WhiteHat SECURITY

WHITEHAT SECURITY WEBSITE STATISTICS REPORT

How Does Your Website Security Stack Up Against Your Peers?

Summer 2012
12th Edition

Written By:
Jeremiah Grossman
Founder and Chief Technical Officer
WhiteHat Security

WhiteHat has been publishing the Website Security Statistics 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 7,000 websites, across hundreds of organization, and is accompanied by WhiteHat expert analysis and recommendations. WhiteHat's report is the only one that focuses solely on unknown vulnerabilities in custom Web applications, code unique to an organization, within real-world websites and does so over time.

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 compliance and avert costly breaches. The WhiteHat's 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 only sustainable competitive advantage is the ability to learn faster than your competitors."

-Arie de Geus

INTRODUCTION

You've seen the headlines: Moody's, AT&T, The Washington Post, PBS, Sega, Nintendo, Gawker, AT&T, the CIA, the US Senate, NASA, and Zynga. The list of website breaches goes on and on, growing every day. No organization is immune, regardless of industry. Hundreds of millions of lost credit-card numbers, hundreds of millions of personal records exposed, gigabytes worth of intellectual property stolen, and defacement galore. The time for using "No one would want to attack us" as a security strategy is clearly over, if it was ever true to begin with. Any company doing business online has something worth hacking into.

The time for using "No one would want to attack us" as a security strategy is clearly over, if it was ever true to begin with.

However, there is good news. WhiteHat's 12th Website Security Statistics Report, based on assessments of more than 7,000, websites reveals a significant reduction in the number of serious vulnerabilities exposing organizations to risk. At the same time, Web application firewalls are offering great opportunities for improvement in mitigation while overall remediation rates are increasing. Great strides are being made in remediating potentially devastating vulnerabilities across verticals and decreasing the Window of Exposure that places enterprise and customer data at risk. The report also uncovers the most secure (and insecure) vertical markets and the Windows of Exposure of each, including the Energy and Non-Profit verticals for the first time.

WHAT'S NEW

WhiteHat's 12th Website Security Statistics Report represents far and away the largest amount of data we've ever analyzed – hundreds of terabytes worth. Just in terms of the total number of websites, it's over twice the number since our last report. It's easily the most complete and longest running study focused on the state of website security.

Within this report we are very excited by the introduction of two new industries, Energy and Non-Profit. Historically, WhiteHat has reported vulnerability metrics generalized across industries. This increased website diversity increases our ability to share lessons learned.

In addition, we're also publishing several new metrics that provide powerful insight into enterprise security improvement such as Vulnerability Re-Open Rates, Industry Scorecards, Remediation Rates and Time-to-Fix metrics by vulnerability class, more consistent use of standard deviation, and even breaking out SQL Injection vulnerabilities that are exploitable in an unauthenticated state.

There is a wealth of information to be learned about website security from this report, and will no doubt lead to many more meaningful questions. Every report advances our understanding of how to make measurable improvement going forward.

KEY FINDINGS IN 2011

1. The average number of serious* vulnerabilities found per website per year was 79, a significant reduction from 230 in 2010 and down from 1,111 in 2007.
2. Cross-Site Scripting reclaimed its title as the most prevalent website vulnerability, identified in 55% of websites.
3. Web Application Firewalls could have helped mitigate the risk of at least 71% of all custom Web application vulnerabilities identified.

Website security is a moving target. New attacks techniques are frequently disclosed. New website launches are common. New Web technologies are made available every day. New application code is released constantly. Enterprises need timely information about how they can best defend their websites, gain visibility into their vulnerability lifecycle, measure the performance of their security programs, and determine how they compare to their industry peers. Establishing these metrics is crucial towards improving enterprise security.

4. There was notable improvement across all verticals, but Banking websites possessed the fewest amount of security issues of any industry with an average of 17 serious* vulnerabilities identified per website.
5. Serious* vulnerabilities were fixed in an average of 38 days or faster, a vast improvement over the 116 days it took during 2010.
6. The overall percentage of serious* vulnerabilities that were fixed was 63%, up from 53% in 2010, and a marked improvement from 2007 when it was just 35%. A rough 7% average improvement per year over each of the last four years.
7. The higher severity that a vulnerability has, the higher the likelihood that the vulnerability will reopen. Urgent: 23%, Critical: 22%, High: 15%.
8. The average number of days a website was exposed to at least one serious* vulnerability improved slightly to 231 days in 2011, from 233 days in 2010.

EXTENDED HIGHLIGHTS AND INSIGHTS

1. The favorite vulnerability class of malicious hackers, SQL Injection, remained the 8th most prevalent website vulnerability, even though it dropped 3 by points to 11% of websites.
2. 55% of SQL Injection vulnerabilities were fixed (down from 56%) and to do so required an average of 53 days (down from 57 days).
3. 5% of all websites had at least one SQL Injection vulnerability exposed that was exploitable without first needing to login to the website. This could help explain the ongoing problem of website infections and drive-by-downloads.
4. 48% of Cross-Site Scripting vulnerabilities were fixed (down from 50%) and to do so required an average of 65 days (down from 64 days).
5. About a quarter of SQL Injection and Cross-Site Scripting vulnerabilities have been reopened at 26% and 24% respectively.
6. Of the total population of vulnerabilities identified, Cross-Site Scripting, Information Leakage, and Content Spoofing took the top three spots at 50%, 14%, and 9% of the respectively.
7. Retail websites improved dramatically over the last year, yet remain the industry possessing the most security issues with an average of 121 serious* vulnerabilities identified per website.
8. The industries that fixed their serious* vulnerabilities the fastest were Energy (4 days), Manufacturing (17 days), and Retail (27 days).
9. The industries that fixed their serious* vulnerabilities the slowest were Non-Profit (94 days), Financial Services (80 days), and Telecommunications (50 days) websites.
10. The industries that remediated the largest percentage of their serious* vulnerabilities were Banking (74%), Telecommunications (69%), and Retail (66%) websites.
11. The industries that remediated the fewest percentage of their serious* vulnerabilities were Energy (40%), Education (46%), and Manufacturing (50%) websites.
12. The industry with the fewest days exposed to at least one serious* vulnerability was Banking at 185 days, but was oddly way up from 74 days during 2012
13. The industry with the most days exposed to at least one serious* vulnerability was Non-Profit websites at 320 days, followed by Education websites at 261 days (up from 164) and Social Networking at 264 days (up from 159).

14. 20% of vulnerabilities identified by WhiteHat Sentinel have been reopened as some point in time, often several times.
15. Vulnerability classes that tend to be exploited by injecting malicious data into URL parameters tend to reopen more often than business logic flaws.
16. OS Command Injection, by comparable vulnerability volume, is statistically non-existent.

AT A GLANCE: THE CURRENT STATE OF WEBSITE SECURITY

Figure 1. Vulnerability Historical Trend
The annual average number of serious* vulnerabilities discovered per website per year

There is a significant drop in the average number of serious* vulnerabilities found per website per year – from 230 identified in 2010 to 79 in 2011. This is much reduced from over a thousand vulnerabilities back in 2007.

While this vulnerability reduction trend is welcome news, there are several possible explanations that must be taken into consideration as the "real" numbers may not be as rosy.

There is a significant drop in the average number of serious* vulnerabilities found per website per year — from 230 identified in 2010 to 79 in 2011.

To preface, some explanations we have first hand knowledge of, others we do not, but we'll do our best to provide accurate contextual analysis. We believe the reduction in vulnerabilities is a combination of all the factors below.

1. Despite the plethora of recent breach headlines, websites could in fact be getting more "secure" – that is to say, less vulnerable. At the same time, notice the huge standard deviation of 670 in Figure 2. While we're certain improvement in website security is part of the answer, at least within the WhiteHat Sentinel customer-base, there is still a great number of websites with hundreds of serious* vulnerabilities. We also know website security is greatly influenced by compliance obligations, customer and partner security requirements, community awareness campaigns, and of course attackers making their presence felt. All of these things serve to improve website security.
2. Another possible explanation is that organizations are more often choosing a less comprehensive form of vulnerability assessment, such as WhiteHat Sentinel Standard or Baseline over Premium Edition. Obviously when we look for fewer issues, we'll find fewer. This may affect our report's statistics to some degree, particularly the vulnerability totals. This is why we routinely remind readers that this is report describes a best-case scenario. Websites are, at a minimum, THIS vulnerable. The same is true for any industry report of this kind.

There is some important background for how, when, and why organizations select particular WhiteHat Sentinel Service lines. Roughly five years ago, it was standard practice for customers to identify their most important websites, a subset of their total asset inventory, and invest the bulk of their resources defending those. This is generally referred to as the "Depth" strategy. If implemented well, the Depth strategy would protect those high value website assets, but would leave the secondary and tertiary websites largely unprotected. This is where attackers, including the infamous mass SQL Injection worms, had much success in injecting malware and infecting visitors.

In an effort to counteract these types of threats, 2-3 years ago enterprises began shifting their WhiteHat Sentinel deployment strategy to one of Breadth rather than Depth. They would select Standard Edition or Baseline Edition and treat all websites the same with the goal of elevating the minimum bar across the board. Compliance obligations, such as PCI-DSS, also motivated the same organization behavior. The Breadth strategy helped to keep out the mass SQL Injection worm traffic and other fully automated threats, but an attacker who wanted to press a little bit harder and exercise some human intelligence could still succeed.

Today, we're seeing a hybrid approach between Depth and Breadth. Enterprises with anything more than a handful of websites now tend to deploy WhiteHat Sentinel Baseline Edition across everything, then elevate coverage to Standard or Premium Edition appropriately to match the complexity and value of the website. In this way organizations strategically match the value of a website and their tolerance for risk with the properly aligned WhiteHat Sentinel Service line.

3. The third factor that could potentially account for the overall vulnerability reduction is that our sampling of websites, especially early on, was not representative. As of June 2012, Netcraft says us there are roughly 697 million websites[1] and increasing in the tens of millions per month. While we are assessing far more websites and far more often than anyone else, it is still a tiny fraction of the whole. It could also be that historically our customers only provided us their most insecure websites first. While there may be some truth to this, we've also seen other reports released by our peers and their numbers are not far off from our own.

While overall vulnerability reduction is a welcome sign, the sheer number of serious* vulnerabilities is still quite stunning. To better understand these numbers, certain variables need to be taken into account, particularly that Sentinel counts individual URL parameters. Consider all the discrete Web applications that make up a website, the many input parameters each has, the many ways each parameter may be exploited by several dozen vulnerabilities, multiply that over a year with frequent code updates. Within that frame of reference, the volume of vulnerabilities may not appear so unreasonable.

Industry	Annual Avg. Vulnerabilities	Std. Dev	Avg. Time-to-Fix (Days)	Average Remediation	Std. Dev	Window of Exposure (Days)	Std. Dev
ALL	79	670	38	63%	36	231	159
Banking	17	554	45	74%	37	185	147
Education	53	885	30	46%	37	261	153
Financial Services	67	853	80	63%	35	227	157
Healthcare	48	461	35	63%	36	239	155
Insurance	92	171	40	58%	32	211	154
IT	85	36	35	57%	31	208	159
Manufacturing	30	56	17	50%	33	252	125
Retail	121	125	27	66%	36	238	160
Social Networking	31	431	41	62%	43	264	162
Telecom	52	82	50	69%	31	271	136
Non-Profit	37	56	94	56%	40	320	168
Energy	31	62	4	40%	35	250	154

Figure 2. At a Glance: The Current State of Website Security (2011) (Sorted by industry)

Vulnerability counts alone do not provide a clear picture of the current state of website security. For this it is necessary to consider the average number of days it takes to fix a serious* vulnerability (Time-to-Fix), the percentage of reported vulnerabilities that are no longer exploitable (Remediation Rate), and the average number of days a website is exposed to at least one serious* vulnerability (Window-of-Exposure). As these metrics are tallied at each organization, specific operational and software development lifecycle deficiencies can be isolated and improved.

It is difficult to conclusively state why certain industries perform better than others, or even why a particular development group in a single organization seemingly performs better than the rest. All told, the data shows that the industry in which a particular website falls seems to, at best, only slightly correlate to an expected security posture. Previous reports have also explored potential correlations that may exist between organization size and development framework / programming language in use, but only small performance variations emerge. Clearly, some organizations have something figured out as their websites are far less vulnerable than others.

Clearly, some organizations have something figured out as their websites are far less vulnerable than others.

What makes a website more secure than another? What behaviors should an organization prioritize first? This remains a question we seek to answer with data.

WHITEHAT SECURITY TOP TEN

Now that we have an overview of the average total number of serious* vulnerabilities, Time-to-Fix, Remediation Rates, and Window of Exposure across industry verticals we'll look at the distribution of vulnerability classes. In Figure 3 the most prevalent vulnerabilities classes are calculated based upon their percentage likelihood of at least one instance being found within any given website. This approach minimizes data skewing in websites that are either highly secure or extremely risk-prone.

1. *In 2010, 64% of websites had at least one Information Leakage vulnerability, which overtook the notorious Cross-Site Scripting (XSS) as the most prevalent issue by just a few tenths of a percent. During 2011, Information Leakage and XSS switched top spots again and both vulnerability classes saw a notable reduction. In 2011,* ***XSS regained its title as the most prevalent website vulnerability being found in 55% of websites****. In the second place on the WhiteHat Top Ten, Information Leakage, identified in 53% of websites.*

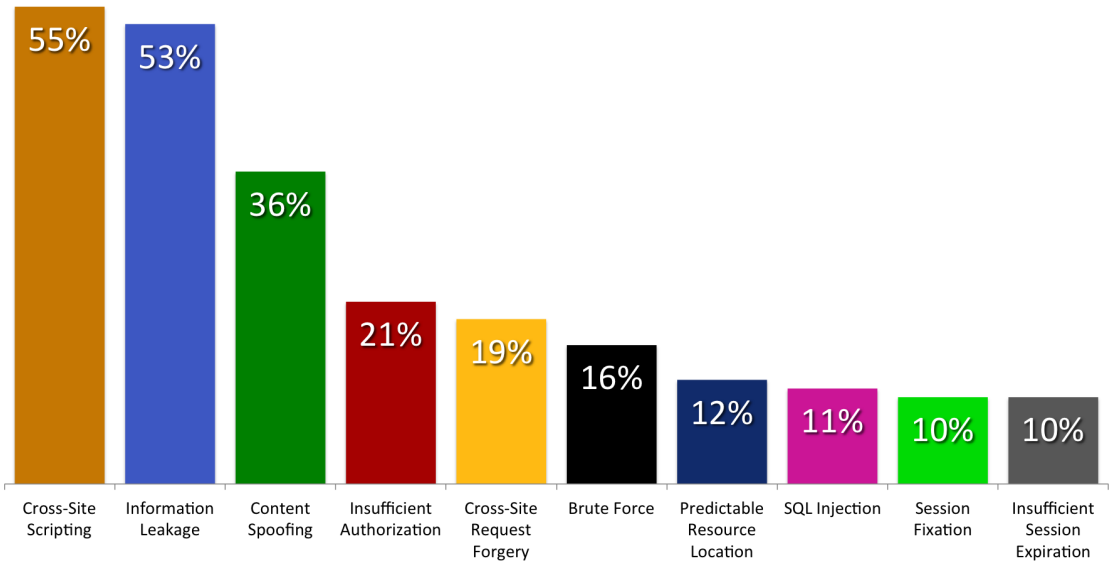


Figure 3. Top Ten Vulnerability Classes (2011)
Percentage likelihood that at least one serious* vulnerability will appear in a website

2. Information Leakage is a catchall term that describes a vulnerability in which a website reveals sensitive data, such as technical details of the Web application, environment, or user-specific data. Sensitive data may be used by an attacker to exploit the system, its hosting network, or users. Common examples are a failure to scrub out HTML/JavaScript comments containing sensitive information (database passwords), improper application or server configurations, or differences in page responses for valid versus invalid data.

3. Also on a downward trend, but remaining at #3, is Content Spoofing at 36% of websites (43% in 2010). Content Spoofing is a very similar vulnerability to Cross-Site Scripting, only without the "script" (HTML, JavaScript). This vulnerability class is most often used to force a website to display something unauthorized to another user. As such, Content Spoofing useful in performing phishing scams and other malicious brand attacks.

4. & 5. Unexpectedly, Cross-Site Request Forgery (CSRF) slid down a spot down to #5 in 2011, overtaken by Insufficient Authorization (21% of websites). CSRF also fell five percentage points to 19% of websites. This is odd because CSRF is widely considered to be the sleeping giant[2] of Web security with the industry consensus asserting that nearly every website has at least one vulnerability. Therefore, we predicted the numbers to go up, and for years they did. To reiterate from our last report, this has nothing to do with websites somehow becoming more vulnerable to CSRF:

> Only a few years back, CSRF was widely disregarded as not a "real vulnerability" and considered an artifact of "the way the Web was designed to work." Over time malicious hacker activity leveraging CSRF has forcibly changed this perception and more website owners are asking them to be reported so they may be fixed.
>
> Instead, a steady improvement in WhiteHat Sentinel identification combined with customer demand to report them accounts for the rise. CSRF attacks involve forcing a victim's Web browser, typically while authenticated, to send an HTTP request to a target website without their knowledge to perform an unintended action as the victim. This action could be a bank wire transfer, email spam, add a friend, and so on. Practically speaking, just about every feature on every website has the potential of being vulnerable to CSRF unless very specific safeguards are put in place.

The best explanation for the reduction in CSRF vulnerabilities is the particular WhiteHat Sentinel Service selected by our customers. There is an increased representation of customer websites in the sample covered by Sentinel Baseline and Standard Edition, which are either configured/performed in an unauthenticated fashion and/or do not comprehensively check for CSRF vulnerabilities. With these observations, the reduction in CSRF could have been foreseen. Scanning for CSRF in a purely automated fashion is well-known to be extremely difficult and false-positive prone, which is why we recommend expert testing as provide with Premium Edition to identify the vulnerability[3]. For these reasons we maintain that CSRF is probably the most prevalent website vulnerability, or a close second, in real terms.

6. Brute Force slipped a spot down to 6th place and dropped only a single percentage point from last year to 16% of websites. The bulk of these Brute Force vulnerabilities occur because a website login field reveals which entry of the username / password combination is incorrect. Due to spammers mining for valid email addresses, which double as usernames on a variety of websites, enterprises have an increased awareness and appreciation for the problem. In these cases we adjust the severity of Brute Force vulnerability accordingly.

7. & 8. Predictable Resource Location (PRL), which are URLs containing sensitive information that may be unlinked but whose location can be guessed, held firm in 7th place while still managing to be reduced by 2% in 2011. If XSS is the most prevalent website vulnerability, SQL Injection is likely the most exploited. Still, SQL Injection remains fixed in 8th place on the WhiteHat Top Ten and has even dropped 3 points down to 11% of websites – several times less prevalent than XSS. This should be a reminder that vulnerability prevalence does not automatically correlate to vulnerability exploitation.

9. & 10. Rounding out the Top Ten are two vulnerability classes that impact website session management, Session Fixation and Insufficient Session Expiration respectively. The former is when a website's session credential, such as a cookie, can be forcibly pre-set to a value that the bad guy knows, which then becomes valid after the user successfully authenticates. The latter is when a system does not invalidate and/or delete a session credential when a user logs-out of the system and it's validity persists. At 10% of websites a piece, both of these are neither rare or strangers to the Top Ten.

By and large the 2011 WhiteHat Top Ten contains many common features with 2010. While progress is being made, wiping out particular vulnerability classes globally, even the most well-known is proving to be a monstrously difficult task. The WhiteHat Threat Research Center credits the reduction in the top three vulnerability classes to increased awareness, stronger development frameworks, and Web Application Firewalls are making a positive impact. The backlog of work required to repair up to 20 years of vulnerable Web code will take a while.

SQL Injection: Unauthenticated

One of the most devastating scourges on the Web is the formidable mass SQL Injection worm, with the most recognizable being LizaMoon[4]. As a class of adversary, SQL Injection worms differ in several ways from other forms of SQL Injection attack.

The first difference is that SQL Injection worms are fully automated and their website targets are chosen indiscriminately. Other varieties of SQL Injection attacks are carried out by a sentient attacker who selects targets far more carefully.

Secondly, worms do not extract data from their website victims. Instead they inject content in the form of malware designed to infect visitors. Sentient SQL Injection attacks do steal data, particularly usernames and passwords, credit card details, and other sensitive information.

The third difference is mass SQL injection worms do not registered user accounts, do not login, do not fill out Web forms with valid information, and as a consequence, their "scans" run completely unauthenticated — anonymous as it were. Targeted SQL Injection attacks behave precisely the opposite. They'll certainly register accounts, login, fill out Web forms if they need to, and scan as an authenticated user. This details of "authenticated" attacks is interesting to consider, particular the rough percentage of websites that have "unauthenticated" SQL Injection vulnerabilities exposed.

To get this figure we counted vulnerabilities identified by WhiteHat Sentinel Baseline Edition (BE), which is a service line where scans are not authenticated and Web forms are not configured. BE is generally for customers who require a level of testing comprehensiveness that meets or exceeds that of unauthenticated and opportunistic attacks like a mass SQL Injection worm. When BE is deployed it is usually done so broadly across an entire website portfolio before increasing the level of service on more valued websites.

Of all BE websites covered under WhiteHat Sentinel before March 2011, which encompassed many hundreds, 5% of websites have had at least one SQL Injection vulnerability without needing to login. This could help to explain why mass SQL Injection worms have been so successful, now infecting millions of URLs across the Web[5].

There is one important potential caveat in the data that may cause the 5% number to rise, even if only slightly. Just because a vulnerability assessment is conducted in a logged-in state does not mean the URL that's vulnerable to SQL Injection can't be exploited while NOT logged-in — an authentication / authorization issue. If you notice, Insufficient Authorization is #4 on the WhiteHat Security Top Ten at 21% of websites.

VULNERABILITY POPULATION

To supplement vulnerability likelihood statistics, the following graph (Figure 4) illustrates prevalence by class in the overall vulnerability population. Notice how greatly it differs from the WhiteHat Top Ten. The reason is that one website may possess hundreds of unique issues of a specific class, while another website may not contain any.

Vulnerability classes Cross-Site Scripting, Information Leakage, and Content Spoofing take the top three spots as they represent 50%, 14%, and 9% of the total population respectively. Also noticeable on the list are Cross-Site Request Forgery, Insufficient Authorization, and SQL Injection all at 4%.

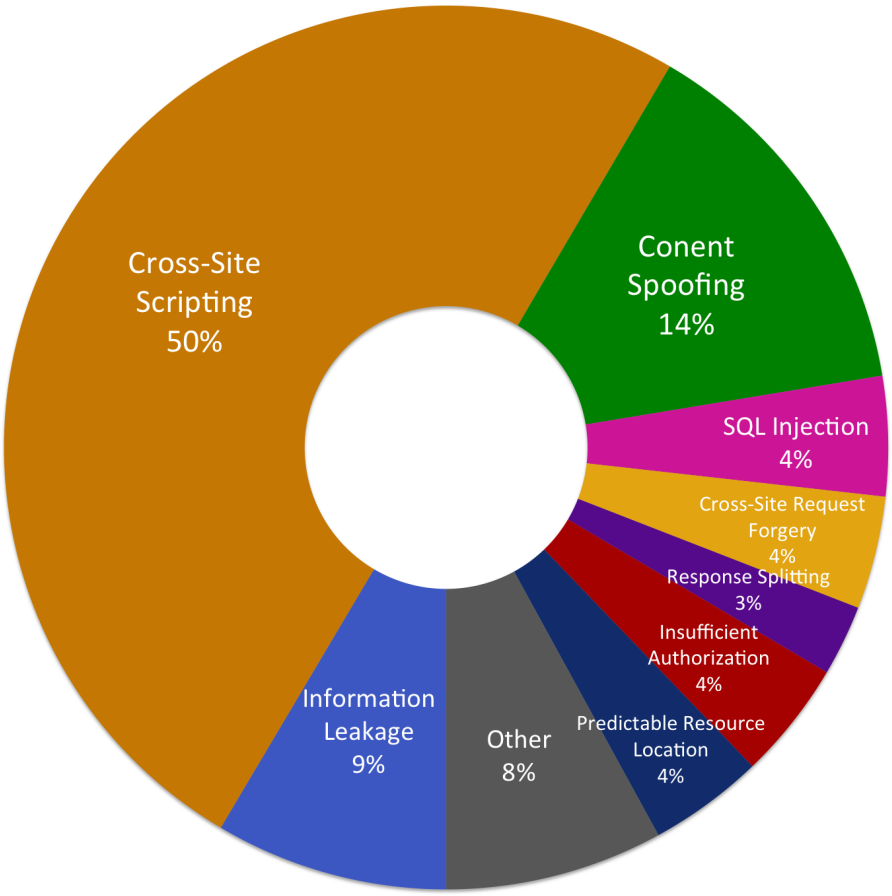


Figure 4. Overall Vulnerability Population (2011)
Percentage breakdown of all the serious* vulnerabilities discovered
(Sorted by vulnerability class)

WEB APPLICATION FIREWALLS BEGIN TO MAKE A DIFFERENCE

Since 2010, Web Application Firewalls (WAF) have seen increased enterprise deployment. Of a WAFs list of positive attributes, one of the most attractive is their ability to strongly mitigate particular vulnerability classes. Not coincidentally, the vulnerabilities WAFs are best at mitigating, such as Cross-Site Scripting, Content Spoofing, SQL Injection, Response Splitting, etc., greatly overlap with the Overall Vulnerability Population. By summing all these percentages up we could safely say that a WAF could feasible help mitigate the risk of at least 71% of all custom Web application vulnerabilities.

It is not uncommon for WAFs to be configured to pull in an XML vulnerability data feed from WhiteHat Sentinel and auto-generate virtual-patch rules specifically for this reason. Customers cite major benefits such as helping to meet tight compliance deadlines, improving their Time-to-Fix and Window of Exposure metrics, and stave off incoming attacks while application code is being fixed.

INDUSTRY SCORECARDS

Business managers often ask their security teams, "How are we doing? Are we safe, are we secure?" Typically, what business managers are really asking for is a sense of how the organization's current security posture compares to peers or competitors. They want to know if the organization is leading, falling way behind, or is somewhere in between. The answers to these questions are also extremely helpful for measuring a security program's effectiveness and setting goals.

Some organizations are 'targets of opportunity,' others are 'targets of choice.' Targets of opportunity are breached when their security posture is weaker than the average organization [in their industry]. Targets of choice possess some type of unique and valuable information, or perhaps a reputation or brand that is particularly attractive to a motivated attacker. Because foolproof security is an unrealistic goal, it is vital for every organization to determine if they are most likely a target of opportunity or choice. By doing so an organization may establish and measure against a "secure enough" bar.

If an organization is a target of opportunity, a goal of being just above average with respect to website security among your peers is reasonable. The bad guy will generally prefer to attack weaker, and therefore easier to breach, targets. If a target of choice, an organization must elevate its website security posture to a point where an attacker's efforts are detectable, preventable, and in case of a compromise, survivable. This is due to the fact that an adversary will spend whatever time is necessary looking for gaps in the defenses to exploit.

Whether you are a target of choice or a target of opportunity, the following Industry Scorecards will assist in comparing how an organization's security posture compares to its peers. Figure 5 then provides a simple example of how an organization may leverage the scorecards, and the data they contain, to report security metrics internally by business unit.

Group	High Severity Vulnerabilities	Avg. Time-to-Fix (Days)	Remediation Rate	Window of Exposure (Days)
2012 Goal	20	30	75%	100
Industry Average	55	32	63%	223
Division A	17	45	74%	195
Division B	53	30	46%	161
Division C	67	66	63%	237
Division D	48	35	69%	232

Figure 5. Simple Example of an Internal Website Security Scorecard

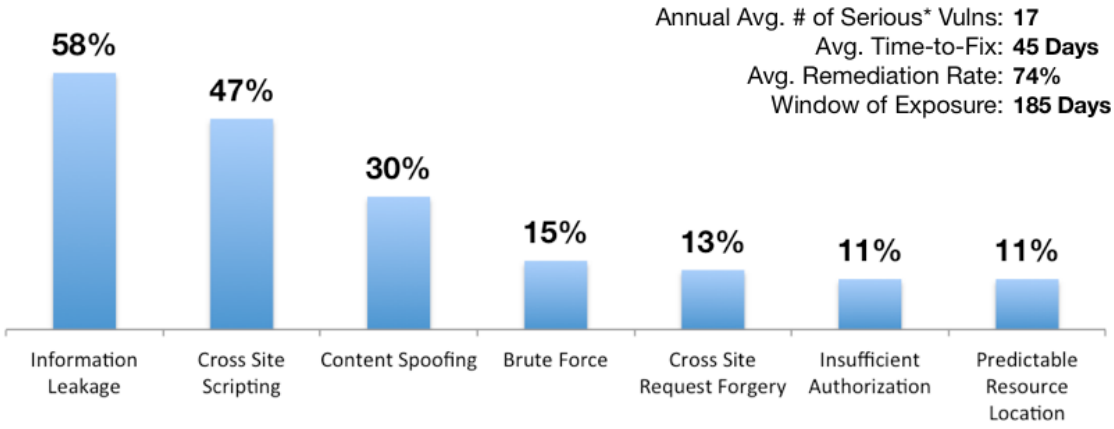


Figure 6. Banking Industry: Website Security Scorecard

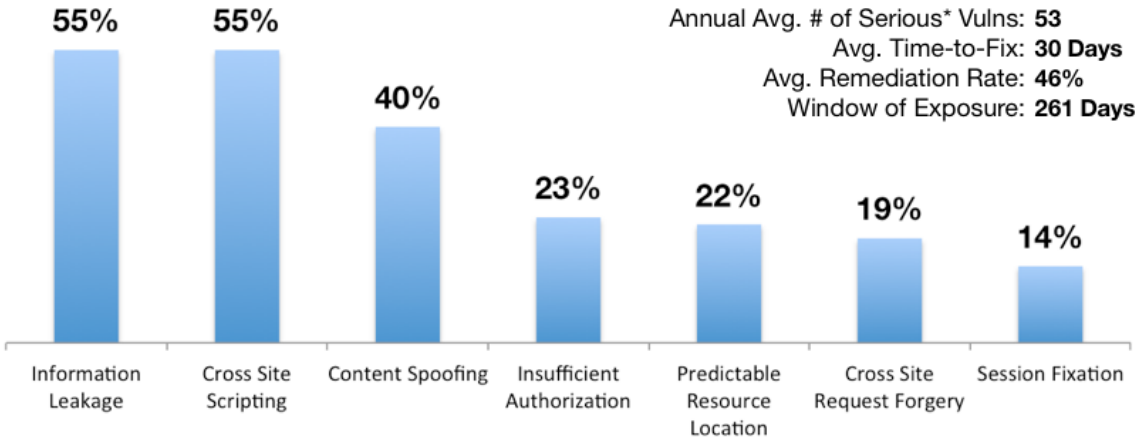


Figure 7. Education Industry: Website Security Scorecard

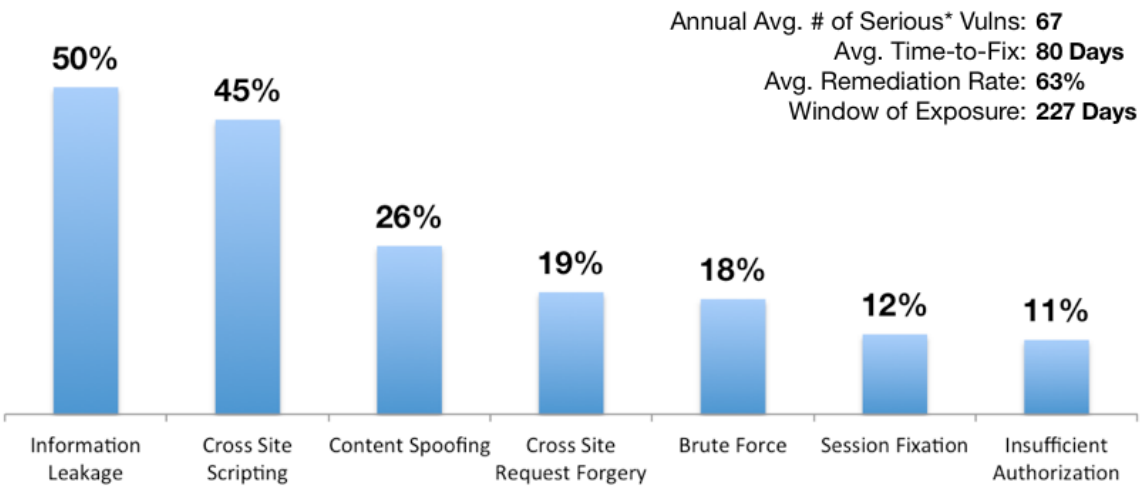


Figure 8. Financial Services Industry: Website Security Scorecard

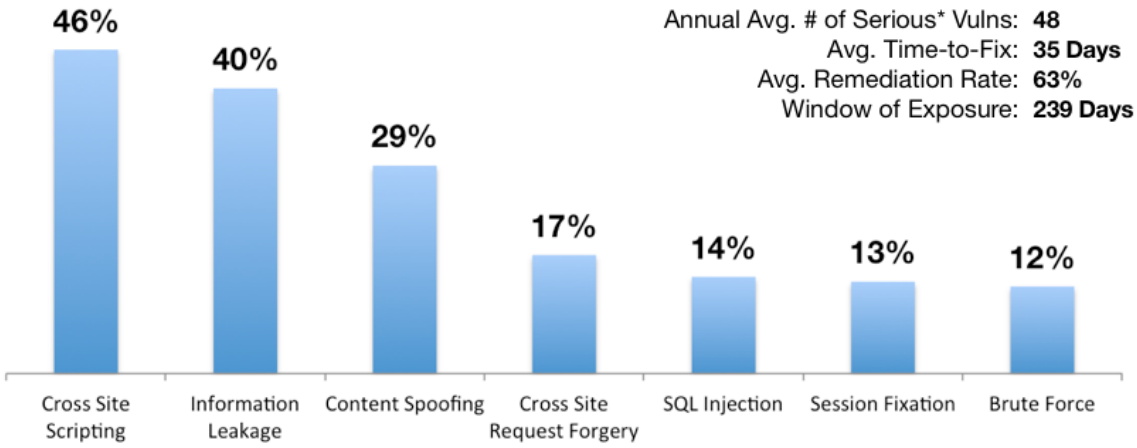


Figure 9. Healthcare Industry: Website Security Scorecard

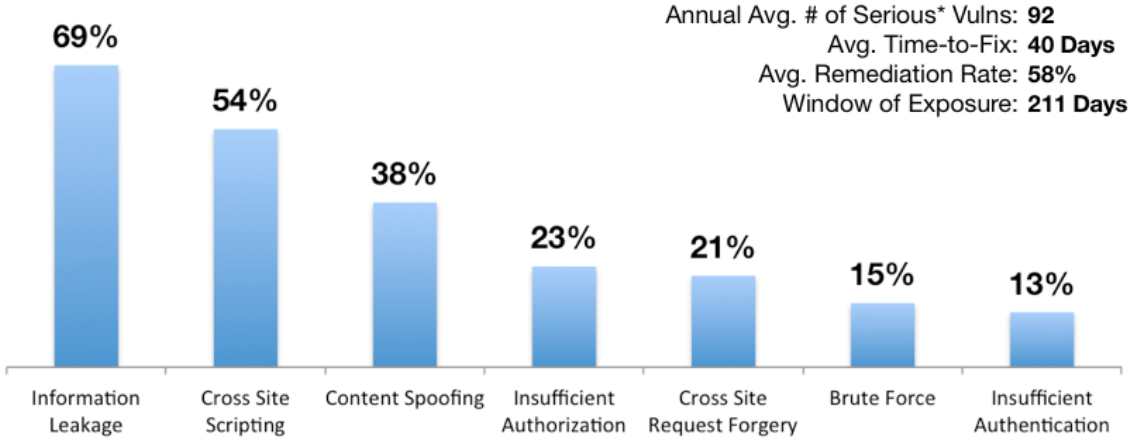


Figure 10. Insurance Industry: Website Security Scorecard

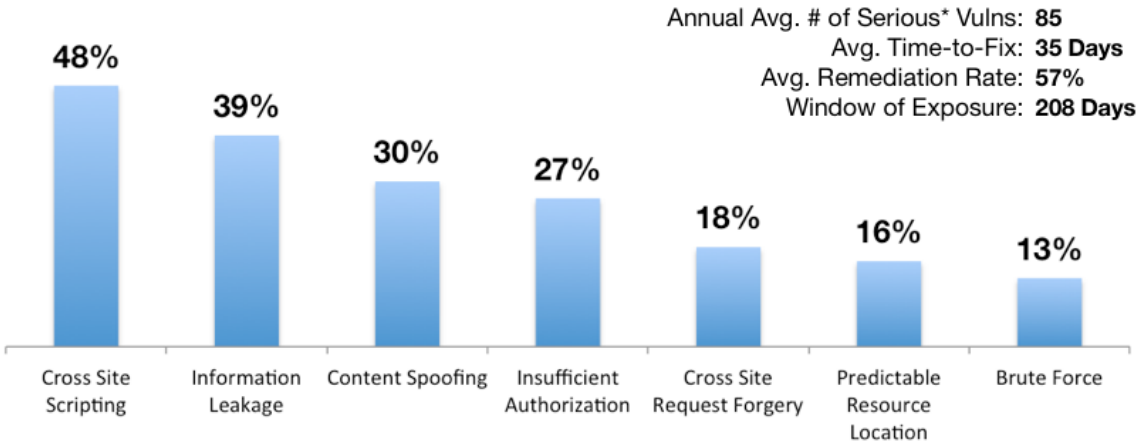


Figure 11. IT Industry: Website Security Scorecard

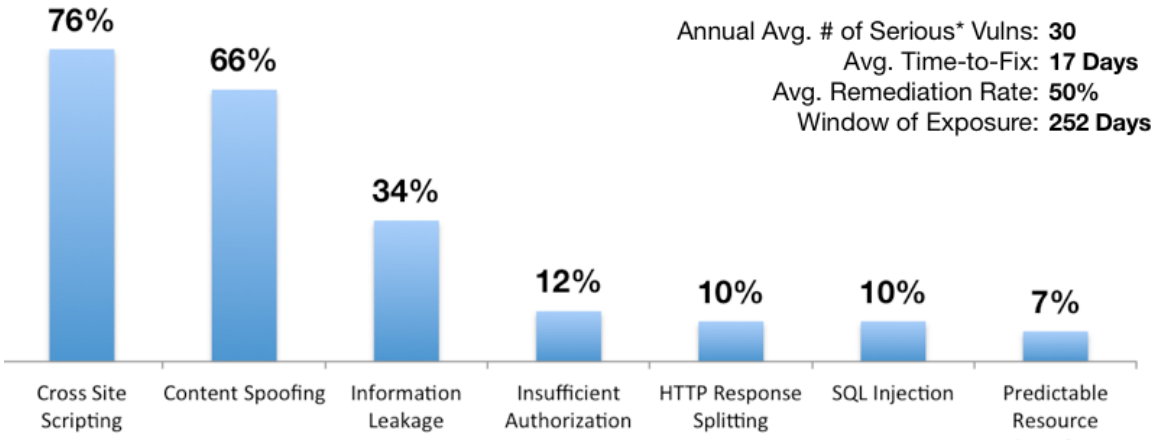


Figure 12. Manufacturing Industry: Website Security Scorecard

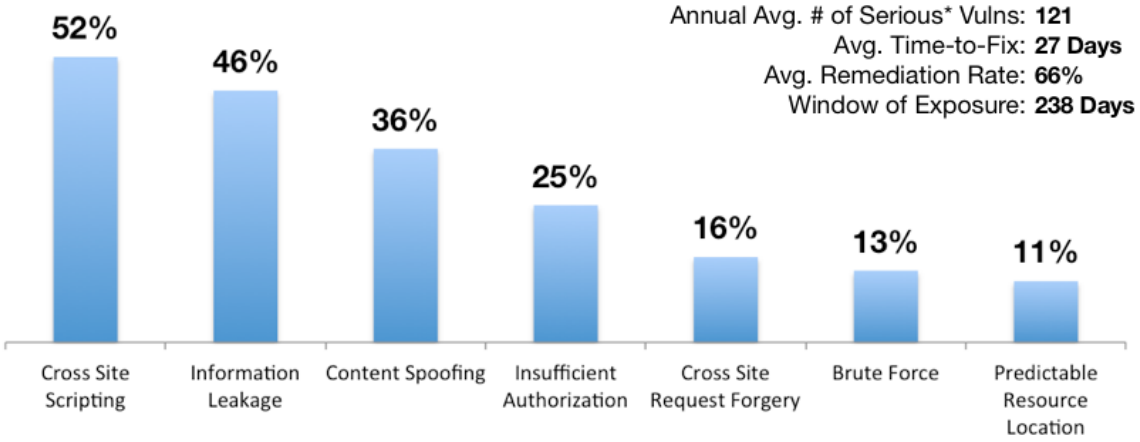


Figure 13. Retail Industry: Website Security Scorecard

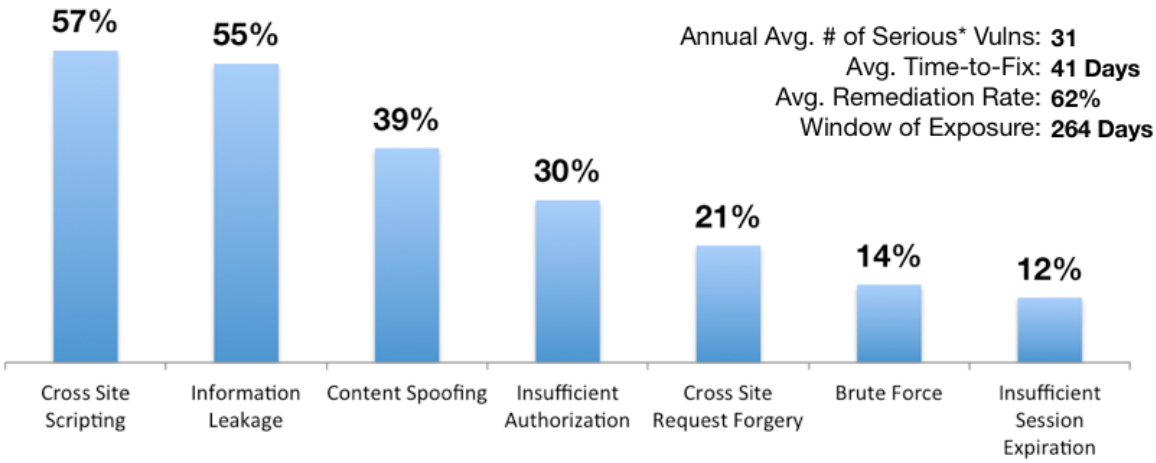


Figure 14. Social Networking Industry: Website Security Scorecard

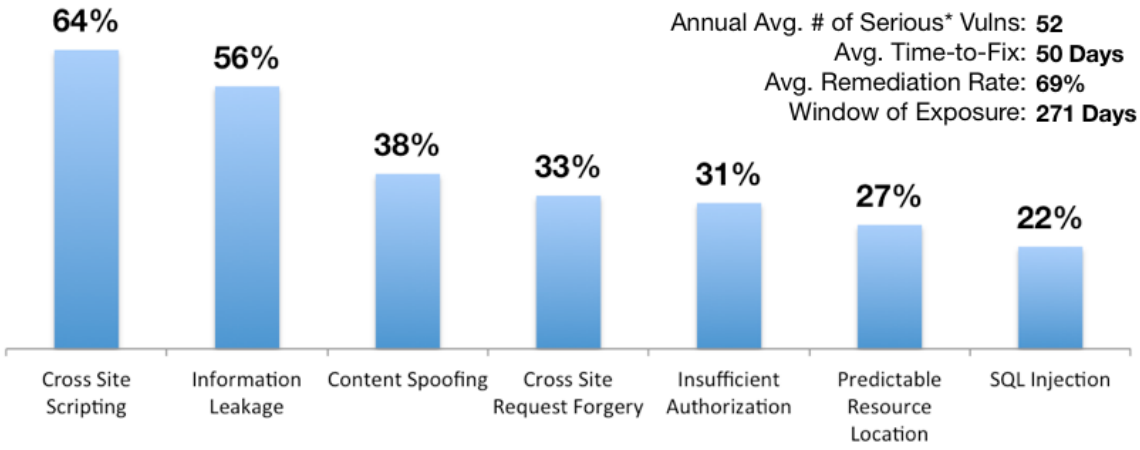


Figure 15. Telecommunications Industry: Website Security Scorecard

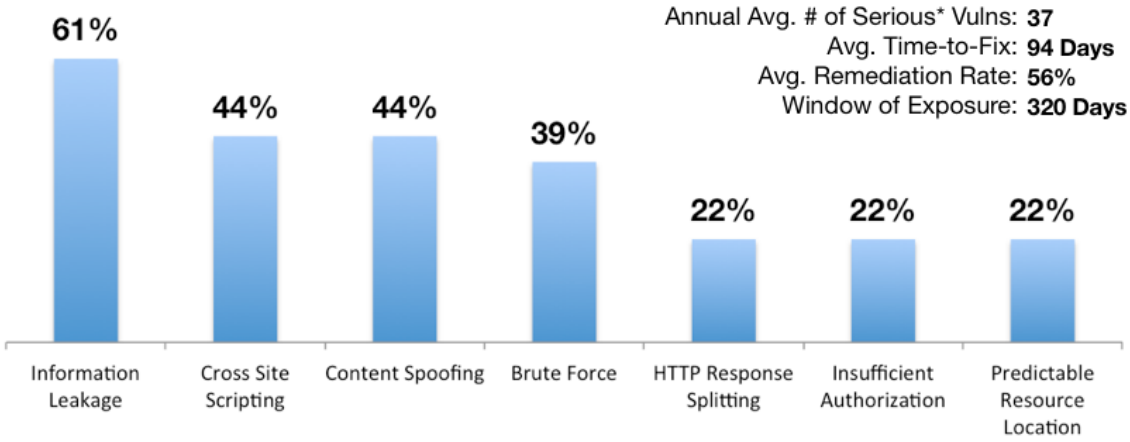


Figure 16. Non-Profit Industry: Website Security Scorecard

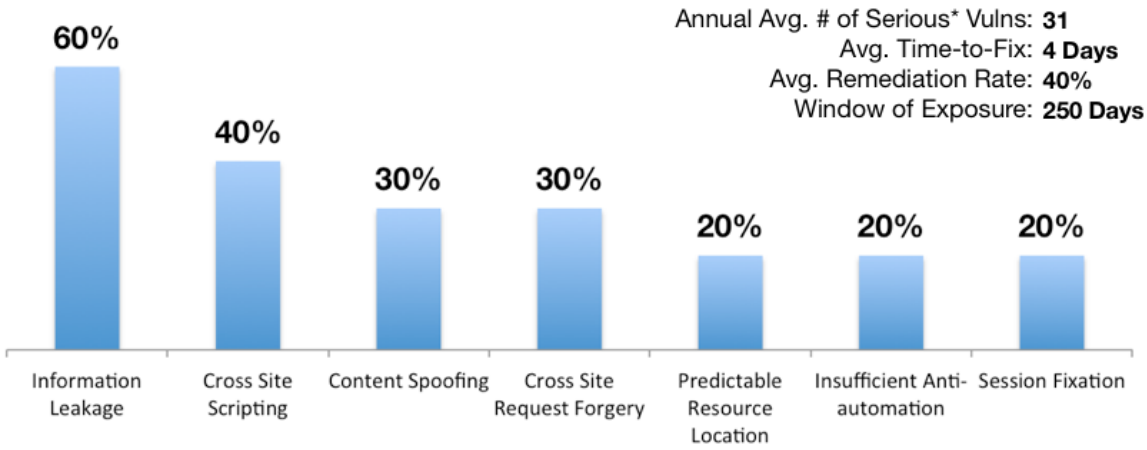


Figure 17. Energy Industry: Website Security Scorecard

INDUSTRIES COMPARED: VULNERABILITY AVERAGES

When approaching information security, it is important not to lose sight of the distinction between what is possible and what is probable. In website security, it is true that a single vulnerability represents a means for an attacker to exploit a system, but not all (vulnerabilities or attackers) are created equal. For example, SQL Injection tends to be exploited far more often than other vulnerability classes that are more common. We also know that attacks are getting more targeted and consequently having numerous serious (Urgent, Critical, or High severity) issues, whatever they be, makes it that much easier for an attacker to achieve a successful compromise. Therefore, it is best to minimize vulnerabilities to the maximum extent possible to increase application security assurance.

By comparing Figure 18 to Addendum Figure 29, "At a Glance: The Current State of Website Security (2010)", the average number of serious* vulnerabilities identified per website per year dropped noticeably from 2010 to 2011 across all industries, except for Healthcare and Insurance websites. During 2011, most industries ranged somewhere between 17 (Banking), to the low 30s (Manufacturing, Social Networking, and Energy) and up to a high of 121 (Retail). It's worth pointing out that while Retail website are technically the most vulnerable in 2011 as they were in 2010, they've seen a dramatic improvement down from 404 vulnerabilities in 2010.

This positive news is tempered by a remarkably high standard deviation. A high standard deviation indicates that a wide gap exists between the highly vulnerable websites, those that are not, and everything in between. Clearly though, there is simply no shortage of website vulnerabilities that are at risk of exploitation at any time. Media headlines over the last 18 to 24 months are a sobering reminder.

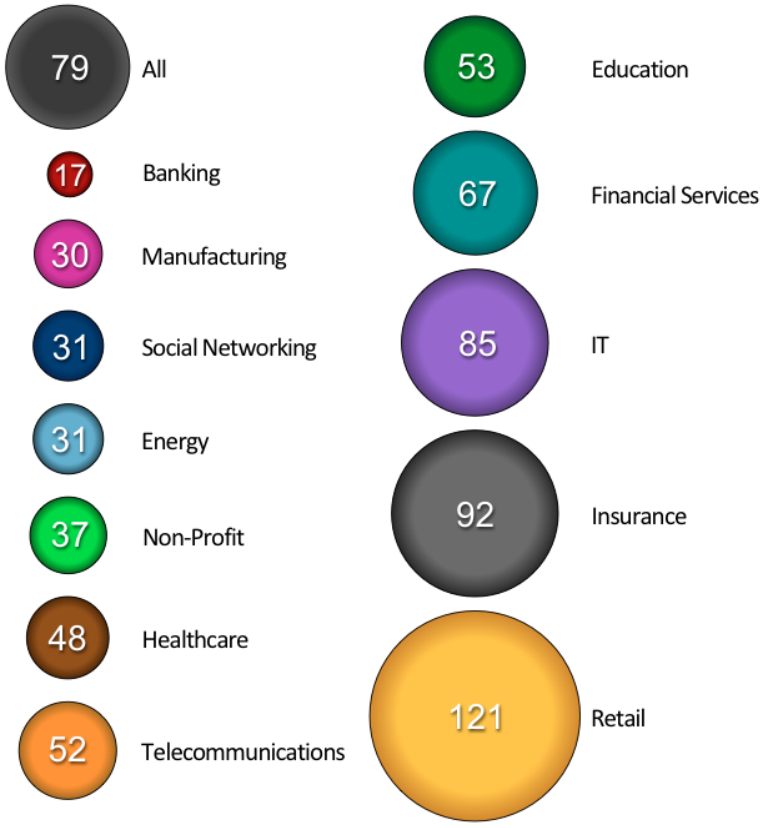


Figure 18. Average Number of Serious* Vulnerabilities (2011) (Sorted by industry)

INDUSTRIES COMPARED: VULNERABILITY TIME-TO-FIX

Once website vulnerabilities are identified, verified, and reported to customers by WhiteHat Sentinel, a certain amount of time transpires before the issue is resolved and confirmed as such. As no remedy can be instantaneous, it is important to measure the amount of time (Time-to-Fix), required to resolve certain vulnerabilities. Resolution could take the form of a software update, configuration change, Web application firewall rule, etc. Open vulnerabilities represent a window of opportunity for malicious hackers to exploit the website.

Custom Web application vulnerabilities cannot usually be resolved by deploying a patch from a third-party vendor. The IT security team must work with the organization's internal or third-party development team to update the code. As a result a negotiation must take place, and determine resource tradeoffs. The organization must decide to either allocate resources to produce a revenue-generating feature or use those resources to remediate an issue that may or may not be exploited. This is rarely an easy and clear-cut risk management decision.

The organization must decide to either allocate resources to produce a revenue-generating feature or use those resources to remediate an issue that may or may not be exploited.

By arranging the Time-to-Fix data in a cumulative fashion we get a sense of the time required to remediate vulnerabilities in a given percentage of an industry's websites. This view also helps answer a common question, "How fast should our organization fix vulnerabilities?" From a risk management perspective, if the organization is a target of opportunity, perhaps a goal of being at or above average is good enough. If however the organization is a target of choice, either ASAP or being among the fastest is more appropriate.

From Figure 19 we can see that average time to fix number is 38 days, much better than the 116 days in 2010. This means as of 2011, **on average 50% of organizations required 38 days or less to remediate their serious* vulnerabilities.** Time-to-Fix statistics across most industries showed solid improvement from 2010 to 2011.

In particular, Education websites improved significantly from about 100 days in 2010 to 30 days in 2011. The industries with the fastest time-to-fix were Energy (4 days), Manufacturing (17 days), and Retail (27 days). The slowest were Non-Profit (94 days), Financial Services (80 days), and Telecommunications (50 days) websites. One thing to notice is that the industries that improved greatly seem to have done so without addressing lagging remediation rates. This is because you can only measure Time-to-Fix when a vulnerability is actually fixed.

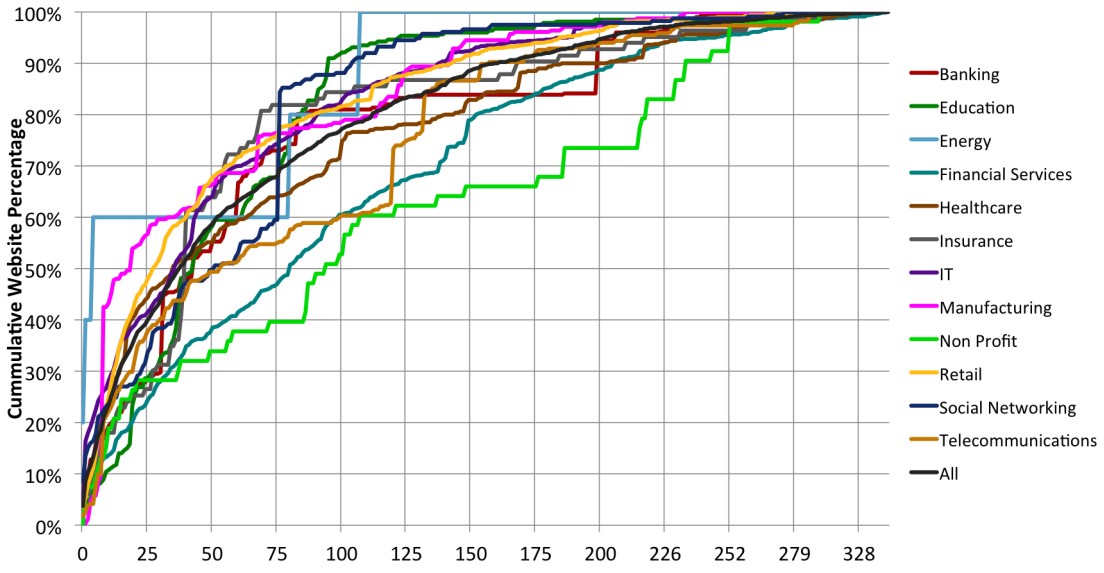


Figure 19. Aggregate Average Time-to-Fix for Serious* Vulnerabilities (Days, 2011) (Sorted by industry)

INDUSTRIES COMPARED: VULNERABILITY REMEDIATION RATE

Even if serious* vulnerabilities are identified, verified, and explained it does not necessarily mean they are fixed, quickly or at all. As such it is important to analyze the resolution rates of organizations that do get their vulnerabilities fixed, or not, and in what volumes (Figure 20). Some organizations target the easier issues first to demonstrate their progress in vulnerability reduction. Others prioritize the high severity issues to reduce overall risk.

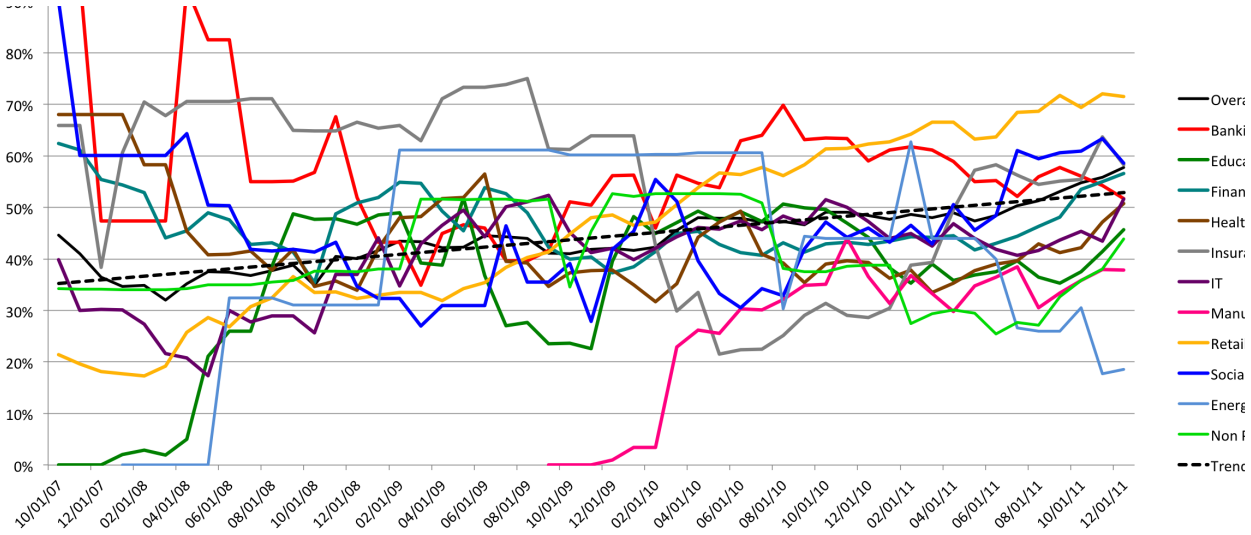


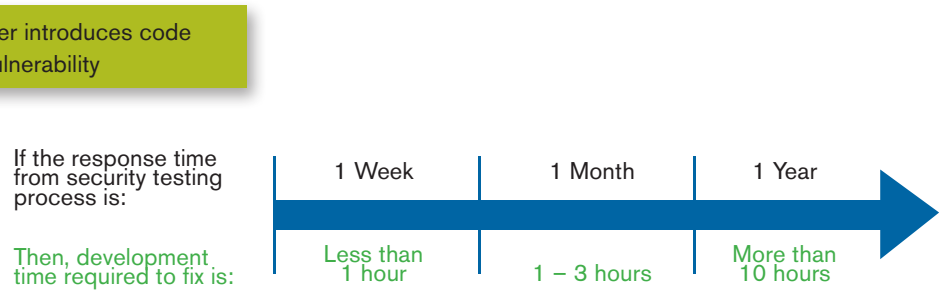
Figure 20. Historical trend of the percentage of reported vulnerabilities that have been resolved (Sorted by industry)

More good news, average Remediation Rates continue to climb upward. **The overall Remediation Rate in 2011 was 63%, up from 53% in 2010, and almost double the rate of 35% in 2007.** This represents a roughly 7% average improvement in the percentage of reported vulnerabilities that have been resolved during each of the last four years (2008, 2009, 2010, 2011).

The industries with the best Remediation Rates were Banking (74%), Telecommunications (69%), and Retail (66%) websites. On the opposite end of the spectrum were Energy (40%), Education (46%), and Manufacturing (50%) websites. Of course we must also take into account standard deviation, which is rather high, meaning there is a huge disparity between the websites where most vulnerabilities are fixed and those that are not.

Factors inhibiting organizations from remediating vulnerabilities:

- No one at the organization understands or is responsible for maintaining the code.
- No one at the organization knows about, understands, or respects the vulnerability.
- Feature enhancements are prioritized ahead of security fixes.
- Lack of budget to fix the issues.
- Affected code is owned by an unresponsive third-party vendor.
- Website will be decommissioned or replaced "soon." To note, we have experienced deprecated websites under the Sentinel Service still in active use for over two years.
- Risk of exploitation is accepted.
- Solution conflicts with business use case.
- Compliance does not require fixing the issue.



Relationship between the time that passes between testing for vulnerabilities and the time required to fix them.

INDUSTRIES COMPARED: WINDOW-OF-EXPOSURE

Websites are an ongoing business concern and security must be assured all the time, not just at a point in time. That's also why it must never be forgotten that an attacker only needs to exploit a single vulnerability, on any given day, to win. That's why the true Key Performance Indicator (KPI) of website security is Window-of-Exposure.

Websites are an ongoing business concern and security must be assured all the time, not just at a point in time.

Window-of-Exposure is the number of days in a year a website is exposed to at least one serious* vulnerability. As such, Window-of-Exposure is an informative combination of the total number of vulnerabilities, time-to-fix, and remediation rates -- taken over time. Any one of these metrics, or a combination thereof, may be the area that has the greatest impact on a given organization's Window-of-Exposure outcome. To provide context, let's consider two identical websites, SiteA and SiteB.

- *SiteA had 10 serious* vulnerabilities identified during the last year and 365 of those days it had at least one of those issues publicly exposed.*
- *SiteB had 100 serious* vulnerabilities identified during the last year and in 10 of those days it had at least one of those issues publicly exposed.*

Despite having 10 times the number of vulnerabilities, we would argue that during the last year SiteB had a substantially better security posture than SiteA as measured by the Window-of-Exposure.

It is revealing to see how various industries perform in the area of Window-of-Exposure. Figure 21 illustrates 2011 performance.

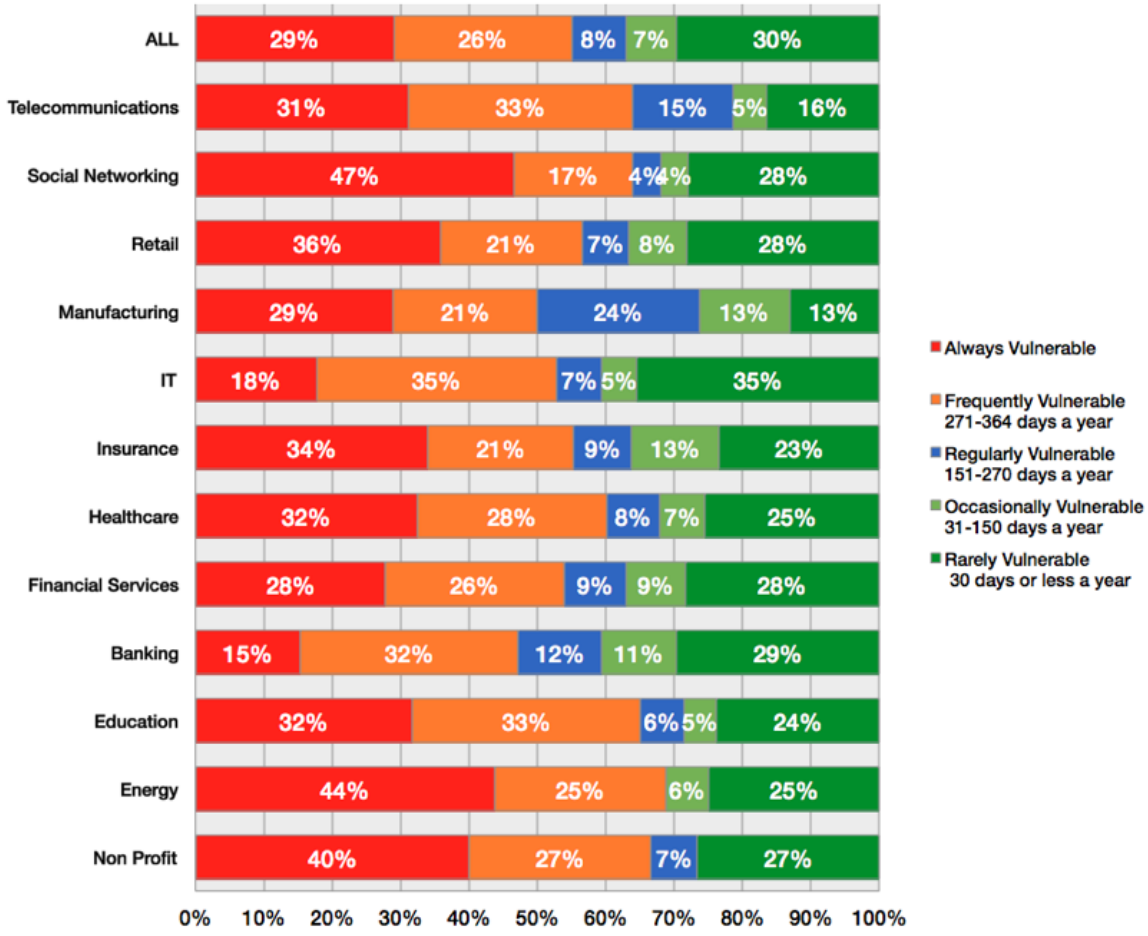


Figure 21. Overall Window of Exposure to Serious* Vulnerabilities (2011)
The percentage of websites that fall within a particular Window of Exposure zone
(Sorted by industry)

From 2010 to 2011 the overall average website Window-of-Exposure did improve, but only slightly from 233 to 231 days respectively.

The industry with the shortest average Window-of-Exposure was Banking at 185 days, but was oddly greatly increased from 74 days the year prior. IT websites also performed among the best at 208 days, two weeks better than 221 days in 2010. Insurance websites did comparatively well in 2011, averaging 211 days, an improvement from 236 days in 2010. The lagging industries were Non-Profit with an average Window-of-Exposure of 320 days, followed by Education at 261 days (up from 164) and Social Networking 264 days (up from 159).

Then as to be expected, the standard deviation of around 150 days across the board is something not to ignore. To better visualize what this means we put together Figure 22.

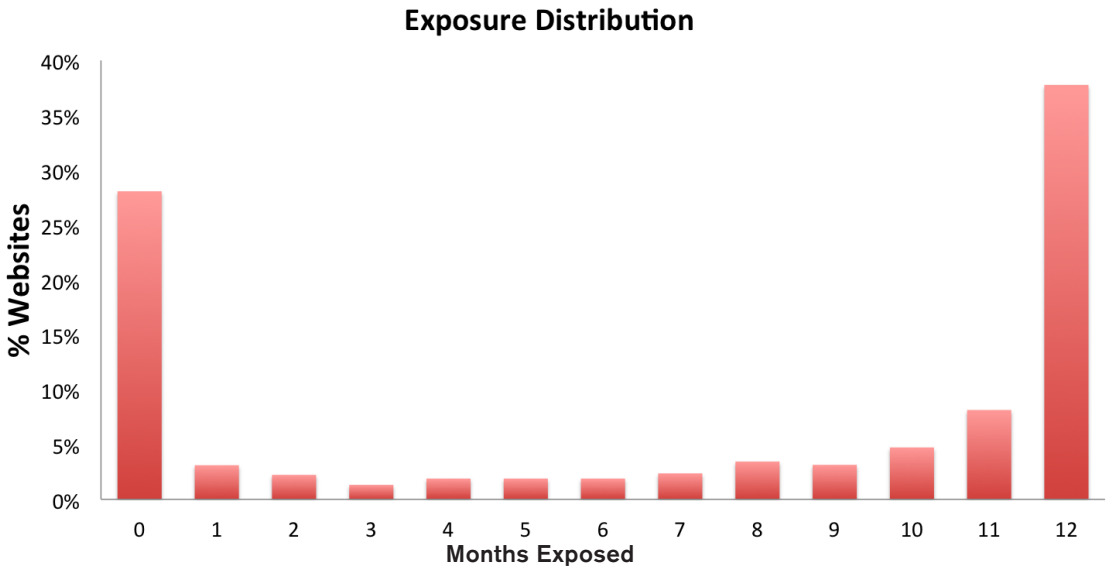


Figure 22. Window of Exposure Monthly Distribution (2011)

Figure 22 shows us that while vulnerabilities can and will happen to essentially every company, progress, and even exceptional performance and significantly reduced risk, is clearly possible when an organization focuses on closing the Window of Exposure. From our experience the difference comes down to how an organization allocates its resources.

For example, if a development team is consistently introducing large volumes of new vulnerabilities, it would be advisable to start by focusing on reducing the number of issues first (stop the bleeding). This may be achieved through an executive level mandate, better security controls in the development framework, awareness training, and security testing during QA.

On the other hand if the development team is generating comparably few vulnerabilities, but the issues remain remotely exploitable for long periods of time, then the greatest emphasis should be on improving the remediation processes. As a matter of policy, organizations may consider implementing remediation time mandates according to vulnerability severity. Window of Exposure data is a critical factor in making these strategic decisions.

COMPARING VULNERABILITY CLASSES

Different vulnerability classes are introduced into code, or a website, in a variety of different ways. Vulnerability classes also differ from each other by how difficult they are to fix. Some might take minutes, others hours, perhaps several days, and in at least some cases a complete application rewrite is necessary.

For example, SQL Injection is generally introduced by taking user-supplied data and funneling it into a concatenated database query. The first and best layer of defense against SQL Injection attacks is parameterized SQL statements. Cross-Site Scripting by contrast also requires user-supplied data, but the first and best defense is context-aware output encoding. Bonus points for adding input validation as an additional layer of protection to both vulnerability classes.

The first and best layer of defense against SQL Injection attacks is parameterized SQL statements.

Because of these differences, not just between SQL Injection and Cross-Site Scripting, but all the others as well, and the limited nature of development resources to fix issues, if a vulnerability class tends to be more difficult to fix than others, you might invest additional resources and effort to not have them in the first place. Or, you may at least gain of a better understanding of future remediation effort if you have these issues. By comparing the Time-to-Fix and the Remediation Rates of vulnerability classes we might be able to tease out new insights.

It is useful to understand the subtle, yet important, difference between Avg. Days to Close and Avg. Days Open in Figure 23.

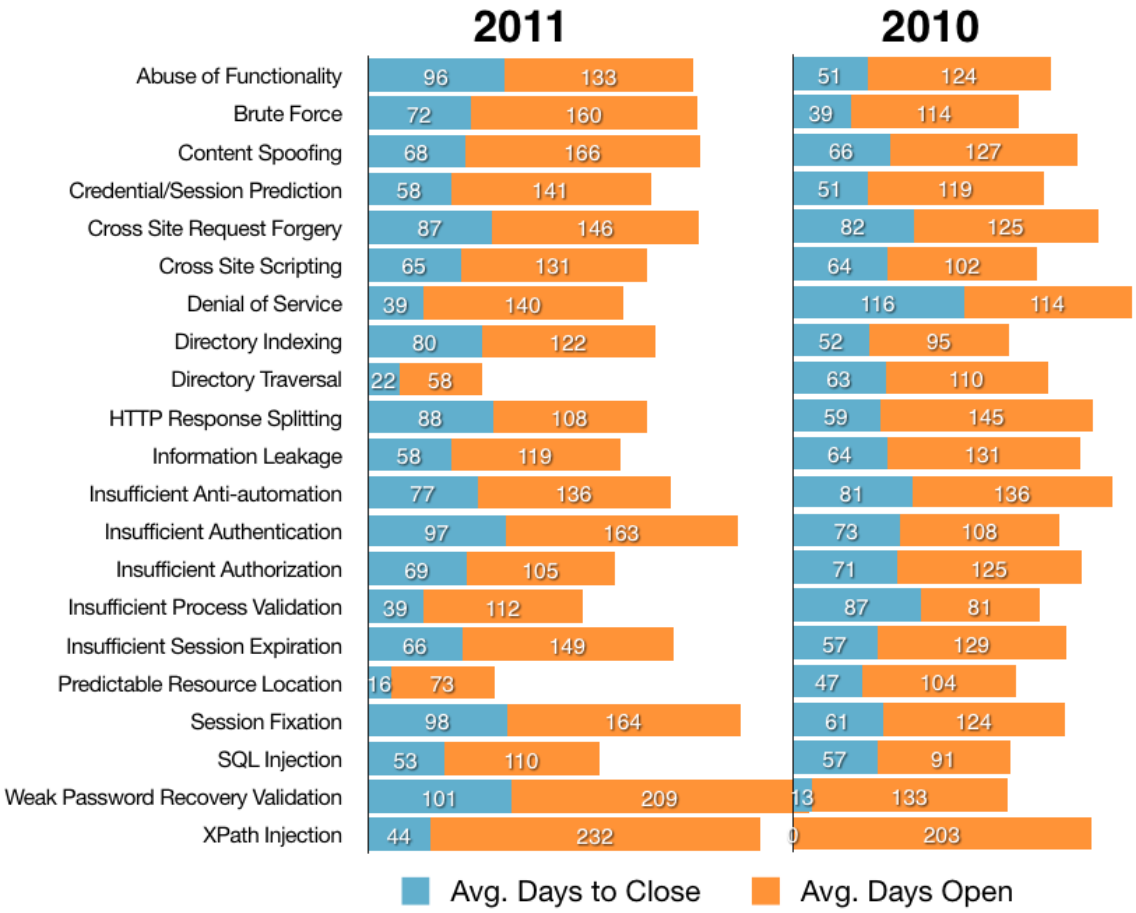


Figure 23. Average Days Opened and Days Exposed by Year

First of all, Avg. Days to Close is obviously only measurable when a vulnerability is actually fixed, which is not guaranteed. And because not all issues are fixed (aka closed), Avg. Days to Close may present a blurred view of the situation. This blur can be put into focus by measuring Avg. Days Open, which is greatly affected by a vulnerability class's Remediation Rate.

Avg. Days Open is the average number of days a vulnerability is open and publicly exposed in a particular calendar year. By showing both statistics, Avg. Days to Close and Avg. Days Open, we get a clearer understanding of what is actually happening.

In 2011, SQL Injection vulnerabilities closed in an average of 53 days, yet remained open for 110 days.

In 2011, SQL Injection vulnerabilities closed in an average of 53 days, yet remained open for 110 days. The delta between the two statistics is striking. Cross-Site Scripting closed in an average of 65 days and remained open for 131 days. Again, a rather noticeable gap due in large part to a low Remediation Rate.

All in all, whether looking at 2010 or 2011, most vulnerability classes are closed in no less than a few weeks on average, but most of the time it took two to three months. The majority of vulnerability classes remain open for at least three to five months. That's staggering when you think about it. Every single one of these issues is a verified and publicly exposed security vulnerability.

About half of the vulnerability classes in Figure 24 are improving their Remediation Rates year over year, while others are not. In either case, many of the improvements and declines are rather slight. For example, SQL Injection stayed basically flat at 55% in 2011, as did Cross-Site Scripting at 48%. So as it turns out, it is extremely difficult to derive meaning from any of the numbers in Figure 23 and Figure 24. While some vulnerability classes are no doubt more difficult to fix than others, the best answer is the rate of improvement is probably most affected by other external factors other than the vulnerability class itself.

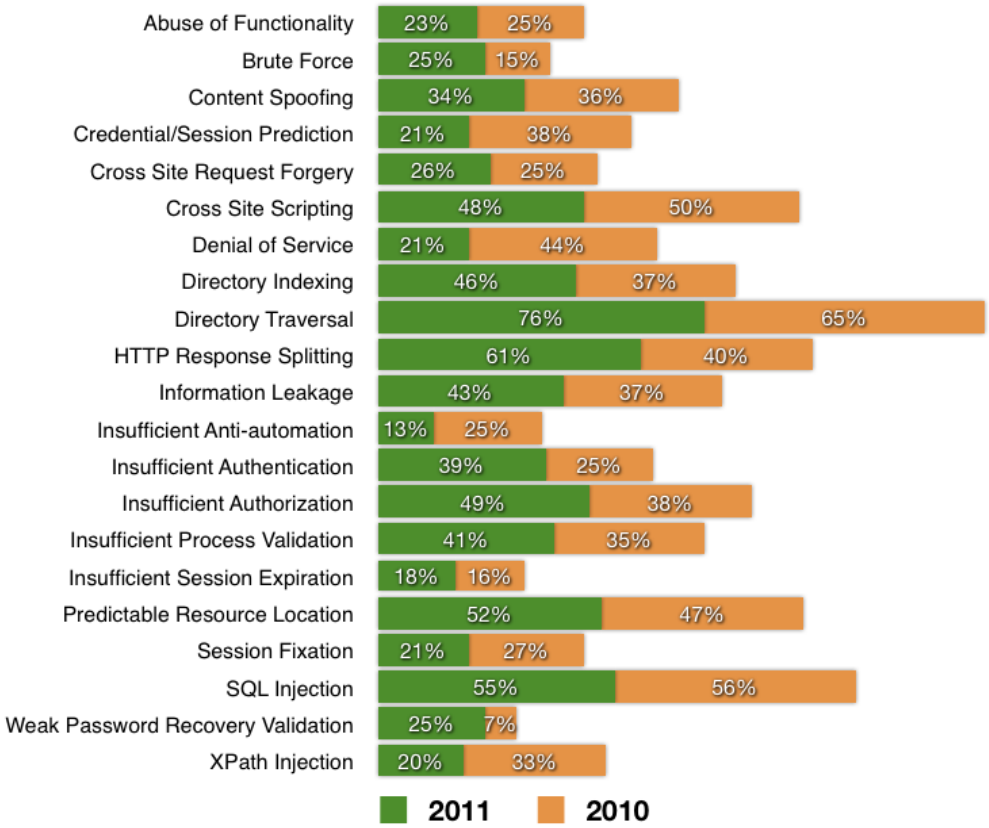


Figure 24. Remediation Rates by Year

VULNERABILITY REOPEN RATES

Website vulnerabilities may exist in application code, be caused by misconfiguration, be introduced via an intermediary device, or some combination thereof. As such, vulnerability remediation or mitigation can take many forms. These include a code update, change in system configuration, new permissions, Web Application Firewall rule, all the way down to outright disabling the vulnerable website or individual page.

When a customer "fixes" a vulnerability, how they do so is largely unknown to us because we're testing from an external point of view. The application security team at a particular organization may not even know precisely how a vulnerability was fixed, only that it was. From the customer perspective, WhiteHat Sentinel is viewed as an up to the minute dashboard of their current website security posture. "Am I currently vulnerable, or am I not?" A secondary concern may later be, "why am I not?".

Whenever a vulnerability is identify and verified, WhiteHat Sentinel constantly checks the status of the vulnerability as open or closed. If the issue is closed, which may happen for a variety of reasons, Sentinel will immediately mark it as such. If later Sentinel finds a vulnerability, which had been previously identified and reported, it will open it back up. This open and close, and reopen sequence may occur a great many times during a vulnerability's life span.

Figure 25 shows that 20% of vulnerabilities identified by WhiteHat Sentinel have been reopened as some point in time, often many times, while 80% have not. To many, a 20% reopen rate sounds like a sizable, almost unbelievably high, number – and for the most part we'd agree. To better understand this statistic we must consider the wide variety of reason why a vulnerability may "close" in the first place.

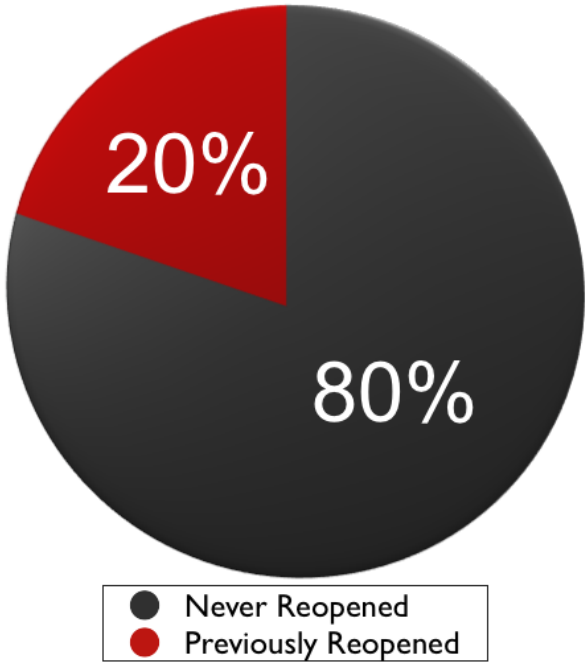


Figure 25. Overall Reopen Rate (2011)
Percentage of serious* vulnerabilities that have been reopened at least once

When and why a vulnerability may close and reopen:

1. WhiteHat Sentinel vulnerability checks, we refer to internally as "ammo," are constantly being improved with the very latest in filter-bypass techniques. If a customer did not fix a vulnerability "properly," a previously closed issue will re-open.
2. Some WhiteHat Sentinel customers specifically blacklist specific strings in WhiteHat Sentinel scans. If we update a string slightly, previously "closed" vulnerabilities will reopen. (i.e. change "<WHXSS>" to "<XSSWH>").
3. Secure code is overwritten with new vulnerable code.
4. New "safe" code is reverted to old "vulnerable" code.
5. The vulnerable URL is taken down, then subsequently and mistakenly, put back up.
6. A system configuration update, or roll back, caused previously closed vulnerabilities to re-open. For example, many Microsoft .Net developers rely upon native framework security protections and whose code would not be safe without it. Sometimes these protections are mistakenly or intentionally disabled causing vulnerable code to be exposed.
7. Inconsistencies across load balancers where one or more (application) servers in the rotation is still running old vulnerable code.
8. A rule in a Web Application Firewall or Intrusion Prevention System is removed or placed in "alert only" mode. It is also possible that the device has for some reason failed in an open state -- a common deployment model.
9. A vulnerability is only exposed to an authenticated user and our login credentials became invalidated.
10. The website has become inaccessible, either voluntarily disabled or our IP address is blocked by a perimeter firewall.

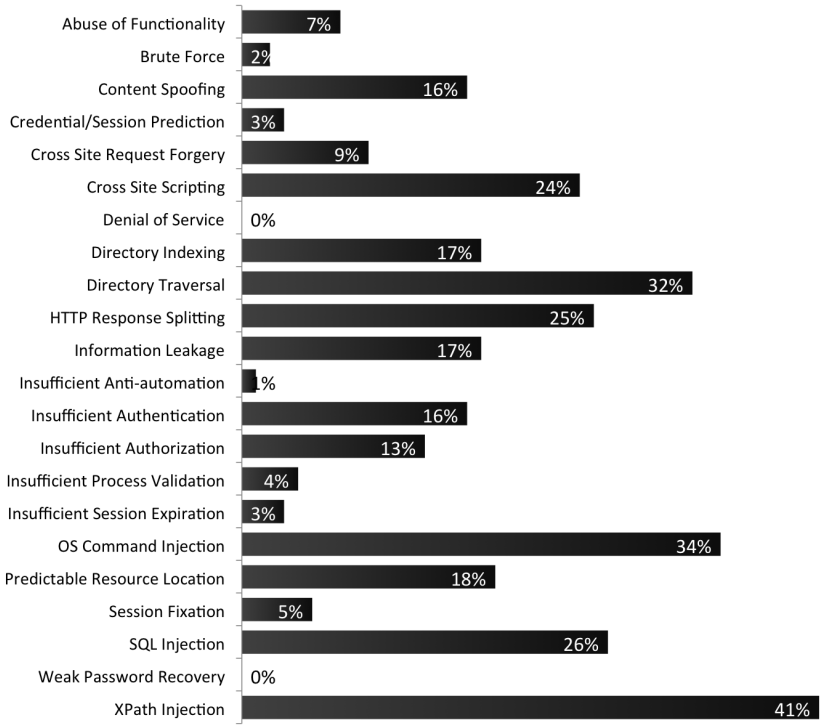


Figure 26. Reopen Rate by Vulnerability Class (2011)
Percentage of serious* vulnerabilities that have been reopened at least once

As you can see, there are a great number of reasons why a vulnerability might "close," but the underlying issue is not really fixed or fixed properly. This alone illustrates the challenges of operational website security. To gain additional insight to reopen rates we can look at it from a vulnerability class perspective. Are some vulnerability classes easier to close and keep closed than others?

In Figure 26, across the board we see wide variance in the percentage of vulnerabilities that have been reopened in specific classes. Notice that 41% of XPath Injection vulnerabilities have been re-opened as compared to 0% for Denial of Service and Weak Password Recovery. Given the threat landscape it is also alarming that about a quarter of SQL Injection vulnerabilities (26%) have been reopened, the same being true for Cross-Site Scripting (24%). The same problem arises with OS Command Injection-- 34% have been reopened at least once.

By looking at the chart as a whole, there may be something to learn that is extremely revealing. Vulnerability classes that tend to be exploited by injecting malicious data into URL parameters tend to reopen most often. These classes are Content Spoofing, Cross-Site Scripting, Directory Traversal, HTTP Response Splitting, OS Command Injection, SQL Injection, and XPath Injection. These classes are commonly referred to as technical vulnerabilities or syntax issues and are the same issues that scanners are most adept at identifying.

Vulnerability classes better classifiable as business logic flaws or semantic issues, when fixed, typically remain that way. These vulnerability classes are Abuse of Functionality, Brute Force, Credential/Session Prediction, Cross-Site Request Forgery, Insufficient Anti-Automation, Insufficient Authentication, Insufficient Authorization, Insufficient Process Validation, Insufficient Session Expiration, and Weak Password Recovery. Nearly all still have a reopen rate, but noticeably lower by comparison.

One might conclude that vulnerabilities where a development framework or central control system is responsible for security, rather than multiple individual developers all writing their own security controls, achieves better results. Perhaps if injection vulnerabilities were more often handled by centralized security controls, system reopen rates would decline.

Figure 27 shows something that is a cause for concern. The higher a vulnerability's severity, the higher the likelihood that a vulnerability will reopen. We must be careful here not to draw too many conclusions -- correlation does not mean causation. What we can do is share at least one possible explanation in which we have first hand experience.

When a potentially damaging vulnerability is identified, an organization will often go into panic mode and their standard software development process is broken. Their application security team, software developers, system engineers, and management race to understand the issues impact, create a (temporary) fix, QA it (maybe), release the change to production, and perform re-tests. In this mad dash, a vulnerability may not be fixed properly. Or, something else that's not uncommon, a production hot-fix may not get back-ported to development. A future release will then overwrite the hot-fix and cause the vulnerability to reopen. See #4 above in "When and why a vulnerability may close and reopen."

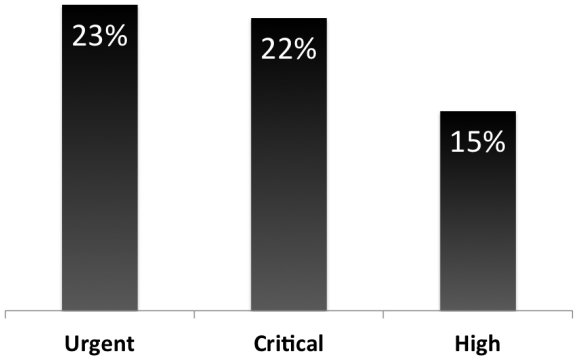


Figure 27. Reopen Rate by Severity (2011)
Percentage of serious* vulnerabilities that have been reopened at least once

RECOMMENDATIONS AND SUMMARY

Judging from the data and a decade of website and Web application security assessment experience, it is clear that many Software Development Lifecycle (SDL) security strategies are not universally effective, but situation-dependent. Whether referring to source code reviews, security testing during QA, Web application firewalls, and other methods, each has an appropriate time and place. In order to cost-effectively increase the organization security posture, the challenge is to determine what to recommend, what to implement, and what to purchase when.

For example, some SDLs produce comparatively few security vulnerabilities with each release, but when issues are identified, they are not fixed quickly or often. In this case, the organization may benefit more from improvements to their remediation process and an evaluation of resource allocation decisions, rather than sending developers to software security training. Failure to measure or understand where an SDL program is deficient, before taking action, is surefire way to waste time and money – both of which are usually extremely limited.

Failure to measure or understand where an SDL program is deficient, before taking action, is surefire way to waste time and money – both of which are usually extremely limited.

An effective step-by-step strategy to build out a website security program that yields results must include the following:

1. *Find your websites, all of them, and prioritize: Prioritization can be based upon business criticality, data sensitivity, revenue generation, traffic volume, number of users, or other criteria the organization deems important. Knowing what systems need to be defended and their value to the business provides a barometer for an acceptable security investment.*
2. *Measure your current security posture, from an attacker perspective: This step is not just about identifying vulnerabilities, it is about understanding what classes of adversaries need to be defended against and your exposure to them. Look at your security posture as a bad guy would.*
3. *Decide if each website is a likely 'target of opportunity' or 'target of choice.': This decision provides the basis for organizational security goals. Should the organization's security posture be on par with, or lead, relative to industry peers.*
4. *Trend and track the lifecycle of vulnerabilities: Is the SDL behind the website producing too many vulnerabilities?; Is the time required to fix issues lagging, simply not fixing enough of them, or some combination? The answer to these questions will serve as a guide for which new and/or improved SDL-related activities are likely to make the most impact and drive toward organizational goals.*

ADDENDUM

Industry	Number of Vulns	Std. Dev	Remediation Rate	Std. Dev	Window of Exposure (Days)
Overall	230	1652	53%	40%	233
Banking	30	54	71%	41%	74
Education	80	144	40%	36%	164
Financial Services	266	1935	41%	40%	184
Healthcare	33	87	48%	40%	133
Insurance	80	204	46%	37%	236
IT	111	313	50%	40%	221
Manufacturing	35	111	47%	40%	123
Retail	404	2275	66%	36%	328
Social Networking	71	116	47%	34%	159
Telecommunications	215	437	63%	40%	260

Figure 29. At a Glance: The Current State of Website Security (2010)
(Sorted by industry)
The average number of serious* vulnerabilities discovered per website, the percentage of reported vulnerabilities that have been resolved (Remediation Rate), and average the number of days a website is exposed to at least one serious* vulnerability (Window of Exposure).

DATA COLLECTION: WHITEHAT SENTINEL

WhiteHat Sentinel[6] is the most accurate, complete and cost-effective website vulnerability management solution available. It delivers the flexibility, simplicity and manageability that organizations need to control their website security and prevent Web attacks. WhiteHat Sentinel is built on a Software-as-a-Service (SaaS) platform that scales massively, supports the largest enterprises, and offers the most compelling business efficiencies to lower your overall cost for website security.

Unlike traditional website scanning software or consultants, only WhiteHat Sentinel combines proprietary scanning technology with custom testing by the industry's only Threat Research Center (TRC)[7]. The WhiteHat Security TRC is an 80+ team of website security experts who act as a critical and integral component of the WhiteHat Sentinel family. The TRC is as an extension of your website security team – actively managing business website risk posture so you can focus on technology and business goals.

DATA, PLATFORM, AND METHODOLOGY

The Data

- 7,000+ production and pre-production websites across 500+ organizations representing many of the world's most recognizable brands across 12 industries. (Banking, Education, Energy, Financial Services, Healthcare, Information Technology, Insurance, Manufacturing, Non-Profit, Retail, Social Networking, and Telecommunications)
- The websites WhiteHat Security's assesses generally represent the more "important" and "secure" websites on the Web, owned by organizations serious about website security.
- Each WhiteHat Sentinel customer self-selects the most appropriate industry label for each website under service.
- While WhiteHat Sentinel customers maintain complete control over their vulnerability assessment schedule. The majority of websites are assessed for vulnerabilities multiple times per month.
- The WhiteHat Sentinel platform has peaked at 1,700 simultaneous scans, collectively generating 940 million HTTP requests per month, consuming roughly 6 TB of data per day, and necessitating 50 Mbps of bandwidth 24x7.
- Vulnerabilities classified according to WASC Threat Classification[8]
- Severity naming convention aligns with PCI-DSS[9]

Important Factors Influencing the Data

1. Websites range from highly complex and interactive with large attack surfaces to static brochureware. Brochureware websites, because of their relatively limited attack surface, tend to have a lower number of "custom" Web application vulnerabilities.
2. The 500+ organizations are largely, but not exclusively US-based, as are their websites. However, there is a significant number of websites localized in a variety of languages.
3. Vulnerabilities are counted by unique Web application and vulnerability class. If three of the five parameters of a single Web application (/foo/webapp.cgi) are vulnerable to SQL Injection, this is counted as 3 individual vulnerabilities (e.g. attack vectors). Secondly, if a single parameter can be exploited in more than one way, each of those are counted as well. We count this way because a vulnerability in each parameter may actually lead to a different problem in a different part of the code.
4. Only serious* vulnerabilities that can be directly and remotely exploitable that may lead to data loss or account compromise are included.
5. "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 policy violation, it must be taken on a case-by-case basis.
6. Our vulnerability assessment processes are incremental and ongoing. The frequency of assessments, which is customer-driven, should not automatically be considered "complete."
7. It is best to view this report as a best-case scenario and there are always more vulnerabilities to be found. New attack techniques are constantly being researched to uncover previously unknown vulnerabilities, including in previously tested and unchanged code. Likewise assessments may be conducted in different forms of authenticated state (i.e. user, admin, etc.).
8. Websites may be covered by different WhiteHat Sentinel Service. Premium (PE), Standard (SE), Baseline (BE), PreLaunch (PL) offer varying degrees of testing criteria, but all include verification. PE covers all technical vulnerabilities and business logic flaws identified by the WASC Threat Classification (and some beyond). SE focuses primarily on the technical vulnerabilities. BE bundles critical technical security checks into a production-safe, fully-automated service.
9. 42% of the all vulnerabilities in this report were found after January 1, 2011.
10. There is some amount of double website / vulnerability counting. Customers sometimes deploy WhiteHat Sentinel simultaneously on production websites and their internal QA/Staging mirrors. As standard practice, our process make no assumption that any website is identical to another. It also is important to note that it is statistically rare for production and QA/Staging to have identical vulnerabilities.

*** Serious Vulnerabilities: Vulnerabilities with a HIGH, CRITICAL, or URGENT severity as defined by PCI-DSS naming conventions. Exploitation could lead to server breach, user account take-over, data loss, or compliance failure.**

The Platform

Built as a Software-as-a-Service (SaaS) technology platform, WhiteHat Sentinel combines proprietary scanning technology with analysis by security experts in its Threat Research Center, to enable customers to identify, prioritize, manage and remediate website vulnerabilities. WhiteHat Sentinel focuses solely on previously unknown vulnerabilities in custom Web applications– code unique to an organization (Figure 31). Every vulnerability discovered by any WhiteHat Sentinel Service is verified for accuracy and prioritized by severity and threat.

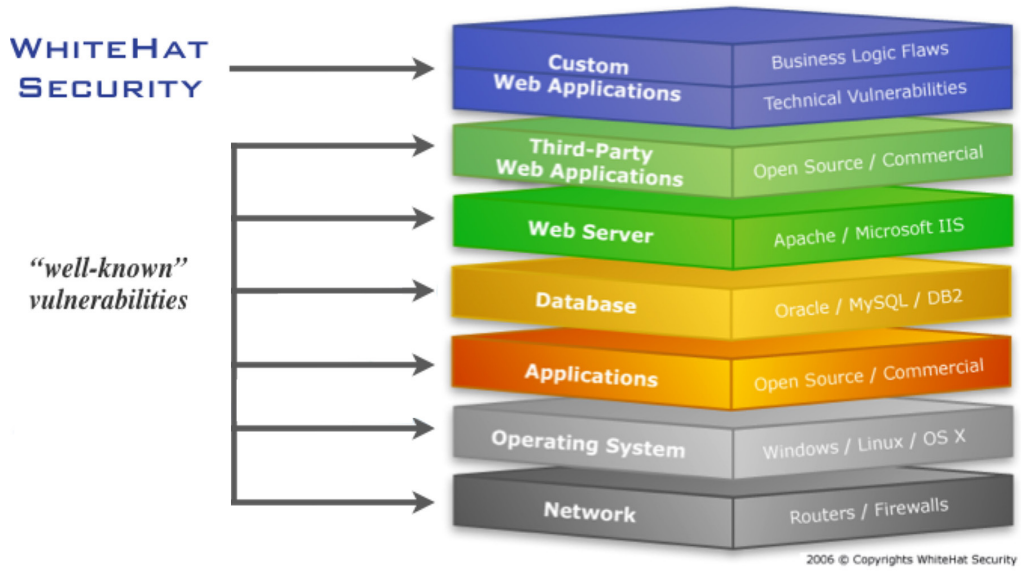


Figure 31. Software / Vulnerability Stack

The Methodology

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[10] (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.

WhiteHat Sentinel offers four different levels of service (Premium, Standard, Baseline, and PreLaunch[11]) to match the level of security assurance required by the organization. Additionally, WhiteHat Sentinel exceeds PCI 6.6 and 11.3.2 requirements for Web application scanning[12].

Scanning Technology

- *Production Safe (PE, SE, BE): Non-invasive testing with less performance impact than a single user.*
- *False-positives: Every vulnerability is verified for accuracy by WhiteHat Security's Threat Research Center.*
- *Web 2.0 Support: JavaScript, Flash, AJAX, Java Applets, and ActiveX are handled seamlessly.*
- *Authenticated Scans: Patented automated login and session-state management for complete website coverage.*
- *Business Logic: customized tests analyze every form, business process, and authentication / authorization component.*

SELECTIION GUIDELINES FOR PRODUCTION WEBSITES			
Type of Service	**Sentinel PE**	**Sentinel SE**	**Sentinel BE / BE Enterprise**
Brief Description of Service	Sentinel PE is ideal for websites that are permanent, mission-critical, and governed by compliance requirements (think transactional, forms-based capabilities). It offers business logic testing as well as all testing offered by SE.	Sentinel PE is ideal for websites that are permanent, mission-critical, and governed by compliance requirements (think transactional, forms-based capabilities). It offers business logic testing as well as all testing offered by SE.	BE - The foundational solution for covering all your website assets or for protecting basic, less-critical websites. BE Enterprise - The foundational solution for covering all your website assets or for protecting basic, less-critical sites, designed to be a massively scalable "best value" solution designed to fit any environment.
Website Type(s)	Permanent, mission-critical websites requiring both technical and business logic vulnerability testing, using multi-step form-based processes and authentication/login. Geared for production environments with no impact on performance (production-safe).	Permanent websites that are not necessarily mission-critical, but using some forms and/or login. Geared for production environments with no impact on performance (production-safe).	Seasonal or temporary websites with limited or shallow use of forms/login. Geared for production environments with no impact on performance (production-safe).
Competitive Set	Consultants or internal website security experts.	Scanning tool that provides verified results without overhead.	Scanning tool that searches for technical vulnerabilities.
Management	WhiteHat TRC handles the initiation, configuration and tuning.	WhiteHat TRC handles the initiation, configuration and tuning.	N/A
Price Sensitivity	Cost is less a factor.	Cost is less a factor.	Cost is the main factor vs. decreasing headcount .
Threat Type	Fully Targeted attacks focused on specific websites using repeated and systematic attacks	Directed Opportunist – Scans far and wide looking for easy opportunities to exploit.	Random Opportunist – Non-targeted attacks such as script kiddies and worms.

Figure 32. WhiteHat Sentinel Selection Guidelines

REFERENCES

1 Netcraft: June 2012 Web Server Survey
http://news.netcraft.com/archives/2012/06/06/june-2012-web-server-survey.html

2 CSRF, the sleeping giant
http://jeremiahgrossman.blogspot.com/2006/09/csrf-sleeping-giant.html

3 WhiteHat Security's Approach to Detecting Cross-Site Request Forgery (CSRF)
https://blog.whitehatsec.com/whitehat-security%E2%80%99s-approach-to-detecting-cross-site-request-forgery-csrf/

4 LizaMoon
http://en.wikipedia.org/wiki/LizaMoon

5 'LizaMoon' Mass SQL Injection Attack Escalates Out of Control
http://www.eweek.com/c/a/Security/LizaMoon-Mass-SQL-Injection-Attack-Escalates-Out-of-Control-378108/

6 WhiteHat Sentinel Service
https://www.whitehatsec.com/sentinel_services/sentinel_services.html

7 WhiteHat Security Threat Research Center (TRC)
https://www.whitehatsec.com/sentinel_services/threat_research.html

8 WASC Threat Classification (v2)
http://projects.webappsec.org/Threat-Classification

9 PCI Data Security Standard (PCI DSS)
https://www.pcisecuritystandards.org/security_standards/pci_dss.shtml

10 PCI Data Security Standard
https://www.pcisecuritystandards.org/

11 WhiteHat Sentinel Selection Guide
https://www.whitehatsec.com/sentinel_services/selection.html

12 Achieving PCI Compliance with WhiteHat Sentinel
http://www.whitehatsec.com/home/services/pci.html

*** Serious Vulnerabilities: Vulnerabilities with a HIGH, CRITICAL, or URGENT severity as defined by PCI-DSS naming conventions. Exploitation could lead to server breach, user account take-over, data loss, or compliance failure.**

Jeremiah Grossman is the Founder and Chief Technology Officer of WhiteHat Security, where he is responsible for Web security R&D and industry outreach. Over the last decade, Mr. Grossman has written dozens of articles, white papers, and is a published auth or. His work has been featured in the Wall Street Journal, Forbes, NY Times and hundreds of other media outlets around the world.

As a well-known security expert and industry veteran, Mr. Grossman has been a guest speaker on six continents at hundreds of events including TED, BlackHat Briefings, RSA, SANS, and others. He has been invited to guest lecture at top universities such as UC Berkeley, Stanford, Harvard, UoW Madison, and UCLA. Mr. Grossman is also a co-founder of the Web Application Security Consortium (WASC) and previously named one of InfoWorld's Top 25 CTOs.

He serves on the advisory board of two hot start-ups, Risk I/O and SD Elements, and is a Brazilian Jiu-Jitsu Black Belt. Before founding WhiteHat, Mr. Grossman was an information security officer at Yahoo!

ABOUT WHITEHAT SECURITY

Founded in 2001 and headquartered in Santa Clara, California, WhiteHat Security provides end-to-end solutions for Web security. The company's cloud technology platform and leading security engineers turn verified security intelligence into actionable insights for customers. Through a combination of core products and strategic partnerships, WhiteHat Security provides complete Web security at a scale unmatched in the industry. WhiteHat Sentinel, the company's flagship product line, currently manages thousands of websites - including sites in the most regulated industries such as top e-commerce, finance and healthcare companies.

WhiteHat SECURITY
WhiteHat Security, Inc. | 3970 Freedom Circle | Santa Clara, CA 95054
408.343.8300 | www.whitehatsec.com

062712