

Website Security Statistics Report 2015

About This Report

WhiteHat Security'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 in order to conduct business online safely.

Website security is an ever-moving target. New website launches are common, new code is released constantly, new web technologies are created and adopted every day; as a result, new attack techniques are frequently disclosed that can put every online business at risk. In order to stay protected, enterprises must receive timely information about how they can most efficiently defend their websites, gain visibility into the performance of their security programs, and learn how they compare with their industry peers. Obtaining these insights is crucial in order to stay ahead and truly improve enterprise website security.

To help, WhiteHat Security has been publishing its Website Security Statistics Report since 2006. This report is the only one that focuses exclusively on unknown vulnerabilities in custom web applications, code that is unique to an organization, and found in real-world websites. The underlying data is hundreds of terabytes in size, comprises vulnerability assessment results from tens of thousands of websites across hundreds of the most well-known organizations, and collectively represents the largest and most accurate picture of website security available. Inside this report is information about the most prevalent vulnerabilities, how many get fixed, how long the fixes can take on average, and how every application security program may measurably improve. The report is organized by industry, and is accompanied by WhiteHat Security's expert analysis and recommendations.

Contents

About This Report	2
Executive Summary	3
Vulnerability Likelihood	6
Window of Exposure	8
Survey Analysis	10
Average Number of Open Vulnerabilities	25
Average Days Open	26
Remediation Rates	27
Data Set & Methodology	28
Conclusion & Recommendations	29
Definitions	30

Executive Summary

More *secure* software, NOT more *security* software.

Unfortunately and unsurprisingly, website breaches have become an everyday occurrence. In fact, hacked websites have become so common that typically only the biggest data breaches capture enough attention to make headlines. The rest get to suffer quietly away from the public eye. Experts have known this eventuality was coming and honestly, the prediction was easy. All one had to do was to look at the pervasiveness of web use in modern society, the amount of data and dollars being exchanged online, and read any industry report about the volume of vulnerabilities exposed on the average website. With this information in hand, the final ingredient that ultimately leads to a breach is a motivated adversary willing to take advantage of the vulnerability, and as headlines tell us, there are plenty of motivated adversaries. Verizon's 2015 Data Breach Investigations Report¹ says for the financial services industry, web applications are the second-leading cause of incidents — just behind crimeware. Further, for healthcare and information technology industries, web applications are fourth and second respectively, when it comes to breach.

To this point, what no one could really predict or quantify were the possible consequences of having no website security measures in place at all. Now, after countless breaches on record, we have a fairly good idea. Website breaches lead directly to fraud, identity theft, regulatory fines, brand damage, lawsuits, downtime, malware propagation, and loss of customers. While a victimized organization may ultimately survive a cyber-crime incident, and fortunately most do, the business disruption and losses are often severe. Recent studies by the Ponemon Institute state that 45% of breaches exceed \$500,000 in losses². In the largest of incidents, many Fortune-listed companies have given shareholder guidance that the losses would range from tens of millions to hundreds of millions of dollars. Obviously, it is far preferable to do something proactive to avert and minimize harm before becoming the next headline.

The answer to web security, and much of information security, is we need more *secure* software, NOT more *security* software. While this is easy to say and has been said by us many times in

the past, the process of actually doing so is anything but solved or widely agreed upon — despite the plethora of so-called best-practices and maturity models. For example, we would all like to say, organizations that provide software security training for their developers experience fewer serious vulnerabilities annually than those who do not provide training. Or, organizations that perform application security testing prior to each major production release not only have fewer vulnerabilities year-over-year, but exhibit a faster time-to-fix. Broadly, these statements cannot be made authoritatively as the supporting data is sparse or nonexistent. At WhiteHat, and in this report, we're changing that.

For this report we utilized a version of BSIMM³ (Building Security In Maturity Model), called vBSIMM⁴ (the 'v' stands for 'vendor'). Think of vBSIMM as a lite version of BSIMM, a software security activity checklist you ask third-party software suppliers to fill out so you get a better idea of what effort they put into it. We modified the vBSIMM checklist slightly for our purposes, added some dates and activity frequency questions, and issued it as a survey to WhiteHat Security customers. We then looked at the aggregated responses of the survey (118 in total) and compared those results to WhiteHat Sentinel vulnerability metrics and mapped those to vBSIMM software security activities and to outcomes. Simple right? No, not really. As you'll see further down, the results were fascinating.

Before getting to the hard numerical statistics, we feel it's important to share what the data is signaling to us at a high level.

- We see no evidence of 'best-practices' in application security. At least, we see no practice likely to benefit every organization that implements them in any given scenario or application security metric. What we found is that certain software security activities (for example static analysis, architectural analysis, operational monitoring, etc.) would help certain application security metrics, but have little-to-no impact on others. For example, an activity might reduce the average number of vulnerabilities in a given application, not improve the speed of which vulnerabilities are fixed or how often. The best advice

1 Verizon 2015 Data Breach Investigations Report (DBIR)
<http://www.verizonenterprise.com/DBIR/>

2 Ponemon: The Post Breach Boom
<http://www.ponemon.org/blog/the-post-breach-boom>

3 The Building Security In Maturity Model (BSIMM)
<https://www.bsimm.com/>

4 BSIMM for vendors (vBSIMM)
<https://www.bsimm.com/related/>

we can give is for an organization to create a metrics program that tracks the area they want to improve upon, and then identify activities that'll most likely move the needle. If an activity does work – great! Keep doing it! If there is no measurable benefit, stop, save the time and energy, and try something else. Frankly, this process is much easier and more effective than blindly following maturity models.

- Another thing we noticed was that over the course of 2014, we saw a lot of high-profile infrastructure vulnerabilities such as Heartbleed⁵, Shellshock⁶, and more. These issues were remotely exploitable, highly dangerous, and pervasive. Some theorized that if we included these types of vulnerabilities into our research alongside our usual custom web application vulnerabilities, it would throw off our analysis. For example, you cannot blame Heartbleed on the software development group as it's the responsibility of IT infrastructure to protect against such an attack and developers were concerned their numbers would be unfairly dragged down. Fair enough. After doing the analysis, we found that including infrastructure vulnerability data actually improved the overall metrics. It seems the IT guys are overall faster and more consistent with patching. Imagine that!
- And finally, we had another industry shift over previous reports. When we asked customers the primary driver for resolving website vulnerabilities, 35% said risk reduction, which beat out compliance by more than 20 points. During our May 2013 report, compliance was the number one driver. We can only speculate on what's changed organizationally, but the leading theory is that most organizations that are required to be compliant with industry regulations have become so... yet the hacks keep happening. To keep hacks from happening, it appears risk reduction has taken center stage – and not a moment too soon.

With these larger themes out of the way, let's look at a few more interesting results:

- Organizations that are compliance-driven to remediate vulnerabilities have the lowest average number of vulnerabilities (12 per website) and the highest remediation rate (86%). Conversely and curiously, organizations driven by risk reduction

to remediate have an average of 23 vulnerabilities per website and a remediation rate of 18%. The skeptical theory is compliance-driven programs are simply incentivized to look only for the vulnerabilities which they are legally required to look for, which is obviously less than the totality. To summarize, if you look for fewer vulnerabilities you will find less. At the same time, compliance is a big corporate stick when it comes to remediating known issues and is likely what drives remediation rates upward. Risk reduction, right or wrong, often finds itself in an accepted business risk and risk tolerance discussion and ultimately drives remediation rates downward. However, risk reduction exhibits the best average time-to-fix at 115 days. The assumption is that if you are using a risk scale you are going after a smaller total pile of vulnerabilities and will therefore close them faster. Compliance on the other hand, with an average of 158 days time-to-fix, organizations believe they can afford to wait to fix vulnerabilities just before the auditor comes back around next year.

- Statistically, the best way to lower the average number of vulnerabilities, speed up time-to-fix, and increase remediation rates is to feed vulnerability results back to development through established bug tracking or mitigation channels. Doing so makes application security front and center in a development group's daily work activity and creates an effective process to solve problems. For organizations that have made the vulnerability feed to development process connection, they exhibit roughly 45% fewer vulnerabilities, fixed issues nearly a month faster on average, and increased remediation rates by 13 points.
- Organizations performing automated static code analysis saw a progressively improved average vulnerability time-to-fix as the activity frequency increased. For organizations who do not employ static code analysis, their time-to-fix was 157 days on average, for those at each major software release it was 138 days, and 96 days for those performing daily. These results are most likely due to the nature of static analysis taking place as code is being written and is fresh in the developer's mind.
- Utilizing a top N list of most important vulnerabilities looks to be a solid way to improve time-to-fix and remediation rates, but interestingly doesn't do very much to affect the average number of vulnerabilities. Organizations using top N lists see a two-month improvement in their time-to-fix vulnerabilities (from 300 to 243 days) and a seven-point increase in remediation rates (from 39% to 46%).

5 Heartbleed vulnerability
<http://heartbleed.com/>

6 Shellshock (software bug)
http://en.wikipedia.org/wiki/Shellshock_%28software_bug%29

- An activity that seems to have a dramatic positive effect on the average number of vulnerabilities is ad hoc code reviews of high risk applications. We found that organizations that never do ad hoc code reviews see an average of 35 vulnerabilities per website, while those who perform the activity with each major release see only 10, which amounts to a 71% decrease! There also seems to be a notable improvement in time-to-fix and remediation rates, making this activity closest to a best practice.
- Frequency of QA feedback of security reviews seems to have no strong correlation to any data points, which is interesting as common sense would tell you that this would have similar data points to frequency of static analysis as it is a small feedback loop. We would venture a guess that this is due to poor communication lines between QA, development, and security teams as they are speaking different languages.

In coordinating the research for this report, we have found that there is good news. For the vast majority of website vulnerabilities that are identified and exploited, we essentially know everything there is to know about them. We know how to prevent them, find them, and fix them. So you might ask: 'why are we still having problems with them?' The answer is two-fold: legacy and new code.

Legacy code. There are mountains of legacy code in existence, even mission-critical code, which is riddled with vulnerabilities waiting to be exploited. This software must be cleaned up and that effort is going to take a while. There is no way around that, but at least we know how. The rest is just going to take a lot of hard work and dedication.

New code. We now have more new code going into production than ever. Today's new code must be more secure than yesterday's code. With the right processes and measurement, it will never be perfect, but it *can* be done and it can significantly reduce the likelihood of a breach. When it's all said and done, once an organization really decides to improve upon application security, the answers are there – and many of those answers are in these pages.

Vulnerability Likelihood

Application vulnerability likelihood has significantly changed in the last few years. In 2012, an application was most likely to have Information Leakage (with 58% likelihood), or Cross-site Scripting (with 55% likelihood) vulnerabilities. However, in 2014, applications are most likely to have Insufficient Transport Layer Protection (with 70% likelihood) or Information Leakage (with 56% likelihood).

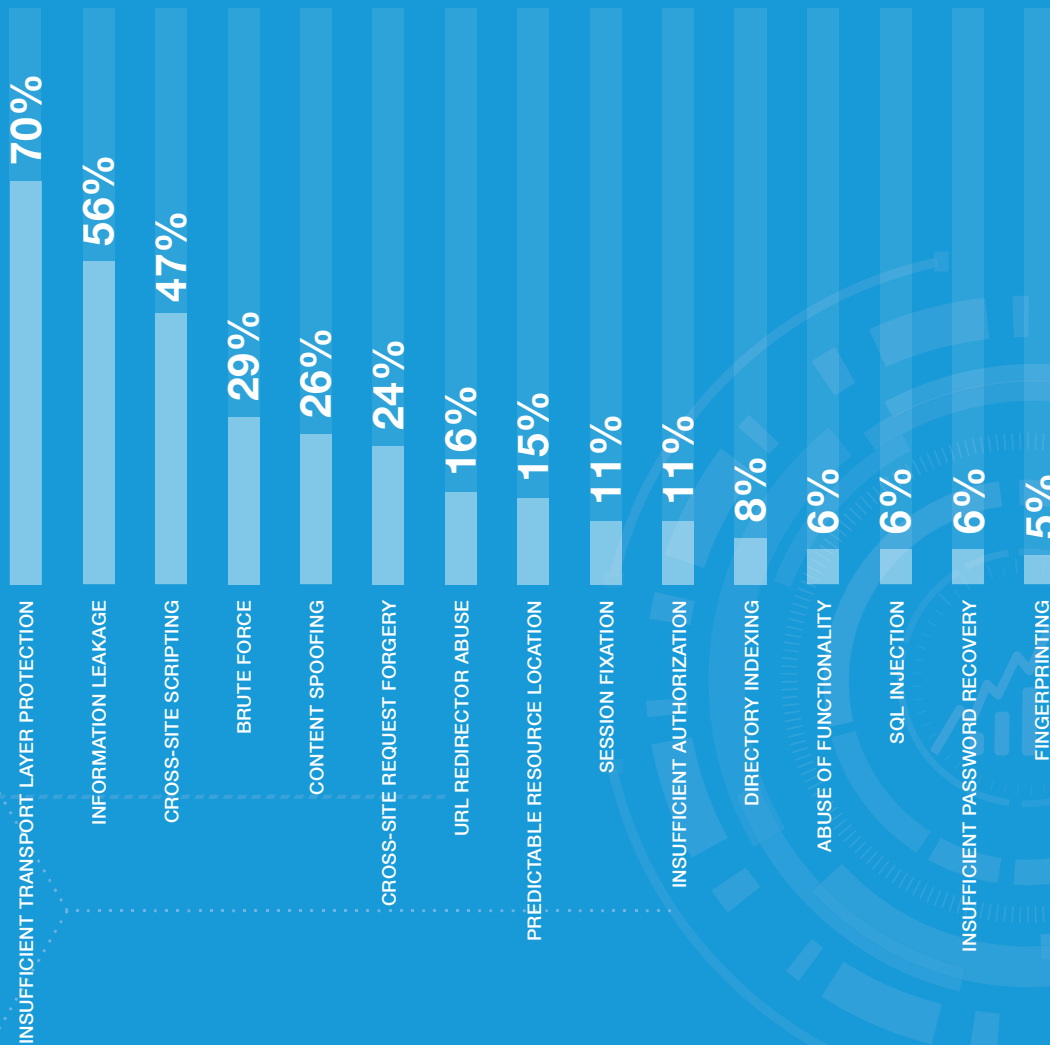
The sharp rise in the likelihood of Insufficient Transport Layer Protection can be explained by discovery of zero-day vulnerabilities such as Heartbleed and the new tests added as a result of that.

Likelihood of Content Spoofing, Cross-site Scripting and Fingerprinting has sharply declined in recent years. Content Spoofing was 33% likely in 2012, but only 26% in 2014. Likelihood of Fingerprinting vulnerabilities has dropped from 23% in 2012 to 5% in 2014. Cross-site Scripting has significantly declined as well (from 53% in 2012 to 47% in 2014).

Insufficient Transport Layer Protection, Information Leakage and Cross-Site Scripting are the most likely vulnerabilities in applications.

- Likelihood of Insufficient Transport Layer Protection: 70%
- Likelihood of Information Leakage: 56%
- Likelihood of Cross-site Scripting: 47%

Vulnerability Likelihood



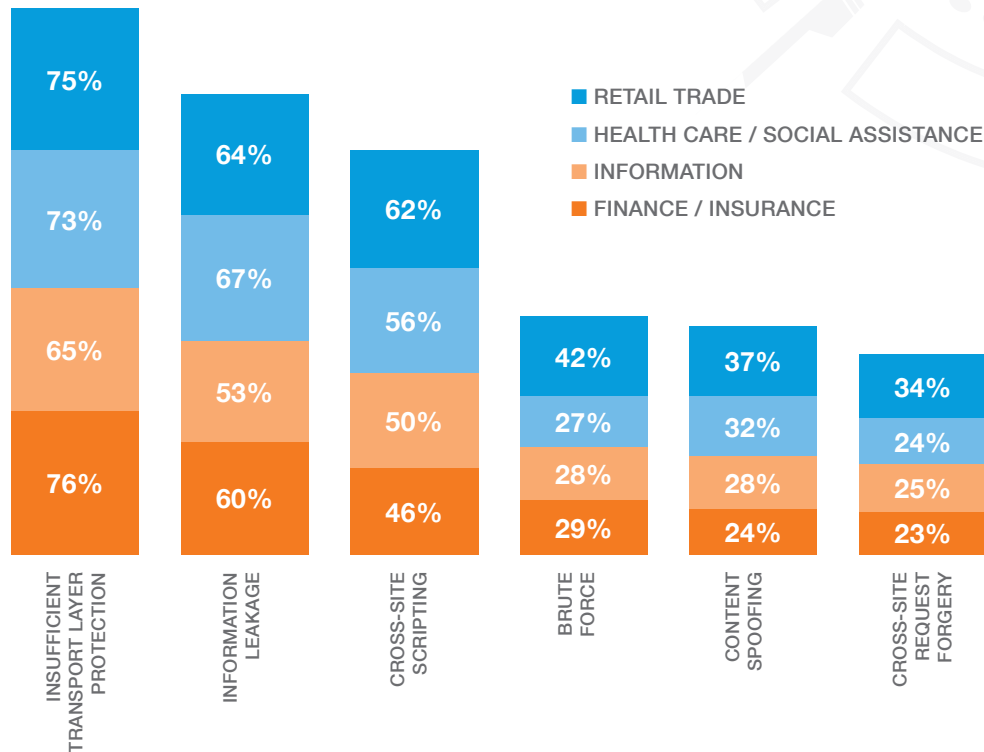
Likelihood of Insufficient Transport Layer Protection has sharply gone up in recent years (from 0% in 2010 to 70% likelihood in 2014).

Insufficient Transport Layer Protection and Information Leakage are the two most likely vulnerabilities in Retail Trade, Health Care / Social Assistance, Information, and Finance/Insurance sites.

Various industries (Retail Trade, Health Care / Social Assistance, Information, and Finance / Insurance) show similar patterns of likelihood for commonly found vulnerability classes.

The pattern of vulnerability likelihood remains unchanged across industries, as shown in the graph below.

Vulnerability Likelihood by Industry



Window of Exposure

Window of exposure is defined as the number of days an application has one or more serious vulnerabilities open during a given time period. We categorize window of exposure as:

Always Vulnerable: A site falls in this category if it is vulnerable on every single day of the year.

Frequently Vulnerable: A site is called frequently vulnerable if it is vulnerable for 271-364 days a year.

Regularly Vulnerable: A regularly vulnerable site is vulnerable for 151-270 days a year.

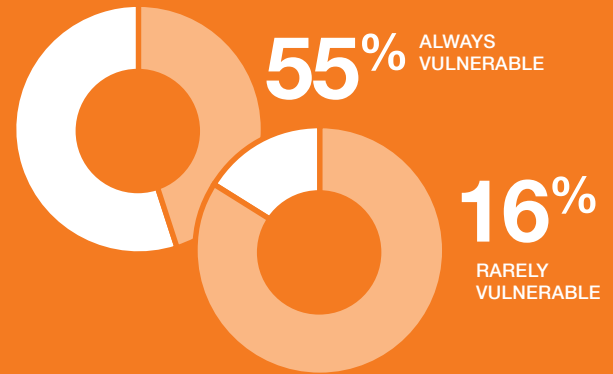
Occasionally Vulnerable: An occasionally vulnerable application is vulnerable for 31-150 days a year.

Rarely Vulnerable: A rarely vulnerable application is vulnerable for less than 30 days a year.

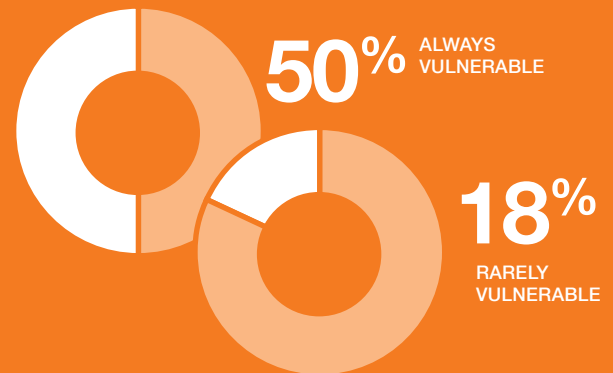
Our analysis shows that 55% of the Retail Trade sites, 50% of Health Care / Social Assistance sites, and 35% of Finance / Insurance sites are **always vulnerable**. Similarly, only 16% of the Retail Trade sites, 18% of Health Care / Social Assistance sites, and 25% of Finance / Insurance sites are **rarely vulnerable**.

Conversely, Educational Services is the best performing industry with the highest percentage of rarely vulnerable sites (40%). Arts, Entertainment, and Recreation is the next best industry with 39% of sites in rarely vulnerable category.

Retail Trade



Health Care / Social Assistance

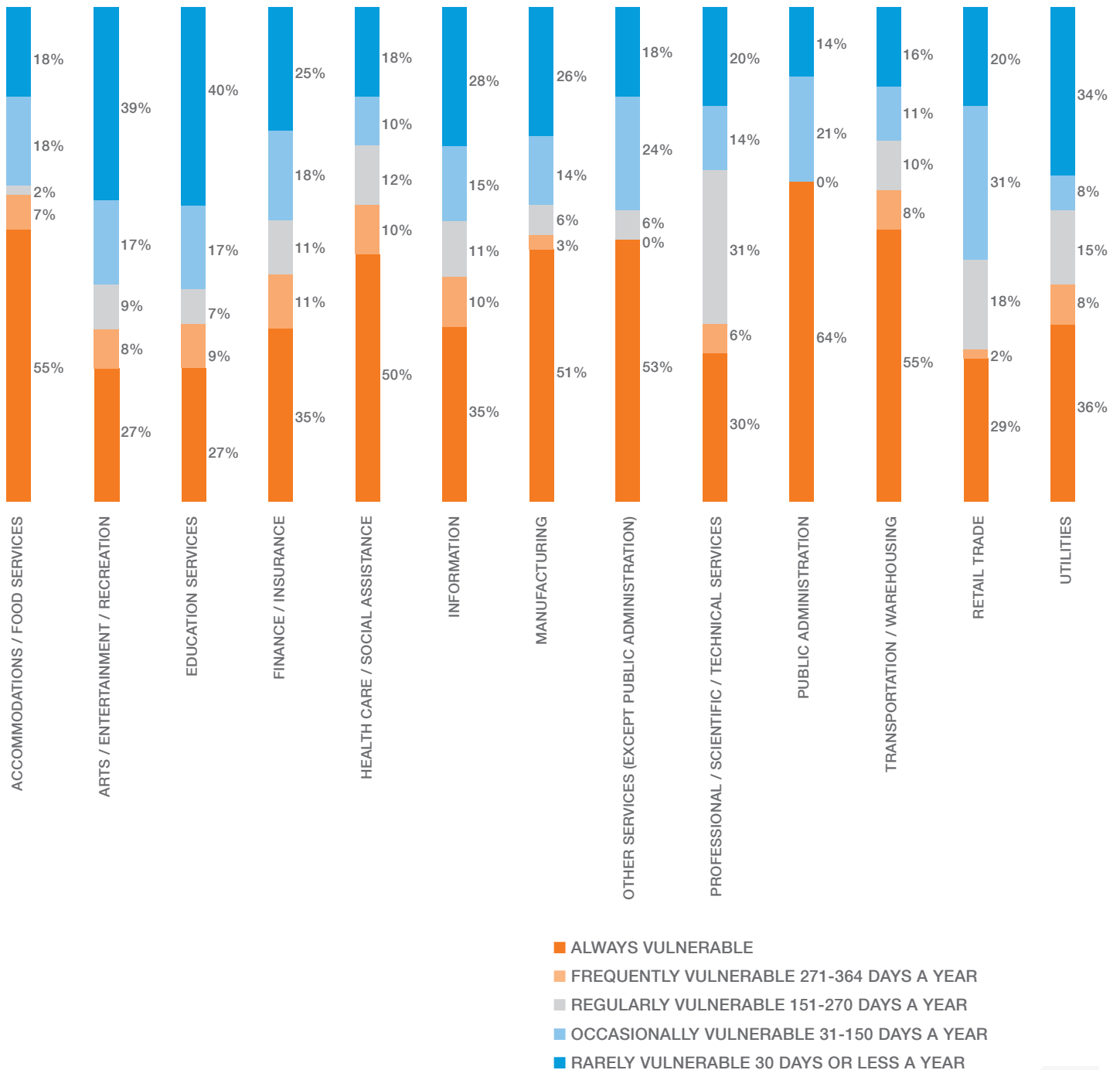


Finance / Insurance



Window of exposure is an organizational key performance indicator that measures the number of days a website has at least one serious vulnerability over a given period of time.

Window of Exposure



Survey Analysis

Overview

The analysis is based on 118 responses on a survey sent to security professionals to measure maturity models of application security programs at various organizations.

The responses obtained in the survey are correlated with the data available in Sentinel to get deeper insights.

- Sentinel data was pulled for 2014 timeframe.
- Data was pulled from sites that were assessed with WhiteHat's premium service covering all WASC vulnerability classes.
- Data included all vulnerability classes except Insufficient Transport Layer Protection, Directory Indexing, URL Redirector Abuse, Improper File System permissions, and Fingerprinting

Survey Responses

Total Responses: 118

- Information, and Finance / Insurance have the highest number of responses.
- Other industries do not have enough responses to draw meaningful industry level conclusions from the survey.

Summary of Survey Analysis

24% of the survey respondents have experienced a data or system breach.

- In Finance / Insurance, 17% have experienced a data or system breach
- In Information, 20% have experienced a data or system breach.

56% of all respondents did not hold any part of the organization accountable in case of data or system breach. Listed below is how various parts of organizations are held responsible for data or system breach:

- Board of Directors 8%
- Executive Management 27%
- Software Development 26%
- Security Department 29%

Risk Reduction is the most commonly cited reason (with 35% of the respondents) for resolving website vulnerabilities. Only 14% of the respondents cited Compliance as the primary reason for resolving website vulnerabilities.

Static Analysis:

- 87% of the respondents perform static analysis. 32% perform it with each major release and 13% perform it daily.

Penetration Testing

- 92% of the respondents perform penetration testing. 21% perform it annually, 26% perform it quarterly and 8% never perform penetration testing.

Basic Adversarial testing

Organizations that do not perform basic adversarial testing tend to have higher number of open vulnerabilities than those that do perform it.

- Open vulnerabilities when adversarial testing is performed on each major release: 12
- Open vulnerabilities when adversarial testing is performed every quarter: 9
- Open vulnerabilities when adversarial testing is never performed: 34

Organizations that do not perform basic adversarial testing have lower remediation rate than those that do perform it.

- Remediation rate when adversarial testing is performed on each major release: 19%
- Remediation rate when adversarial testing is performed every quarter: 50%
- Remediation rate when adversarial testing is never performed: 11%

79% of the respondents performed ad-hoc code reviews on high risk applications

Organizations that do not perform ad-hoc code reviews on high risk applications have higher open vulnerabilities than the overall average open vulnerabilities.

- Open vulnerabilities when adhoc code review is never performed: 35
- Open vulnerabilities when adhoc code review is performed in a planned manner: 6
- Open vulnerabilities when adhoc code review is performed with each major release: 10
- Remediation rate when adhoc code review is never performed: 18%
- Remediation rate when adhoc code review is performed in a planned manner: 25%
- Remediation rate when adhoc code review is performed with each major release: 29%

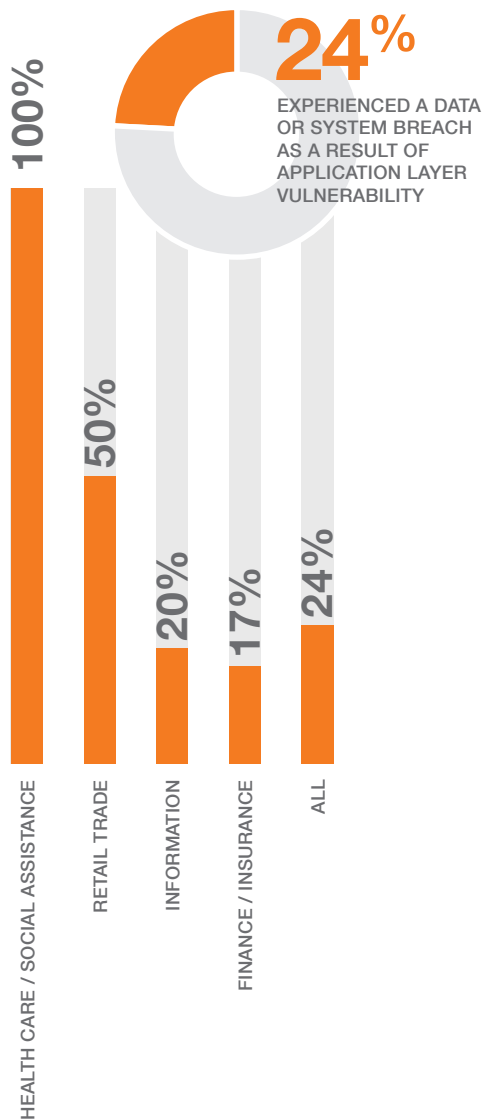
This is how integrating application security best practices into the SDLC processes affected vulnerability count and remediation rate:

- After QA team began performing adversarial testing, average number of open vulnerabilities declined by 64% (from 13 to 5) and average remediation rate increased from 30% to 33%
- After organizations began using penetration testers, average number of open vulnerabilities declined by 65% (from 31 to 11) and average remediation rate increased from 22% to 31%
- After organizations began performing adhoc code reviews, average number of open vulnerabilities declined by 59% (from 32 to 13) and average remediation rate increased from 36% to 38%
- After organizations began sharing security result reviews with the QA Department, average number of open vulnerabilities declined 21% (from 20 to 16) and average remediation rate grew from 35% to 42%
- After incident response plan was updated, average open vulnerability count declined 60% (from 12 to 5) while average remediation rate declined from 29% to 28%
- After organizations began performing architecture analysis, average open vulnerability count declined 47% from 12 to 6 while average remediation rate declined from 32% to 31%
- After organizations began performing security focused design reviews, average open vulnerabilities count declined 17% from 8 to 7 while average remediation rate went up from 33% to 37%
- After organizations began empowering a group to take the lead in performing architecture analysis, average number of open vulnerabilities declined by 43% (from 9 to 5) while average remediation rate declined from 40% to 36%
- After organizations began using a risk questionnaire to rank applications, average number of vulnerabilities declined 35% from 9 to 6, while average remediation rate declined from 39% to 38%
- After organizations began feeding penetration testing results back to development, average open vulnerabilities declined by 45% (from 12 to 7) while average remediation rate went up from 27% to 41%

Have any of your organizations website(s) experienced a data or system breach as a result of an application layer vulnerability?

24% of the survey respondents have experienced a data or system breach

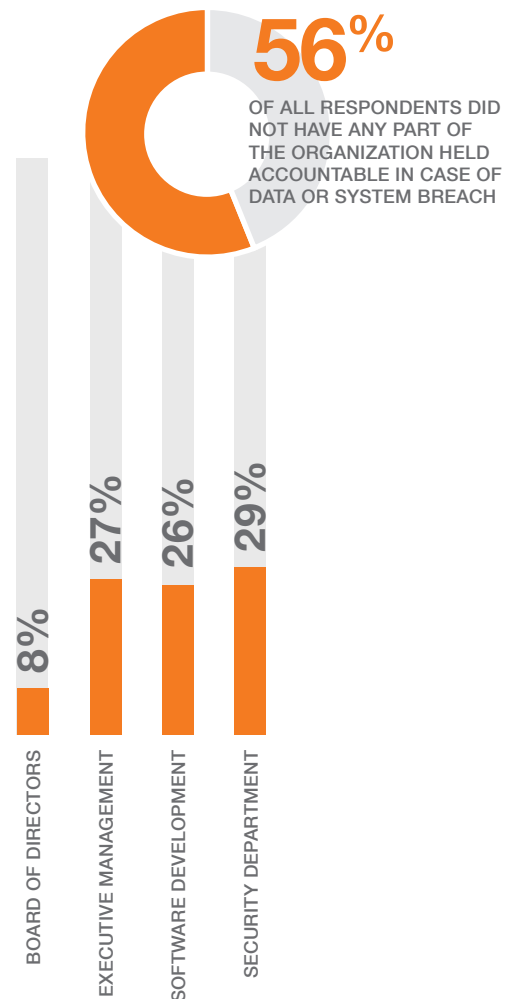
- In Finance / Insurance 17% have experienced a data or system breach
- In Information, 20% have experienced a data or system breach



If an organization experiences a website(s) data or system breach, which part of the organization is held accountable and what is its performance?

56% of all respondents did not hold any part of the organization accountable in case of data or system breach.

- Board of Directors 8%
- Executive Management 27%
- Software Development 26%
- Security Department 29%

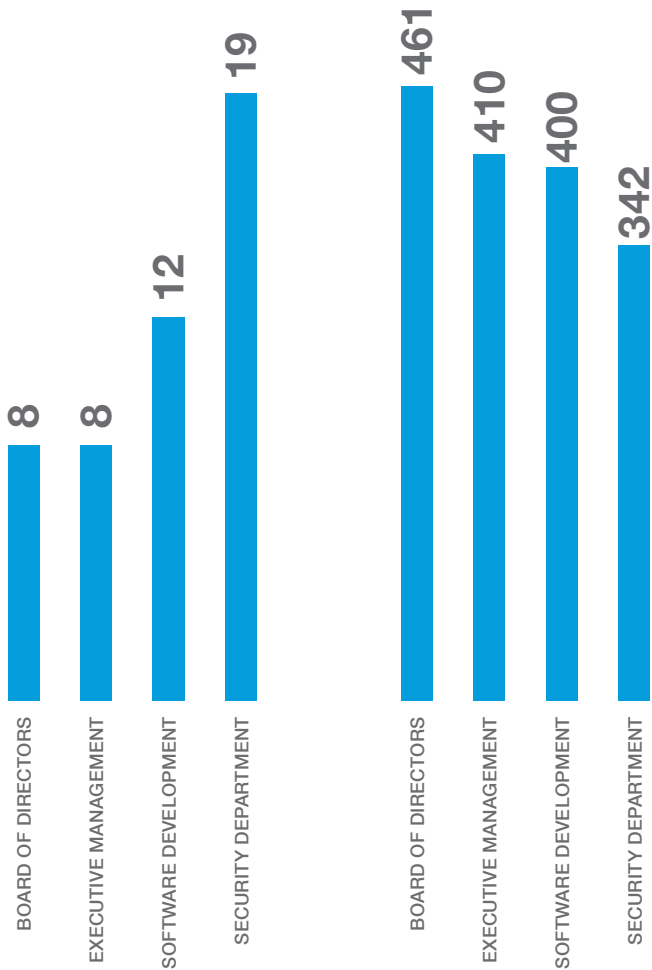


If an organization experiences a website(s) data or system breach, which part of the organization is held accountable and what is its performance?

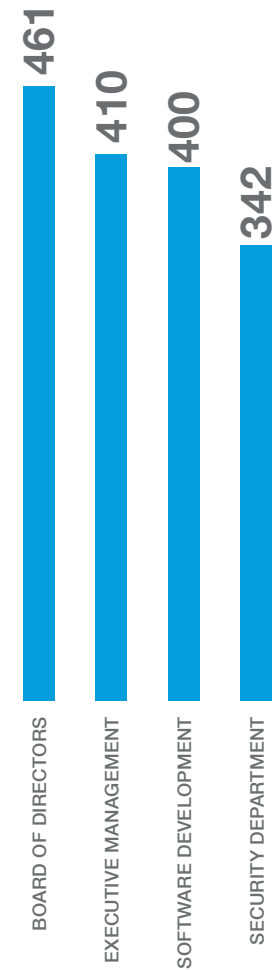
- Count of open vulnerabilities is lowest (at 8) and remediation rate is highest at 40% when Board of Directors is held responsible for breach.
- Remediation rate is lowest (at 19%) when software development is held accountable for a system breach.
- Average number of open vulnerabilities is highest (at 19) when security department is held accountable for a system breach.

- Organizations with accountability tend to find and fix more vulnerabilities than those that don't have clear accountability.
- 24% remediation in organizations without accountability vs. 33% for those with accountability.
- 16 average open vulnerabilities in organizations with accountability versus 13 in those without accountability.

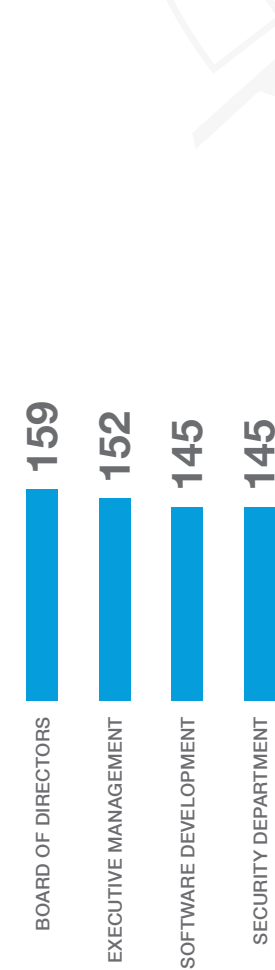
Average Number of Vulnerabilities Open



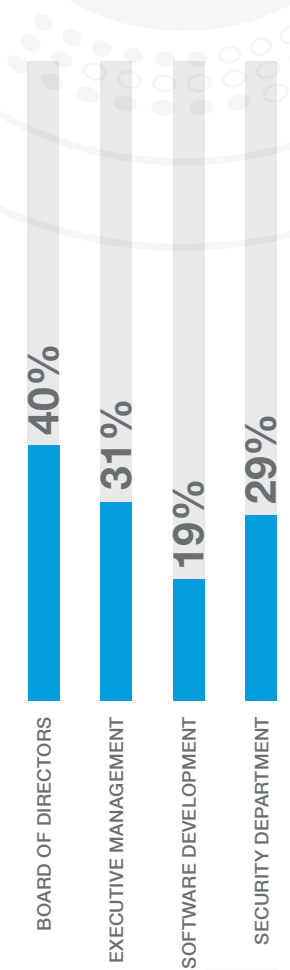
Average Time Open



Average Time to Fix



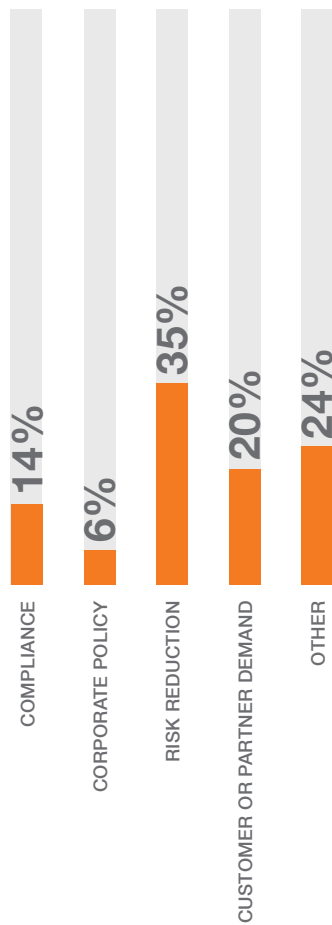
Remediation Rate



Please rank your organization's drivers for resolving website vulnerabilities. 1 being the lowest priority, 5 the highest.

- 14% of the respondents cite Compliance as the primary reason for resolving website vulnerabilities
- 6% of the respondents cite Corporate Policy as the primary reason for resolving website vulnerabilities
- 35% of the respondents cite Risk Reduction as the primary reason for resolving website vulnerabilities
- 20% of the respondents cite Customer or Partner Demand as the primary reason for resolving website vulnerabilities
- 24% of the respondents cite other reasons for resolving website vulnerabilities

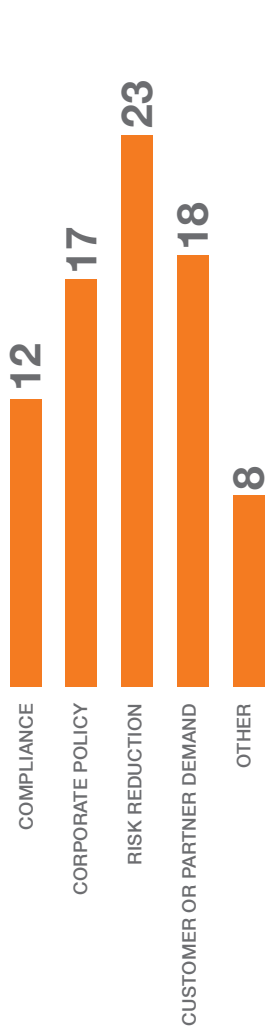
Primary Driver for Resolving Website Vulnerabilities



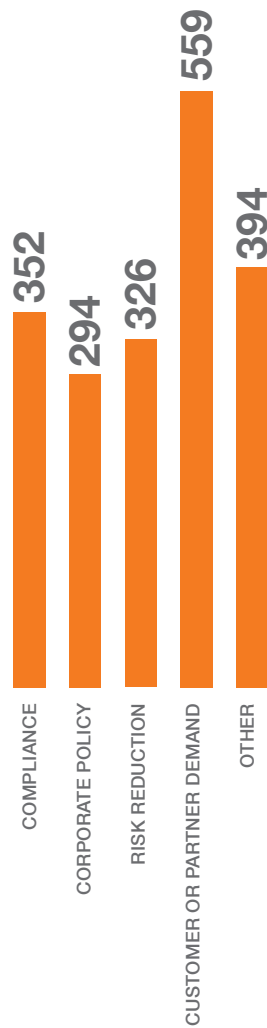
Please rank your organization's drivers for resolving vulnerabilities.

- Average number of open vulnerabilities is highest (at 23) when Risk Reduction is the primary reasons for fixing vulnerabilities.
- Average remediation rate is highest at 86% when compliance is the primary driver for fixing vulnerabilities.

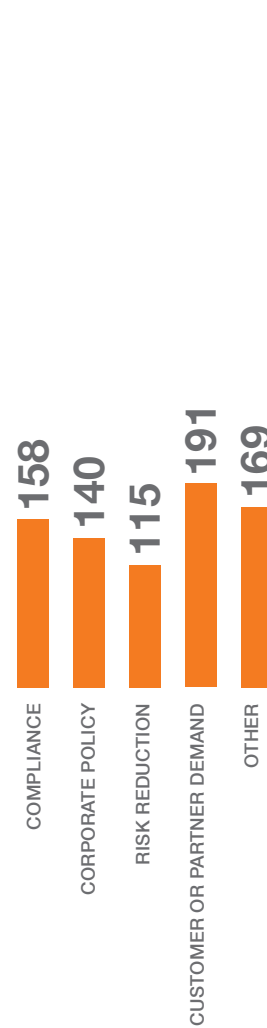
Average Number of Vulnerabilities



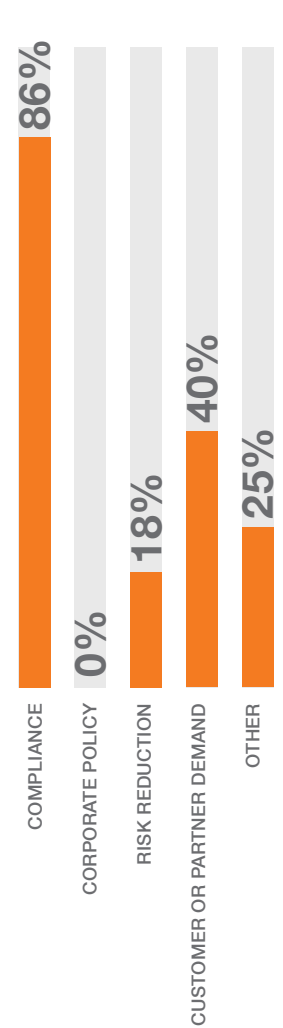
Average Time Open



Average Time to Fix



Remediation Rate



How frequently do you perform automated static analysis during the code review process?

Percent of respondents for various frequencies of automatic static analysis:

- Daily: 13%
- With each major release: 32%
- Never: 13%

Number of open vulnerabilities for various frequencies of automatic static analysis:

- Daily: 5
- With each major release: 28
- Never: 12

Average time open for various frequencies of automatic static analysis:

- Daily: 400 days
- Each major release: 325 days
- Never: 423 days

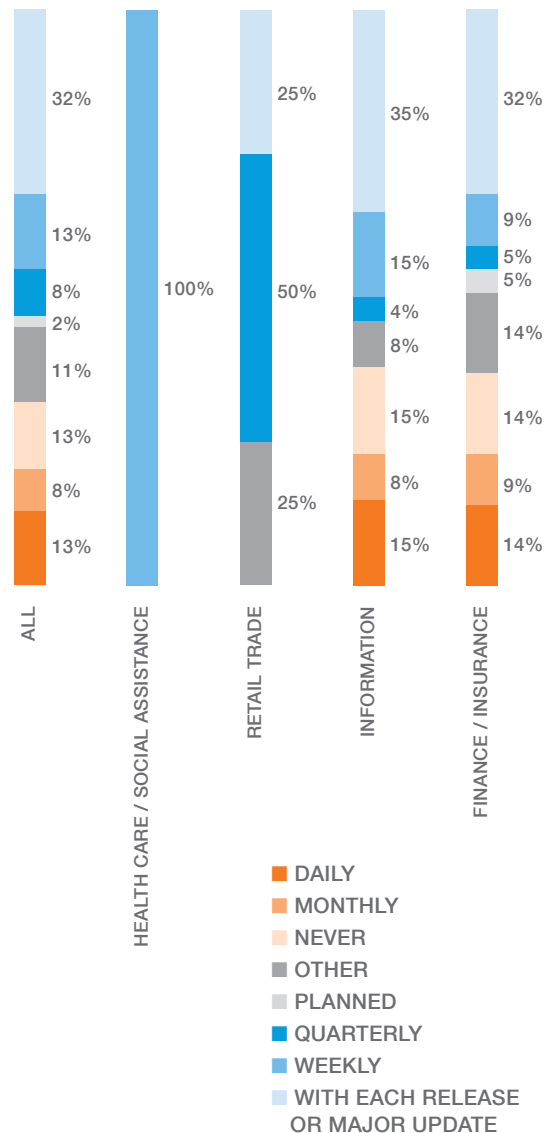
Remediation rate for various frequencies of automatic static analysis:

- Daily: 17%
- Each major release: 38%
- Never: 29%

Time to fix for various frequencies of automatic static analysis:

- Daily: 96 days
- Each major release: : 138 days
- Never: 157 days

Frequency of Automated Static Analysis by Industry



How frequently does the QA team go beyond functional testing to perform basic adversarial tests (probing of simple edge cases and boundary conditions; example: What happens when you enter the wrong password over and over?)

% of respondents for various frequencies of adversarial testing:

- Each major release: 32%
- Quarterly: 11%
- Never: 21%

Number of open vulnerabilities for various frequencies of adversarial testing:

- Each major release: 12
- Quarterly: 9
- Never: 34

Average time open for various frequencies of adversarial testing:

- Each major release: 383 days
- Quarterly: 391 days
- Never: 295 days

Remediation rate for various frequencies of adversarial testing:

- Each major release: 19%
- Quarterly: 50%
- Never: 11%

Time-to-fix for various frequencies of adversarial testing:

- Each major release: 144 days
- Quarterly: 139 days
- Never: 153 days

How frequently do you use external penetration testers to find problems?

% of respondents for various frequencies of penetration testing:

- 21% Annually
- 26% Quarterly
- 8% Never

Number of open vulnerabilities for various frequencies of penetration testing:

- Annually: 10
- Quarterly: 32
- Never: 22

Average time open for various frequencies of penetration testing:

- Annually: 292 days
- Quarterly: 302 days
- Never: 431 days

Remediation rate for various frequencies of penetration testing:

- Annually: 50%
- Quarterly: 36%

Time-to-fix for various frequencies of penetration testing:

- Annually: 168 days
- Quarterly: 116 days
- Never: 149 days

How often does your organization use defects identified through operations monitoring feedback to development and used to change developer behavior?

% of respondents for various frequencies of operation monitoring feedback:

- 17% Daily
- 17% With each major release
- 9% Never

Number of open vulnerabilities for various frequencies of operation monitoring feedback:

- Daily: 38
- With each major release: 19
- Never: 6

Average time open for various frequencies of operation monitoring feedback:

- Daily: 332 days
- With each major release: 369 days
- Never: 273 days

Remediation rate for various frequencies of operation monitoring feedback:

- Daily: 13%
- With each major release: 44%
- Never: 0%

Time-to-fix for various frequencies of operation monitoring feedback:

- Daily: 99 days
- With each major release: 218 days
- Never: 121 days

How frequently does your organization perform ad hoc code reviews of high risk applications in an opportunistic fashion?

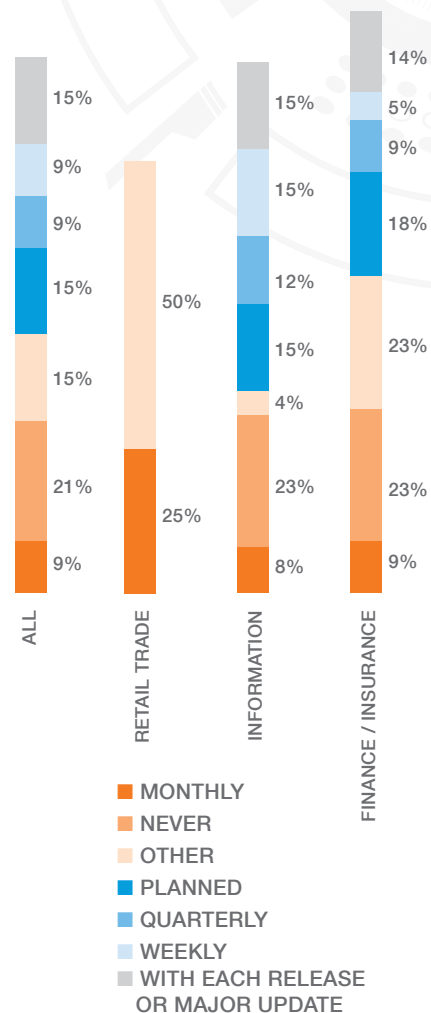
% of respondents for various frequencies of ad hoc code reviews:

- 21% Never
- 15% Planned
- 15% with each major release

Number of open vulnerabilities for various frequencies of ad hoc code reviews:

- 35 Never
- 6 Planned
- 10 with each major release

Frequency of Adhoc Code Review by Industry



Average time open for various frequencies of ad hoc code reviews:

- Never: 335 days
- Planned: 282 days
- With each major release: 293 days

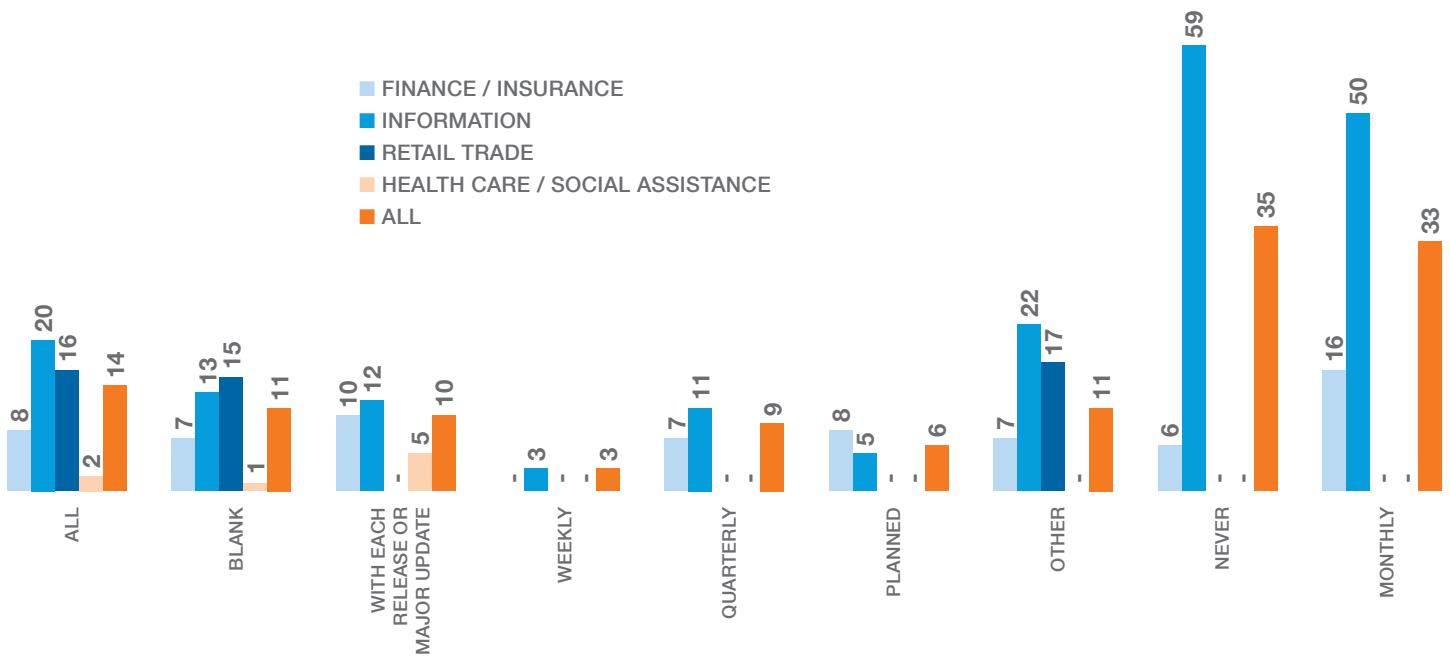
Remediation rate for various frequencies of ad hoc code reviews:

- Never: 18%
- Planned: 25%
- With each major release: 29%

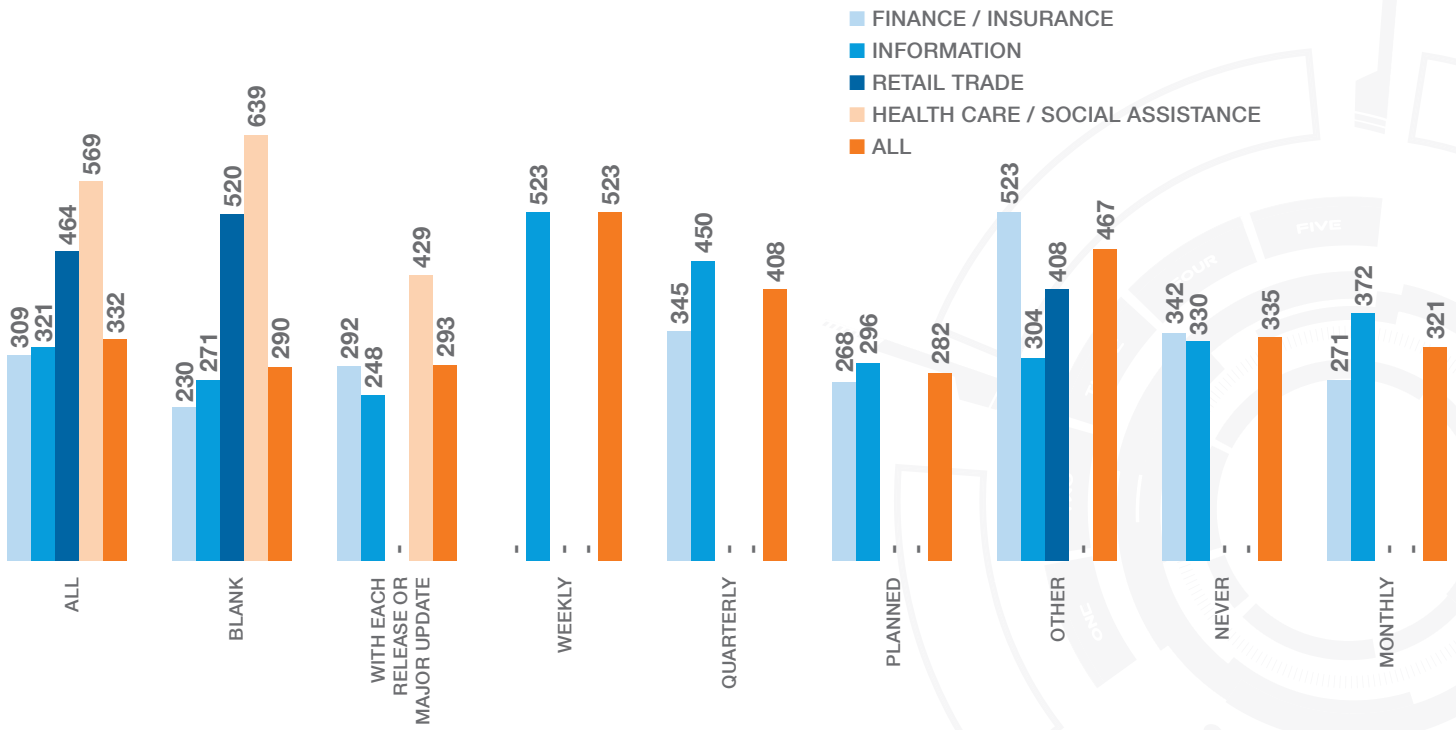
Time-to-fix for various frequencies of ad hoc code reviews:

- Never: 163 days
- Planned: 117 days
- With each major release: 133 days

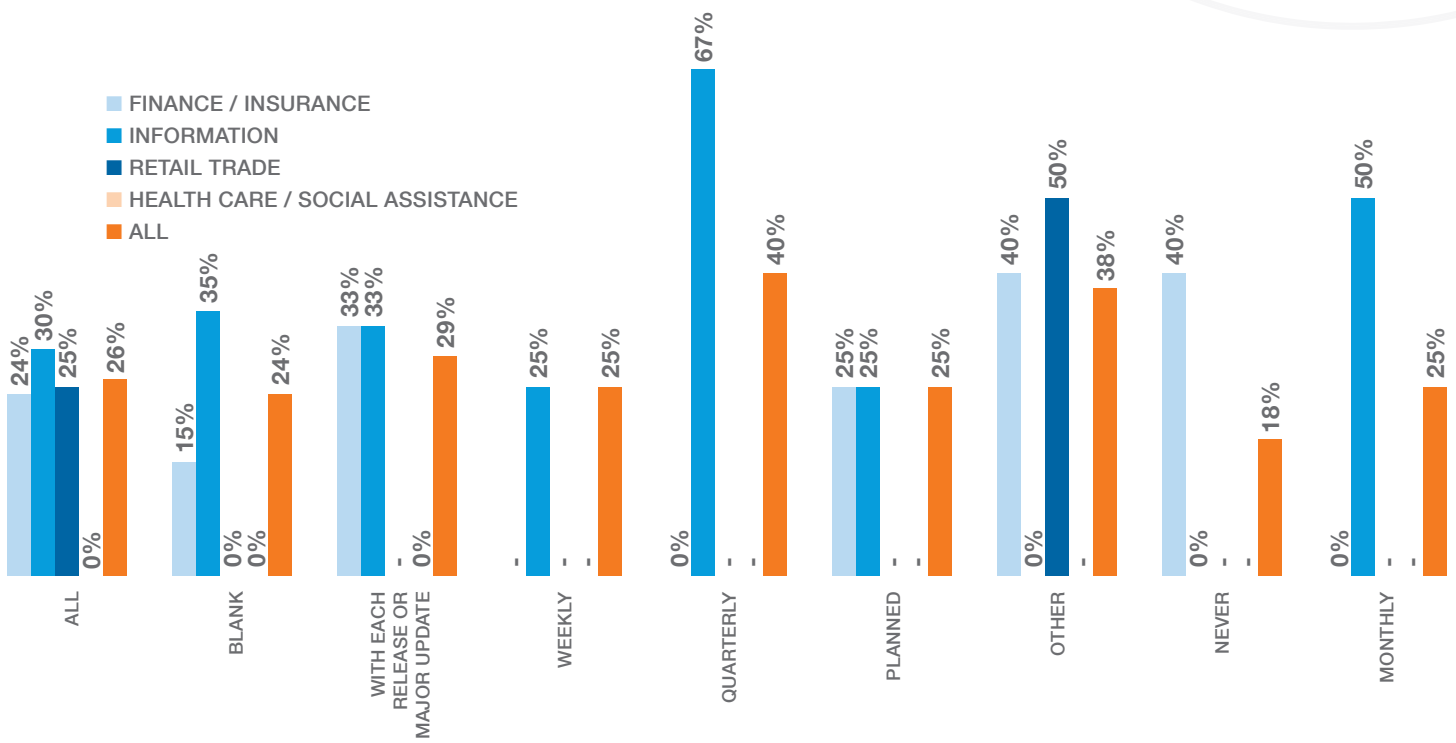
Average Number of Vulnerabilities at Different Frequencies of Adhoc Code Review



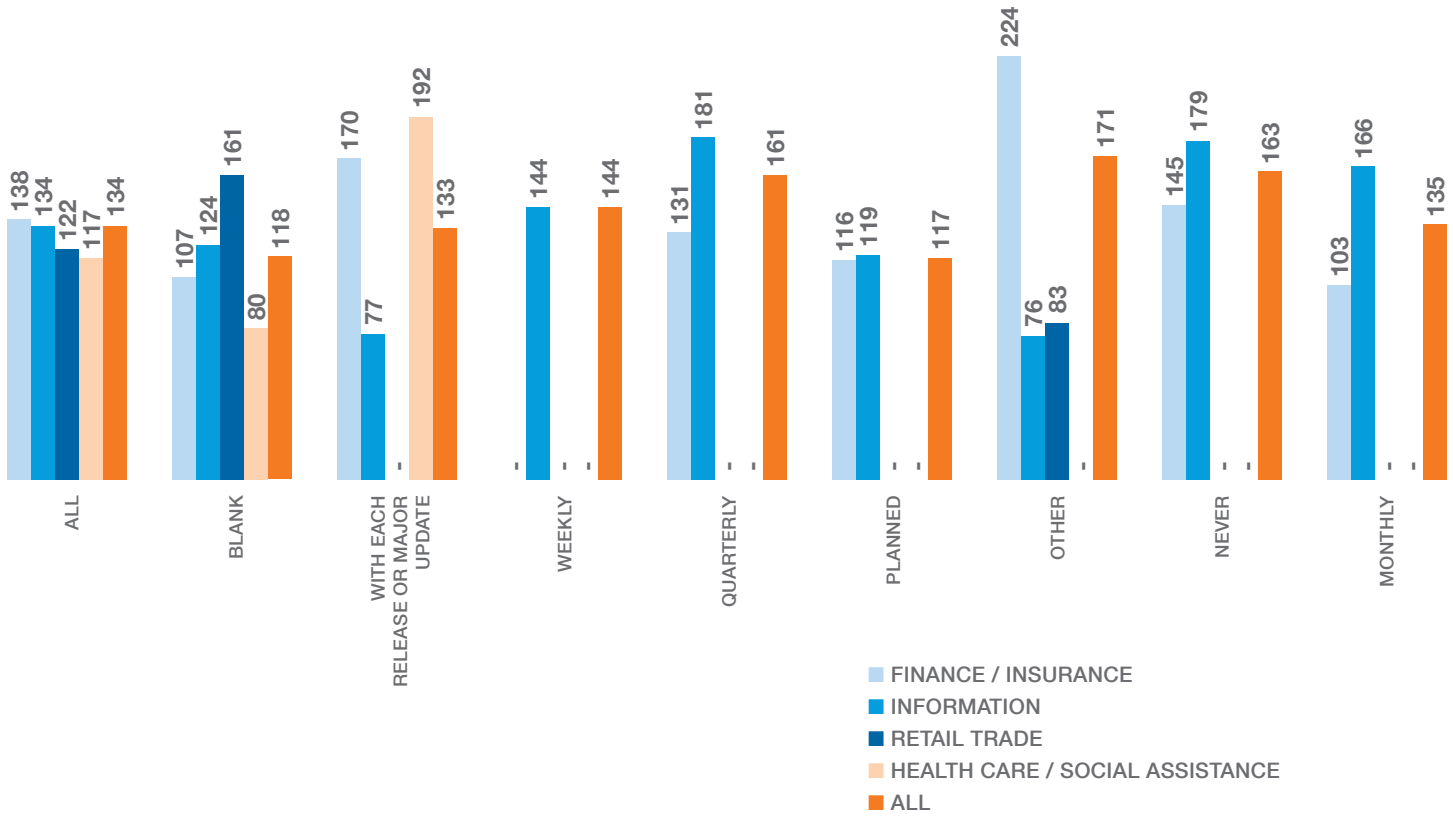
Average Time Open at Different Frequencies of Adhoc Code Review



Average Remediation Rate at Different Frequencies of Adhoc Code Review



Average Time-to-Fix at Different Frequencies of Adhoc Code Review



How frequently does your organization share results from security reviews with the QA department?

% of respondents for various frequencies of security review sharing:

- Monthly: 13%
- With each major release: 28%
- Never: 19%

Number of open vulnerabilities for various frequencies of security review sharing:

- Monthly: 10
- With each major release: 26
- Never: 18

Average time open for various frequencies of security review sharing:

- Monthly: 309 days
- With each major release: 436 days
- Never: 307 days

Remediation rate for various frequencies of security review sharing:

- Monthly: 43%
- With each major release: 21%
- Never 0%

Time-to-fix for various frequencies of security review sharing:

- Monthly: 116 days
- With each major release: 192 days
- Never: 122 days

When did your organization incorporate automated static analysis into the code review process?

After incorporating static analysis into the code review process:

- Average number of vulnerabilities slightly increased (from 15 to 18)
- Average time-to-fix declined (from 174 days to 150 days)
- Average time open increased (175 days to 197 days)
- Remediation rate declined (from 33% to 29%)

When did the QA team begin performing basic adversarial testing?

After QA team began performing basic adversarial testing:

- Average number of vulnerabilities declined (from 13 to 5)
- Average time-to-fix declined (from 97 days to 94 days)
- Average time open increased (295 days to 432 days)
- Remediation rate increased (from 30% to 33%)

When did your organization begin using penetration testers?

After organizations began using penetration testers:

- Average number of vulnerabilities declined (from 31 to 11)
- Average time-to fix decreased (from 203 days to 195 days)
- Average time open increased (from 198 days to 257 days)
- Remediation rate increased (from 22% to 31%)

When did your organization begin performing ad hoc code reviews?

After organizations began performing ad hoc code reviews:

- Average number of vulnerabilities declined (from 32 to 13)
- Average time to fix declined (from 191 days to 174 days)
- Average time open increased (from 202 days to 282 days)
- Remediation rate increased (from 36% to 38%)

When did your organization begin sharing results from security reviews with the QA department?

After organizations began sharing security review results with the QA department:

- Average number of vulnerabilities declined (from 20 to 16)
- Average time-to-fix declined (from 179 days to 175 days)
- Average time open increased (from 214 days to 246 days)
- Remediation rate increased (from 35% to 42%)

When was your incident response plan updated to include application security?

After incident response plan is updated to include application security:

- Average number of vulnerabilities declined (from 12 to 5)
- Average time-to-fix increased (from 216 days to 221 days)
- Average time open increased (from 188 days to 220 days)
- Remediation rate decreased (from 29% to 28%)

When did you begin performing architecture analysis focused on security features (authentication, access control, use of cryptography, etc.)?

After organizations began performing architecture analysis:

- Average number of vulnerabilities declined (from 12 to 6)
- Average time-to-fix decreased (from 285 days to 280 days)
- Average time open increased (from 182 days to 245 days)
- Remediation rate decreased (from 32% to 31%)

When did your organization begin using operational monitoring to improve or change developer behavior?

After organizations began using operational monitoring:

- Average number of vulnerabilities declined (from 4 to 3)
- Average time-to-fix increased (from 135 days to 151 days)
- Average time open increased (from 195 days to 304 days)
- Remediation rate decreased (from 37% to 34%)

When did your organization begin performing security focused design reviews of web applications?

After organizations began performing security focused design reviews:

- Average number of vulnerabilities declined (from 8 to 7)
- Average time-to-fix declined (from 230 days to 202 days)
- Average time open increased (from 226 days to 284 days)
- Remediation rate increased (from 33% to 37%)

When did your organization form or empower a group to take a lead in performing architecture analysis?

After organizations began forming a group to take a lead in architecture analysis:

- Average number of vulnerabilities declined (from 9 to 5)
- Average time-to-fix declined (from 184 days to 165 days)
- Average time open increased (from 237 days to 348 days)
- Remediation rate declined (from 40% to 36%)

When did your organization begin using a risk questionnaire to rank applications?

After organizations began using a risk questionnaire:

- Average number of vulnerabilities declined (from 9 to 6)
- Average time-to-fix decreased (from 160 days to 155 days)
- Average time open increased (from 163 days to 244 days)
- Remediation rate declined (from 39% to 38%)

When did your organization begin maintaining a company specific top N list of the most important kinds of bugs that need to be eliminated?

After organizations began maintaining a company specific top N list of the most important kinds of bugs that need to be eliminated:

- Average number of vulnerabilities declined (from 8 to 7)
- Average time-to-fix declined (from 300 days to 243 days)
- Average time open increased (from 183 days to 239 days)
- Remediation rate increased (from 39% to 46%)

When did your organization begin feeding penetration-testing results back to development through established defect management or mitigation channels/systems?

After organizations began feeding penetration-testing results back to development:

- Average number of vulnerabilities declined (from 12 to 7)
- Average time-to-fix declined (from 207 days to 197 days)
- Average time open increased (from 209 days to 270 days)
- Remediation rate increased (from 27% to 41%)

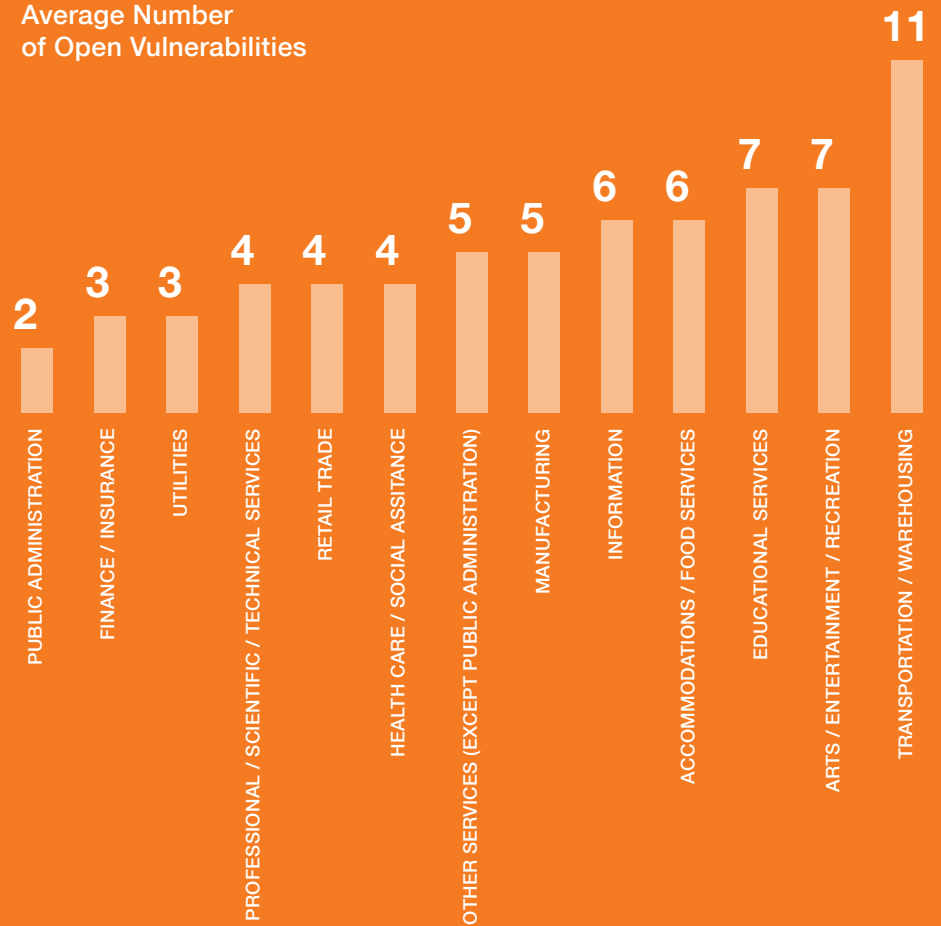
Have any of your organizations website(s) experienced a data or system breach as a result of an application layer vulnerability?

- Those who have experienced a data or system breach have higher average number of open vulnerabilities than those who haven't experienced a breach (18 vs. 17)
- Those who have experienced a breach have lower remediation rate than those who haven't experienced a breach (34% vs. 27%)
- Those who have experienced a breach have higher average time open than those who haven't experienced a breach (361 days vs. 394 days)
- Those who have experienced a breach have lower average time to fix than those who haven't experienced a breach (130 days vs. 155 days)

Average Number of Open Vulnerabilities

While the window of exposure is high for websites, average number of open vulnerabilities is relatively small, ranging from 2 (for Public Administration sites) to 11 (for Transportation and Warehousing sites). Finance / Insurance, Health Care / Social Assistance, Retail Trade and Information have average number of open vulnerabilities fairly low at 3, 4, 4 and 6 respectively.

Average Number of Open Vulnerabilities



Average Days Open

On average, vulnerabilities stay open for a long time in all industries. The smallest average time open is observed in Transportation and Warehousing industry (at 299 days or ~1 year) and the longest average time open is observed in Public Administration industry (at 1033 days, or ~3 years). Listed below are the average time open data for some of the key industries:

Health Care / Social Assistance: 572 days (~1.6 years)

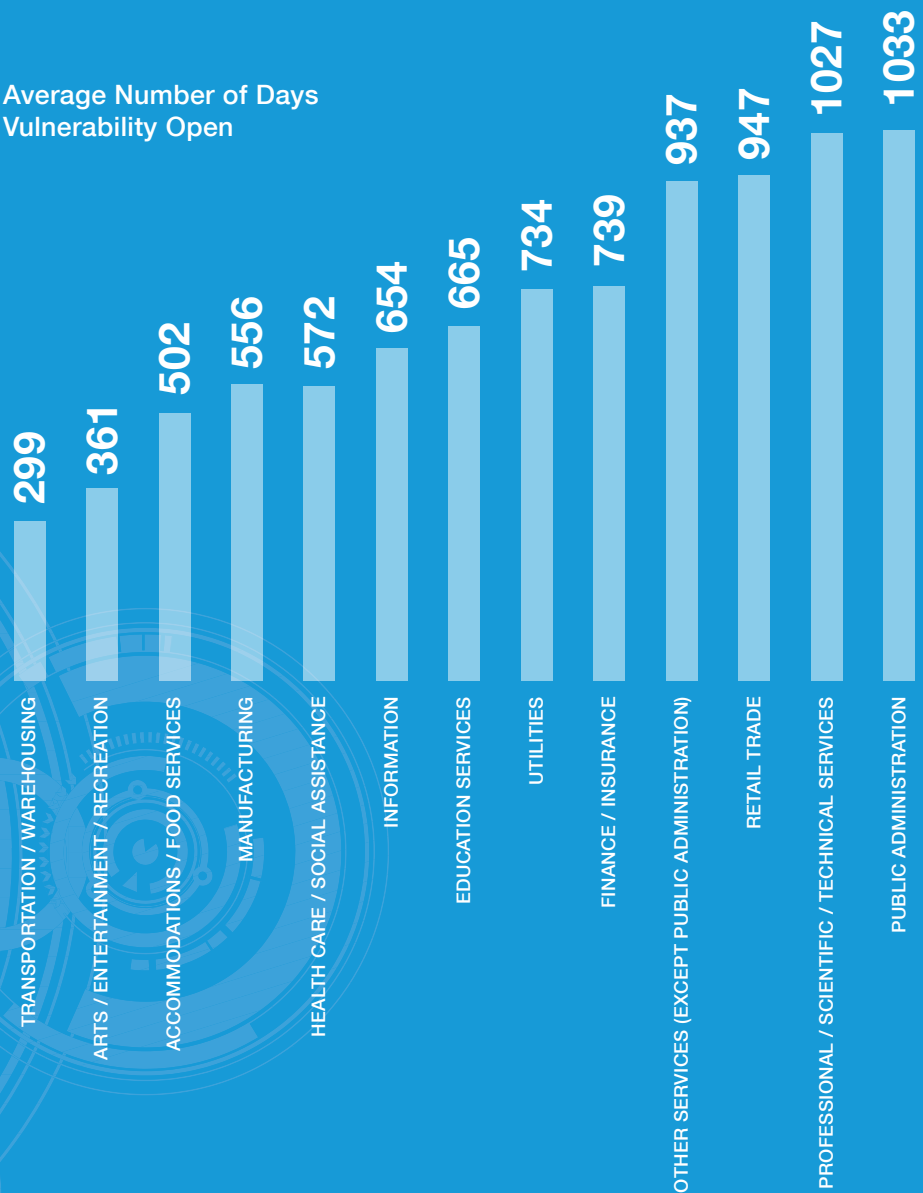
Information: 654 days (~1.8 years)

Finance / Insurance: 739 days (~2 years)

Retail Trade: 947 days (~2.6 years)

Retail trade ranked third from the bottom after Professional, Scientific, and Technical Services (with 1027 average days open) and Public Administration (1033 days open)

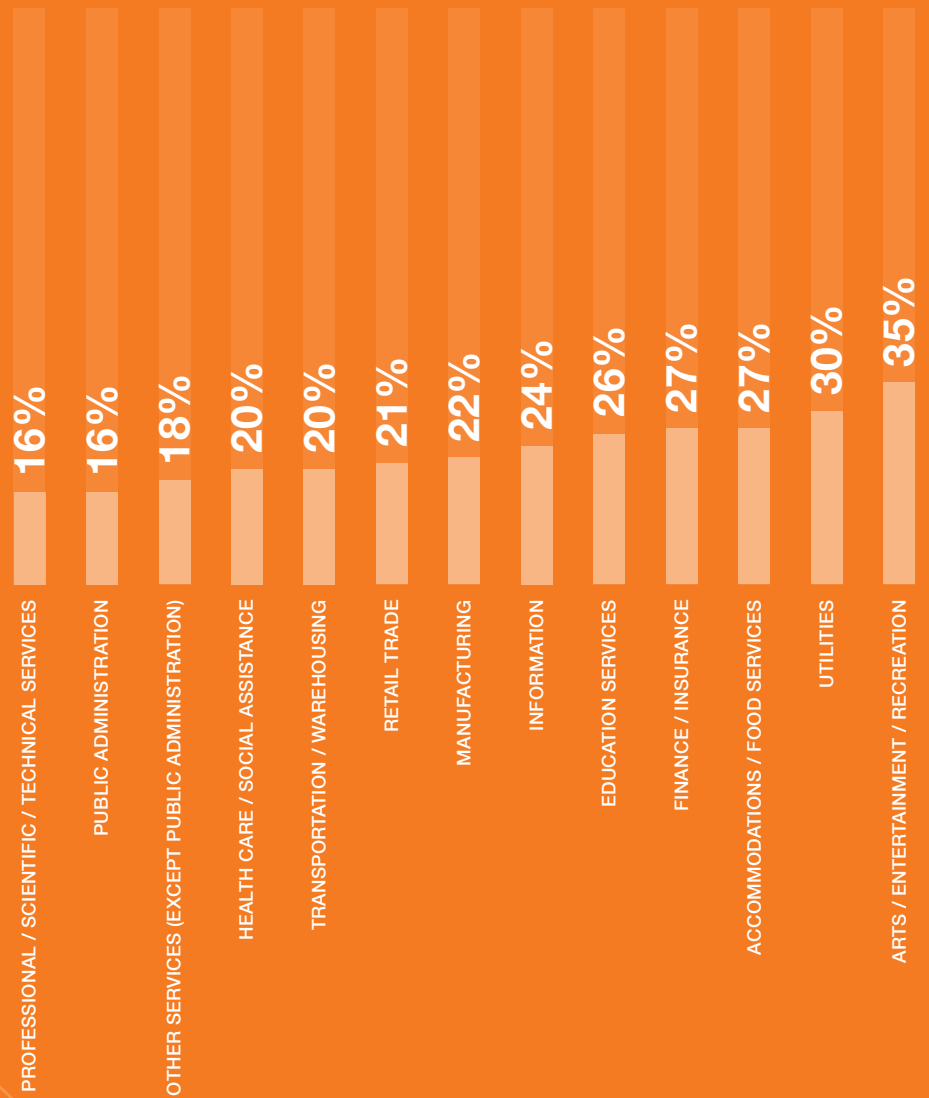
Average Number of Days Vulnerability Open

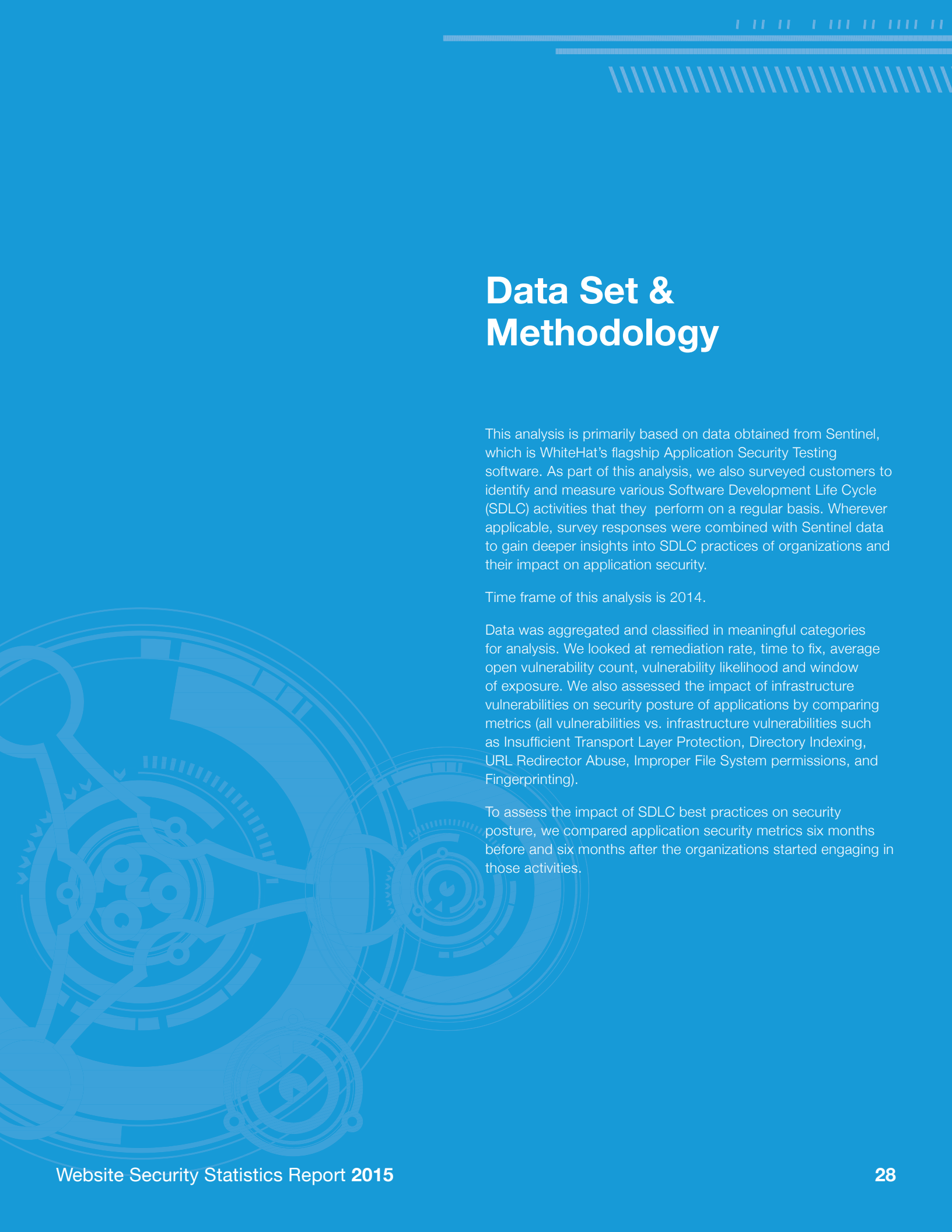


Remediation Rates

Average remediation rate for industries varies significantly from 16% (for Professional, Scientific, and Technical Services sites) to 35% (for Arts, Entertainment, and Recreation sites). Sites in Health Care / Social Assistance, Retail Trade and Information industries have comparatively low average remediation rates at 20%, 21% and 24% respectively. Finance / Insurance sites have an average remediation rate of 27%.

Average Remediation Rate





Data Set & Methodology

This analysis is primarily based on data obtained from Sentinel, which is WhiteHat's flagship Application Security Testing software. As part of this analysis, we also surveyed customers to identify and measure various Software Development Life Cycle (SDLC) activities that they perform on a regular basis. Wherever applicable, survey responses were combined with Sentinel data to gain deeper insights into SDLC practices of organizations and their impact on application security.

Time frame of this analysis is 2014.

Data was aggregated and classified in meaningful categories for analysis. We looked at remediation rate, time to fix, average open vulnerability count, vulnerability likelihood and window of exposure. We also assessed the impact of infrastructure vulnerabilities on security posture of applications by comparing metrics (all vulnerabilities vs. infrastructure vulnerabilities such as Insufficient Transport Layer Protection, Directory Indexing, URL Redirector Abuse, Improper File System permissions, and Fingerprinting).

To assess the impact of SDLC best practices on security posture, we compared application security metrics six months before and six months after the organizations started engaging in those activities.

Conclusion & Recommendations

In this year's report, we strive to make one thing perfectly clear: we at WhiteHat Security, and the industry at large, have become incredibly adept at finding vulnerabilities. And while everyone should continue to look and increase their skills at finding vulnerabilities, it has become crucial for everyone to focus on helping make the vulnerability remediation process faster and easier. Remediation, more than anything else, is the hardest problem in application security. It should go without saying that vulnerabilities found but not fixed, does not make things more secure. Making the web progressively more secure is the mission that we as a community are collectively working towards every day. And together, we can do exactly that!

This is also a good opportunity to look back on everything we have learned in our quest to figure out what works and what does not in application security, both technically and procedurally. What is it that really makes some websites, and their underlying code, secure – or at least more secure than others? That's the question we have been seeking to answer since we started this research. Answering that question first required us to know approximately how many or what kinds of vulnerabilities exist in the average website and how long they remain exposed as a way to measure performance.

We accomplished this and in the process we learned a great deal: we learned that vulnerabilities are plentiful, they stay open for weeks or months, and typically only half get fixed. And while a great many websites are severely lacking in security, many websites are actually quite secure. So, what's the difference between them? Is it the programming language that matters when it comes to security? Is it the industry the organizations are in? Is it the size of the organization? Is it the process they use to develop their software? Is it something else?

At present time we can say that all of these aforementioned items don't matter much, and if they do, it's only slightly and under very specific conditions. On the whole, what matters more than anything else ends up first being a non-technical answer – visibility and accountability. The websites and organizations that are more secure than others have a solid understanding of the performance of their software development lifecycle and have developed a security metrics program that best reflects how to maintain security across that lifecycle. Additionally, these same organizations have a culture of accountability – both in terms of when and if a breach occurs – and they can measure performance. Without an executive-level mandate, it's going to be very challenging, if not impossible, to adequately protect an organization's systems. The incentives simply won't be in alignment.

And here is the point where we get to very specific guidance as a take away from this report. Like we've recommended many times in previous reports, the first order of business is to determine what websites an organization owns and then to prioritize as much metadata about those websites as possible. Grouping them by department or business unit is even better. Secondly, through dynamic or static vulnerability assessment, begin creating an application security metrics program; something that tracks the volume and type of vulnerabilities that exist, how long reported issues take to get fixed, and the percentage that are actually getting fixed. As the saying goes, anything measured tends to improve. With visibility through data, the answers to the problem become much clearer.

Once these steps have been achieved, the organization can then set goals for which metrics need to improve, by how much and when. With these goals in hand, it becomes much easier and more efficient to begin implementing or improving the SDLC process with very specific activities designed to positively affect whatever metrics that are missing. For example, if the reasons SQL Injection vulnerabilities are not getting fixed fast or comprehensively enough is that the developers are not well educated on that type of vulnerability? If so, the organization might decide to host a workshop that focuses just on that class of attack. Or perhaps the reason so many Cross-site Scripting vulnerabilities enter the system with each release is the lack of a helpful centralized security framework. In which case, create one, advertise it's existence internally, and mandate its usage.

Tactical approaches like the above that are straight-forward and customizable are ideal because very little in application security is one-size-fits-all. Every organization is different: the software being built is different; the tolerance for risk is different; the goal in the market place is different. These variables cannot be accounted for in a one-size-fits all model. So, what security teams can do is support the SDLC process by bringing visibility and expertise to the table and let the business guide what's acceptable from an outcome perspective. Steadily adding, improving, and measuring the effect of very specific security controls is the best way to ensure better and more secure code.

Definitions

Days Open: This represents the number of days a vulnerability has been open. This is calculated by subtracting the date the vulnerability opened from the current date. Days Open is calculated for currently open vulnerabilities only.

Time to Fix: The time to fix is the time it takes to fix vulnerabilities and is calculated for vulnerabilities that have a close date.

Remediation Rate: The Remediation Rate is the ratio of the number closed vulnerabilities over the number of open vulnerabilities. It is calculated over a window of time. Vulnerability is considered closed if it closed during the analysis period. Vulnerability is considered open if it was open during the analysis period.

Vulnerability Class Likelihood: Likelihood is calculated as the number of sites that have at least one open vulnerability in a given class over the total number of active sites.

Window of Exposure: This is calculated as the number of sites that had at least one serious vulnerability open over the analysis period.

Serious Vulnerability: Vulnerability with a severity of 3 or greater as defined by WhiteHat's Vulnerability Classification System.



About WhiteHat Security

Founded in 2001 and headquartered in Santa Clara, California, WhiteHat Security is the leader in application security, enabling businesses to protect critical data, ensure compliance, and manage risk. WhiteHat is different because we approach application security through the eyes of the attacker.

Through a combination of technology, more than a decade of intelligence metrics, and the judgment of real people, WhiteHat Security provides complete web security at a scale and accuracy unmatched in the industry. WhiteHat Sentinel, the company's flagship product line, currently manages tens of thousands of websites – including sites in highly regulated industries, such as top e-commerce, financial services, and Health Care companies.

For more information on WhiteHat Security, please visit www.whitehatsec.com.



WhiteHat Security, Inc. | 3970 Freedom Circle | Santa Clara, CA 95054 | 1.408.343.8300 | www.whitehatsec.com

©2015 WhiteHat Security, Inc. All rights reserved. WhiteHat Security and the WhiteHat Security logo are registered trademarks of WhiteHat Security, Inc.

All other trademarks are the property of their respective owners.

051915