteHat Sentinel / F5 Application Security Manager integration²⁰)*

The cold reality is that vulnerabilities happen despite the most regimented software development lifecycle. Historically option #1 (taking down the website) is employed when an incident has occurred; option #2 (rolling back the code) is preferable when a hot fix is not back-ported to development and is later overwritten. Practically speaking, the vast majority of website owners default to option #3 (do nothing), essentially assuming the risk rather than halt business. Option #4 is taking hold for organizations who require immediate protection and additional time to resolve issues in the code as time and budget allow.

The remediation challenges most organizations face is the time consuming process of allocating the proper personnel, prioritizing the tasks, QA / regression testing the fix, and finally scheduling a production release. So, again the process takes time. But what is important to understand is how much.

To perform this analysis, we focused on vulnerabilities identified and resolved within the last twelve months between July 31, 2007 and July 31, 2008. The data was then sorted by the most common URGENT, CRITICAL, and HIGH severity issues. There are two aspects worth noting that may bias the sampling in opposing directions.

- *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. However, 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).*

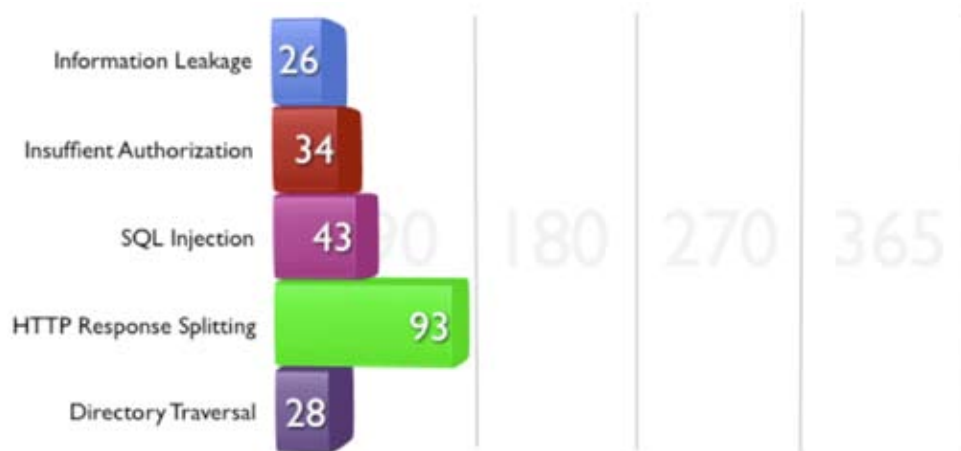


Figure 6. Average number of days for the top five URGENT severity vulnerabilities to be resolved

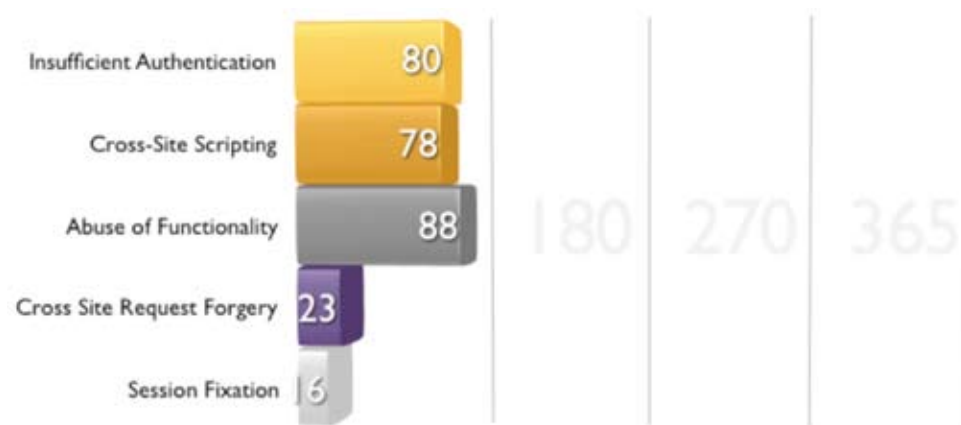


Figure 7. Average number of days for the top five CRITICAL severity vulnerabilities to be resolved

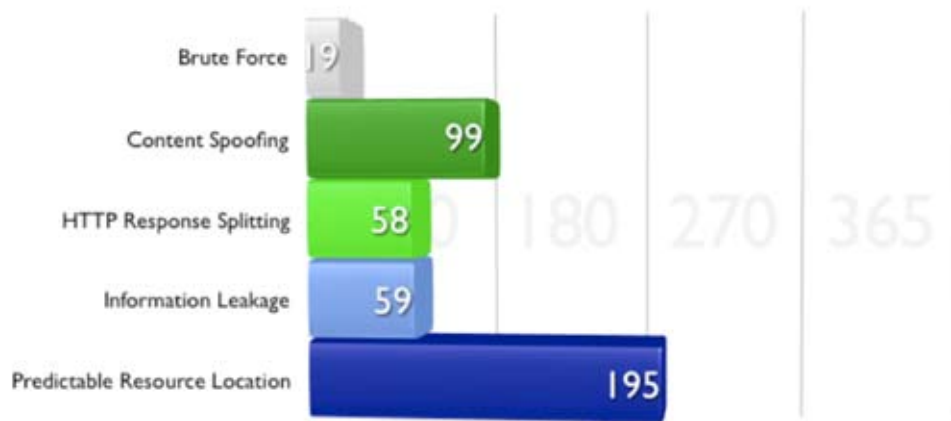


Figure 8. Average number of days for the top five HIGH severity vulnerabilities to be resolved

Class of Attack	% resolved	severity
Information Leakage	50%	urgent
Insufficient Authorization	42%	urgent
SQL Injection	66%	urgent
HTTP Response Splitting	83%	urgent
Directory Traversal	31%	urgent
Insufficient Authentication	26%	critical
Cross-Site Scripting	55%	critical
Abuse of Functionality	41%	critical
Cross-Site Request Forgery	48%	critical
Session Fixation	11%	critical
Brute Force	8%	high
Content Spoofing	26%	high
HTTP Response Splitting	31%	high
Information Leakage	34%	high
Predictable Resource Location	31%	high

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

Clearly the “time-to-fix” measurements still has room for improvement. Most Urgent severity issues are taking roughly a month to fix, while Critical issues average around three months. However, when you compare these metrics to those of the last report there is substantial improvement. IT Security and development organizations are coordinating new procedures when it comes to dealing with website vulnerabilities and are working to close the time-to-fix gap. Still, challenges remain to a speedy remediation cycle including:

- A disconnect between IT Security and the software development groups. IT Security possesses little control over the security of the website in comparison with that of the network or its hosts.
- IT Security has a difficult time explaining the details of a vulnerability to an unfamiliar audience and conveying the overall risk. This stems largely from a lack of adequate secure software development training for developers.
- The business may not allocate resources necessary to resolve the issue, instead opting to focus on features rather than vulnerability remediation.

Comparing Industry Verticals

Figure 9 shows the percentage of websites with at least one Urgent, Critical, or High severity issue sorted by industry vertical. Quickly you will see the majority of websites have these types of issues, which also would likely not allow them to be classified as PCI-DSS compliant. Beyond the generally poor state of website security, we did notice that the retail sector continues to outperform other vertical since the last report. We maintain the likely cause is that retail websites receive a larger amount of battlefield testing.

The bulk of a retail website's functionality is accessible without the need to login. This means more external attackers are able to target these websites and spot weaknesses, often exploited, which are then remedied by the organization. This is in contrast to the financial services or insurance sectors where the bulk of functionality is protected behind a login screen. So, an account is harder to access without doing business directly with the company. Once an attacker gets an account, considerably less people have tested these areas of functionality before them.

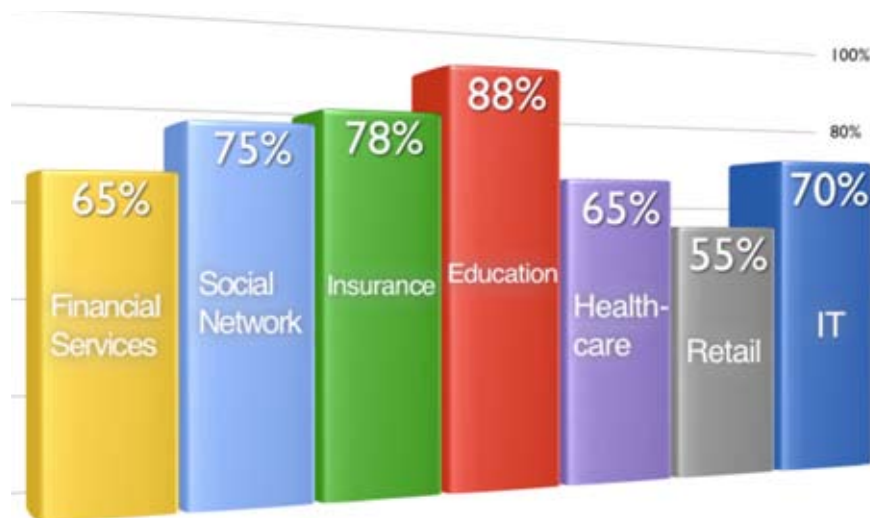


Figure 9. Percentage of websites with a URGENT, CRITICAL or HIGH severity vulnerability (sorted by industry vertical)

Conclusion

With the right solution in place to provide visibility into overall Web security posture, such as ongoing vulnerability assessments, website security can be measurably improved. Newly introduced vulnerabilities can be identified in near real-time, enabling organizations to take immediate action, empowering root-cause analysis, and demonstrate effective due diligence – something that annual consulting engagements and desktop vulnerability scanners have never been able to achieve. As evidenced in this report, WhiteHat Sentinel customers have fixed two-thirds of identified issues, done so in an increasingly speedy manner, and decreased their average number of vulnerabilities.

WhiteHat Security is dedicated to improving website security and website vulnerability management for its customers and the industry at-large. With 8 out of 10 websites vulnerable to attack, the first step toward stemming the onslaught of attacks is a thorough understanding of the nature of the problem. To make informed security decisions, enterprises require information about the vulnerabilities that exist, their impact, and how to prevent them from occurring. Through this type of industry awareness, we are determined to help organizations decrease the number and severity of vulnerabilities across the board. Organizations are encouraged to do the following:

- Find and prioritize all websites by designating their importance to the business and a party responsible for their security.
- Find and fix website vulnerabilities before the bad guys exploit them by assessing them for weaknesses with each code change. Prioritize remediation efforts based on severity, difficulty of exploitation, and business value of the website.
- Implement a secure software development process, utilizing an organizational standard development framework, scheduled developer security education program, and success incentives based upon known trouble spots.
- Utilize a defense-in-depth website security strategy that includes a Web Application Firewall, providing organization with additional security against zero-day threat and difficult to resolve issues.

Following these best practices enables organizations to conduct online business with confidence. No company can be expected to write flawless code, or have staff available around-the-clock to address all its Web application vulnerability issues, but every company needs a Web application security strategy. That is why WhiteHat created WhiteHat Sentinel, a website vulnerability management service that's customer controlled and expert managed. WhiteHat Sentinel is available 24/7, enabling companies to identify, prioritize and ultimately remediate the vulnerabilities that leave websites open to attack.

Glossary: The Top Ten Defined

- 1. Cross-Site Scripting (7 out of 10 websites)** – Cross-site Scripting²¹ (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 Worms²², “Phishing with Superbait²³” 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 (2 in 5 websites)** – Information Leakage²⁴ 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 (1 in 5 websites)** – Content Spoofing²⁵ 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 (1 in 5 websites)** – Insufficient Authorization²⁶ 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 (1 in 6 websites)** – SQL Injection²⁷ 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 Location²⁸ (PRL) (1 in 6 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. Insufficient Authentication (1 in 6 websites)** Insufficient Authentication²⁹ flaws are typically found within the business logic of an application. Successful exploitation leads to an attacker gaining unauthorized access to protected sections of a website. For example, while logged-in as a normal user, an attacker could impersonate another user on the system. These types of issues are common in financial, healthcare systems, and general content management systems where there is a high concentration of complex business logic functionality.
- 8. HTTP Response Splitting (1 in 11 websites)** – HTTP Response Splitting³⁰ 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.

9. Abuse of Functionality³¹ (1 in 12 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.”

10. Cross-Site Request Forgery (1 in 12 websites) – Cross-Site Request Forgery³² (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.

References

- 1 Get Rich or Die Trying (BlackHat USA 2008): <http://jeremiahgrossman.blogspot.com/2008/08/get-rich-or-die-trying-blackhat-usa.html>
- 2 Teen uses worm to boost ratings on MySpace.com: <http://www.computerworld.com/securitytopics/security/holes/story/0,10801,105484,00.html>
- 3 Over 1.5 million pages affected by the recent SQL injection attacks: <http://blogs.zdnet.com/security/?p=1150>
- 4 Phishers and Malware authors beware!: <http://googleonlinesecurity.blogspot.com/2007/06/phishers-and-malware-authors-beware.html>
- 5 Italian Bank’s XSS Opportunity Seized by Fraudsters: http://www.webappsec.org/projects/whid/byid_id_2008-02.shtml
- 6 Information stolen from geeks.com: http://www.webappsec.org/projects/whid/byid_id_2008-01.shtml
- 7 LexisNexis Data Breach: http://www.webappsec.org/projects/whid/byid_id_2005-65.shtml
- 8 More Social Security numbers leaked at Montana State University: http://www.webappsec.org/projects/whid/byid_id_2007-83.shtml
- 9 Guidance Software: http://www.webappsec.org/projects/whid/byid_id_2005-62.shtml
- 10 Drive-by download: http://en.wikipedia.org/wiki/Drive-by_download
- 11 WhiteHat Sentinel: <http://www.whitehatsec.com/home/services/services.html>
- 12 Internet Security Threat Report: <http://www.symantec.com/business/theme.jsp?themeid=threatreport>
- 13 Vulnerability Type Distributions in CVE: <http://cwe.mitre.org/documents/vuln-trends/index.html>
- 14 IBM Internet Security Systems X-Force® 2007 Trend Statistics: http://www.iss.net/x-force_report_images/2008/index.html
- 15 SANS Top 20: <http://www.sans.org/top20/>
- 16 WASC (Web Security Threat Classification): <http://www.webappsec.org/projects/threat/>
- 17 PCI Data Security Standard: <https://www.pcisecuritystandards.org/tech/index.htm>
- 18 Encoded, Layered and Transcoded Syntax Attacks (Black Hat USA 2008): <http://www.blackhat.com/html/bh-usa-08/bh-usa-08-speakers.html#Evans>
- 19 Get Rich or Die Trying - Making Money on the Web, the Black Hat Way (Black Hat USA 2008): <http://www.blackhat.com/html/bh-usa-08/bh-usa-08-speakers.html#Grossman>
- 20 WhiteHat Sentinel and F5 WAF Integration: <http://www.whitehatsec.com/home/assets/movies/F5WAFIntegration640.html>
- 21 Cross-Site Scripting: http://www.webappsec.org/projects/threat/classes/cross-site_scripting.shtml
- 22 Cross Site Scripting Worms and Viruses: <http://www.whitehatsec.com/home/assets/WP5CSS0607.pdf>
- 23 Phishing with Superbait: http://www.whitehatsec.com/home/assets/presentations/phishing_superbait.pdf
- 24 Information Leakage: http://www.webappsec.org/projects/threat/classes/information_leakage.shtml
- 25 Content Spoofing: http://www.webappsec.org/projects/threat/classes/content_spoofing.shtml
- 26 Insufficient Authorization: http://www.webappsec.org/projects/threat/classes/insufficient_authorization.shtml
- 27 SQL Injection: http://www.webappsec.org/projects/threat/classes/sql_injection.shtml
- 28 Predictable Resource Location: http://www.webappsec.org/projects/threat/classes/predictable_resource_location.shtml
- 29 Insufficient Authentication: http://www.webappsec.org/projects/threat/classes/insufficient_authentication.shtml
- 30 HTTP Response Splitting: http://www.webappsec.org/projects/threat/classes/http_response_splitting.shtml
- 31 Abuse of Functionality: http://www.webappsec.org/projects/threat/classes/abuse_of_functionality.shtml
- 32 Cross-Site Request Forgery: http://en.wikipedia.org/wiki/Cross-site_request_forgery

The WhiteHat Sentinel Service – Total Website Security

Find and Fix Vulnerabilities, Protect Your Website – The WhiteHat Sentinel Service is a unique combination of expert analysis and proprietary automated scanning technology that delivers the most comprehensive website vulnerability coverage available. Worried about the OWASP Top Ten vulnerabilities or the WASC Threat Classification? Scanners alone cannot identify all the vulnerabilities defined by these standards. WhiteHat Sentinel can. Many of the most dangerous vulnerabilities reside in the business logic of an application and are only uncovered through expert human analysis.

Virtually Eliminate False Positives – No busy security team has time to deal with false positives. That's why the WhiteHat Sentinel Security Operations Team verifies the results of all scans. Customers see only real, actionable vulnerabilities, saving time and money.

Virtual Patching is Now a Reality – WhiteHat Sentinel can directly configure policies on a WAF to protect against vulnerability exploits (e.g., cross-site scripting, SQL injection) that were found during the scanning process. Normally, this will be a two step process: (1) identify vulnerabilities using WhiteHat Sentinel and (2) create highly-targeted policies on WAF. This makes the process simpler for the end user – find the problem, then fix the problem with the click of a button. This integration makes “virtual patching” a reality.

Dynamic Improvement and Refinement – WhiteHat Sentinel stays one step ahead of the latest website attack vectors with persistent updates and refinements to its service. Updates are dynamic – as often as one day to several weeks, versus up to six months or longer for traditional software tools. And, Sentinel uses its unique “Inspector” technology to apply identified vulnerabilities across every website it evaluates. Ultimately, each site benefits from the protection of others.

Total Control – WhiteHat Sentinel runs on the customer's schedule, not ours. Scans can be manually or automatically scheduled to run daily, weekly, and as often as websites change. Whenever required, 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 them.

Unlimited Assessments, Anytime Websites Change – With WhiteHat Sentinel, customers pay a single annual fee, with unlimited assessments per year. And, the more applications under management with WhiteHat Sentinel, the lower the annual cost per application. High volume e-commerce sites may have weekly code changes, while others change monthly. WhiteHat Sentinel offers the flexibility to assess sites as frequent as necessary.

Simplified Management – There is no cumbersome software installation and configuration. Initial vulnerability assessments can often be up-and-running in a matter of hours. With WhiteHat Sentinel's Web interface, vulnerability data can be easily accessed, scans or print reports can be scheduled at any time from any location. No outlays for software, hardware or an engineer to run the scanner and interpret results. With the WhiteHat Sentinel Service, website vulnerability management is simplified and under control.

About WhiteHat Security, Inc.

Headquartered in Santa Clara, California, WhiteHat Security is the leading provider of SaaS based website security solutions. WhiteHat delivers turnkey solutions that enable companies to secure valuable customer data, comply with industry standards and maintain brand integrity. WhiteHat Sentinel, the company's flagship service, is the only solution that incorporates expert analysis and industry-leading technology to provide unparalleled coverage to protect critical data from attacks. For more information about WhiteHat Security, please visit our website, www.whitehatsec.com.



WhiteHat Security, Inc. | 3003 Bunker Hill Lane, Suite 220 | Santa Clara, CA 95054 | 408.343.8300 | www.whitehatsec.com

Copyright © 2008 WhiteHat Security, Inc. | Product names or brands used in this publication are for identification purposes only and may be trademarks or brands of their respective companies.

08.27.08