The Business Case For Vulnerability Scanning

How and why vulnerability scanning plays such a foundational and impactful role in securing web apps and technology systems against malicious attack.

The majority of data breaches and malware attacks occur as a result of vulnerabilities that are patchable – and thus potentially preventable.

Attacks on technology infrastructure, computer systems, and websites are more common than ever, and bad actors have never been more successful nor more persistent at their malicious work. Hackers attack internet-connected computers 2,244 times a day, or every 39 seconds on average, according to the University of Maryland.[i]

That's just as true of the public sector as of the private sector. A report from the Georgia Institute of Technology says, "Low-intensity online nation-state conflicts become the rule, not the exception."[ii]

Unfortunately, such sheer persistence pays off. A survey from computer manufacturer Dell reports that nearly two-thirds (63%) of companies say their data has been potentially breached within the last 12 months.[iii]

And when they do happen, data breaches and other malware attacks are expensive. The average attack costs victims somewhere between $2.6 million and $3.92 million, according to studies from IBM and Accenture. That cost incorporates everything from fines to lost work time.[iv] [v]

In short, security threats pose a serious problem.

But most of them are also entirely preventable.

Sixty percent of successful breaches occur because an attacker exploited a vulnerability for which a patch was available but unapplied.[vi] "Detecting and prioritizing and getting vulnerabilities solved seems to be the most significant thing an organization can do [to prevent] getting breached," says Piero DePaoli, senior director of marketing at software and cloud computing firm ServiceNow.[vii]

Worse, according to a report issued by ServiceNow and the Ponemon Institute after surveying thousands of IT professionals worldwide, nearly two out of every five organizations *don't even scan for vulnerabilities*.

"That was one of the most surprising results," DePaoli says. "In order to detect vulnerabilities, you need to scan for them."

> **In order to detect vulnerabilities, you need to scan for them.**

Perhaps that's because those organizations are among the surprisingly large number that are overconfident in their security management processes.

Specifically, one study found a striking difference between where organizations *believe* they are in security posture and where they *actually* are. When the study's authors compared how security and IT executives described their vulnerability remediation processes (84% described such programs at their organizations as "mature") to what their programs actually *do*, they found that most of them had actually "only completed very basic tasks and were many stages away from a 'mature' program."[viii]

In other words, it's *very* easy to become over-confident about vulnerability management, even when a huge percentage of the security risk facing organizations and government agencies is potentially preventable.

Vulnerability scanning is the solution.

But what is vulnerability scanning? How does it work, and is it really worth the investment?

Those are the questions this paper will address.

What is vulnerability scanning?

A vulnerability is an exploitable flaw in software code or computing infrastructure that hackers can use to steal data, compromise performance, or take control of a system. Unfortunately, it's impossible to avoid such flaws; today's technology is just so complex that vulnerabilities are nearly impossible to prevent entirely. Knowing this, bad actors constantly probe software systems and websites looking for these vulnerabilities and develop sophisticated attacks that can take advantage of them. But software developers and security analysts do the same; and as soon as they discover vulnerabilities, they will develop and issue patches for them.

But where does that constant race leave organizations that just want to stay secure? How does a customer know if *their* system or website is host to a vulnerability primed for exploitation? The answer is vulnerability scanning.

Specifically, sophisticated security tools operated by IT experts will scan both infrastructure like servers as well as programming source code to look for known vulnerabilities, any potential problems with coding, and any malicious code that may have been inserted if the organization has already been hacked. If any vulnerabilities or problems are identified, they can then be fixed, e.g., by applying any appropriate security patches.

There are two major types of vulnerability scans: static and dynamic.

Static Application Security Testing (SAST)

Here, a tool scans source code for vulnerabilities and bugs, comparing against a database of all known vulnerabilities. The scan also compares the code against coding standards set by the coding platform.

As a result, the static scan can identify maliciously inserted code, any portions of the code that may allow hackers to manipulate it, or any other security weaknesses. Note that static code scans typically run in the production environment, rather than running scans during testing or staging new code.

- Performed in a non-runtime environment
- Focused on preventing vulnerabilities
- Has full access to the source code
- Separate tool typically required for each programming language

Dynamic Application Security Testing (DAST)

Here, a tool scans the code as it is executed, assessing the app as it works its way through various infrastructure like app or database servers, load balancers, etc.

The scan analyzes how those systems are performing against the app as built. One of the main purposes of dynamic code scanning is to test how robust servers are against any threats. As part of most dynamic code scans, the tools can mimic hacking process ("penetration testing").

- Performed while an app is in operation
- Focused on finding existing vulnerabilities
- Doesn't require access to source code
- Finds vulnerabilities that manifest at runtime

In general, static and dynamic testing complement each other. While there's definitely overlap in the kinds of vulnerabilities and issues they can identify, each would be incomplete without the other. For example, without access to source code, dynamic scans cannot identify code violations. Similarly, static scans cannot incorporate the impacts of operating environment, web server, or database content when an application is actually being executed. As a result, *both* types of scans are critical to comprehensively protect against security vulnerabilities.

What kinds of flaws and security issues can vulnerability scanning identify?

Modern day apps run in a dynamic environment that is in constant motion. Organizations never know when a new threat or weakness will be identified and exploited by bad actors. The core job of vulnerability scanning is thus to establish a robust first line of defense to keep servers and code as secure as possible against both known and emergent threats. The end result is less risk, less cost, more uptime, and better performance and productivity.

Scanning is also irreplaceable because the array of potential threats are mind-bogglingly vast, thanks to the sheer complexity of today's technology. Vulnerabilities can vary tremendously, and they are not all equally severe. For this reason, most scanning tools today do more than just spot vulnerabilities; they also help operators to triage potential risk by classifying vulnerability on a severity scale of 1 to 5. Vulnerabilities rated a 5 are the most critical: they can potentially fully compromise the system, up to and including total shutdown or loss of control.

This rating system enables operators to prioritize which vulnerabilities to resolve first, which is key to effective risk mitigation when security resources are often scarce and the task of security proprietary systems so monumental.

But what are the specific kinds of vulnerabilities that scanning can identify and help to remediate?

1 Code Violations

A code violation occurs when code does not meet established quality guidelines or standards for that programming language and the organization using it. Most code violations are relatively trivial – nuisances more than serious threats – and include issues like quotations not being used properly or comments not in the proper structure. However, code violations *can* potentially introduce three serious risks:

- Security vulnerabilities resulting from unintentional bugs or logic errors
- Performance issues (like lag) because code executes in suboptimal ways
- Opportunity costs because non-standard code is harder to maintain over time

Sometimes the problem is *incomplete* code. Since static scans are often run only when the application is updated, it's possible for new code still in development to sneak in by accident. With access to the full source code, static scans can analyze code quality and compare against established standards for the programming language to identify and protect against all of these types of issues.

2 Vulnerable JavaScript Libraries

Developers can use JavaScript libraries to ease and speed development – why reinvent the wheel if someone else has already done the work? – but if a web application ends up relying on an outdated library, it can introduce vulnerabilities to the web app. For example, it can make the app vulnerable to cross-site scripting, described below.

In other words, the web app relies on third-party code segments to operate, but if those segments become outdated and are compromised, they represent a security risk. According to a study from Northeastern University, at least 37% of websites use a JavaScript library with at least one known vulnerability; and due to some limitations of the study, the actual number may be even higher.[ix]

What makes this type of vulnerability so challenging is that it can change quickly; a web app can be built using a modern, up-to-date, secure library, only to have the library become compromised after the fact.

Static scans with access to source code can help to uncover issues related to vulnerable coding libraries.

3 Cross-site Request Forgery (CSRF)

As the name implies, this type of vulnerability refers to a forged request; the request *looks* trustworthy but is actually fraudulent.

Specifically, a web server might receive a request from a web browser whose credentials have been authenticated, but the request itself is unauthorized. A hacker might trick a user into signing into a website but then click on a link that sends a request to execute an action the user doesn't intend.

Note that this type of vulnerability doesn't rely on links exclusively; the malicious request can take many forms. But the end result is always the same: the web app erroneously treats a bad request as legitimate because it has authorized the user. These requests can also take many forms, including: making financial transfers, changing a user's login credentials, etc.

Static scans can help reveal if vulnerabilities in the code make this possible.

4 Cross-site Scripting (XSS)

With an XSS vulnerability, a hacker will add malicious scripts to a web page. In essence, web apps have to execute scripts in the user's local browser, so if the hacker can add a script to be run, they can trick the browser into running malicious code. So, if the user visits a website into which a hacker has injected a malicious script, the user's browser may execute the script (usually JavaScript).

XSS vulnerabilities are quite common, particularly when developers fail to validate user input on a web page or web app. In many cases, these vulnerabilities are just nuisances, but they can also create significant security risk if the data handled by the website is sensitive.

Dynamic testing is very helpful at spotting potential XSS vulnerabilities.

5 Misconfiguration issues

Misconfiguration issues are one of the most common types of security threats in the wild, with one survey citing it as the #1 cloud security threat in 2020.[x] In fact, a study by DivvyCloud of publicly reported data breaches found that misconfiguration issues cost companies an estimated *$5 trillion* in 2018 and 2019 combined.[xi] The issue is that computing systems are incredibly complex, with many different pieces that have to be individually configured in order to work properly for the intended application, particularly on the infrastructure side. Incorrect configurations, in turn, can create serious vulnerabilities. For instance, if insecure default configurations are not changed, or if temporarily incomplete configurations are never finished, they will expose the system to attack. And sometimes configurations just fall out of date; the app changes, but the hardware configuration doesn't. Regardless, they create significant risk. Indeed, a misconfiguration error was at the heart of the huge CapitalOne data breach in 2019.[xii] Dynamic scans can help reveal misconfiguration issues.

6 Other vulnerabilities

There are *many* types of vulnerabilities; those listed above are just a few examples of some of the most common, but they are also only the tip of the iceberg.

Remote code execution, SQL injection, file and directory traversal, and other vulnerabilities can all potentially pose a serious threat to the security of organizations that run websites, web apps, and software services.

And an outright vulnerability isn't even the only worry. Sometimes code that's just sloppy – more precisely, code that doesn't meet the standards set by its developers – can result in performance problems and possible security risks even without a known vulnerability. Vulnerability scanning can help to uncover all of these kinds of issues.

Why is vulnerability scanning so important? Vulnerabilities risk major losses: of time, money, opportunity, reputation, user confidence, and even the ability to conduct business at all.

The most severe issues literally have the power to shut an operation down.

Not all vulnerabilities are equally severe, but it only takes one to wreak havoc until the issue has been remediated, even as it incurs huge financial losses and sometimes incalculable data losses.

But even less severe vulnerabilities can still be damaging, resulting in anything from exfiltration of data to performance slowdowns that frustrate users and expose them to unnecessary risk.

Worse, these vulnerabilities, when successfully exploited, can have a multitude of secondary impacts that then hit like aftershocks. For example, the loss of reputation and customer confidence can have an even more wide-ranging impact on an organizations ability to conduct business than the malware attack itself. Loss of trust can impede organization's from fulfilling their missions because their users or customers simply refuse to work with them.

Then, there are potential investigations and audits resulting from regulatory non-compliance, along with any associated penalties and legal fees.

Then, there's yet another layer of impacts: the cost and time required to remediate the problem after the fact. It's *always* more expensive, more labor-intensive, and more time-consuming to fix problems after they have occurred. The traditional IT approach has always been based on a reactive break-fix mentality (only *after* there's a break do you/can you fix it). Then there's operational and labor expense, as well as opportunity cost due to resources being drawn away from other tasks in favor of fixing the problem that should have been prevented to begin with.

This brings us back to the importance of vulnerability scans as a first line of defense.

The problems described above can compound if an organization doesn't have enough qualified security professionals – and there is currently a shortage in this area. Cybersecurity is a hard-to-fill job role; one study found that cybersecurity professionals effectively have a 0% unemployment rate. There just aren't enough knowledgeable experts to answer all the demand in the marketplace.[xiii]

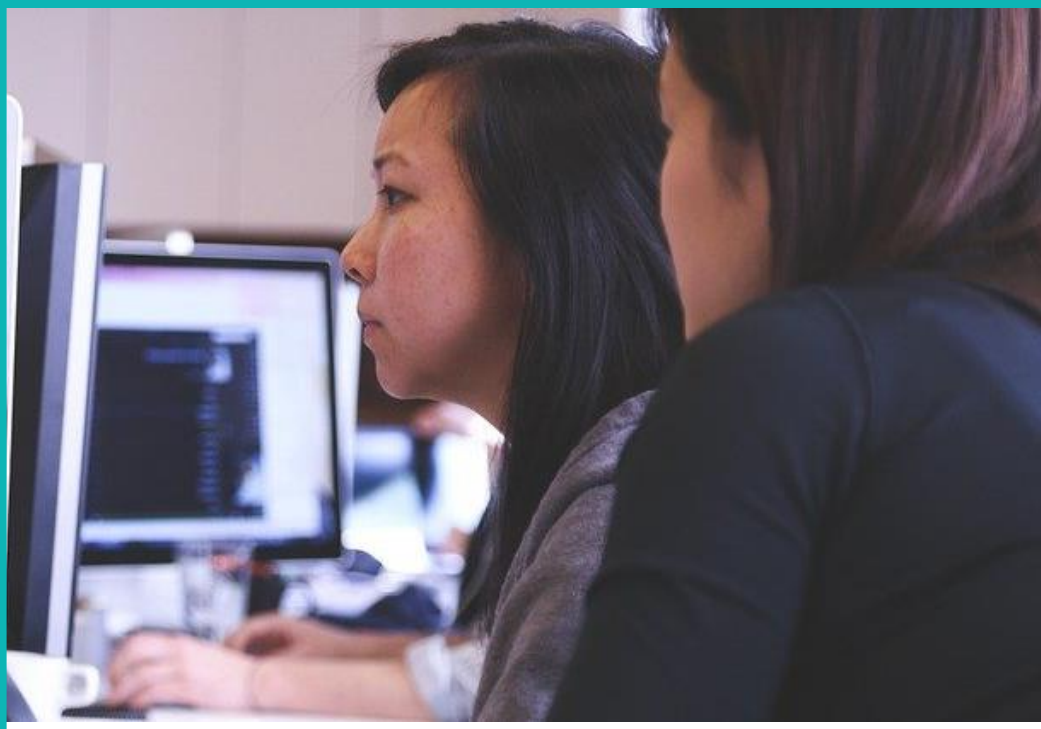
Vulnerability scanning can secure organizations against all of the 60% of attacks and breaches that result from patchable-but-unpatched vulnerabilities and thus stop the multilayered shockwaves of damages described above that can result from a security incident. They can also help identify potential issues that may not be patchable yet but allow organizations to take precautionary steps.

Even better, vulnerability scanning is a largely automated process that reduces reliance on a large security staff; and, because the scans can help to triage and prioritize the vulnerabilities, organizations don't have to manually comb through every potential vulnerability one at a time to determine severity and fix them. That saves time, money, and labor. Most organizations simply wouldn't have a security team or even security vendor expansive enough to handle such an enormous and time-intensive task.

Vulnerability scanning is the foundational, critical cornerstone needed to answer these issues, enabling organizations to take preventive, proactive steps to secure themselves and their users and – along the way – save themselves huge losses and serve their users with an excellent, reliable experience.

232 Healthcare Portal

Practical Application: How Do Static and Dynamic Scans Work in Practice?



The Challenge

The original code underlying the 232 Healthcare Portal needed to be standardized, updated, and secured. When PSL first started scanning, they found dozens of vulnerabilities every month.

The Solution

PSL performed static scans of source code in non-runtime environments via SonarQube and dynamic scans via Qualys and then remediated any vulnerabilities or issues discovered.

The Outcome

Three years after the start of the project, scans turn up only a few or no issues. Web app performance has improved, and HUD and ORCF have greater peace of mind that the Portal is secure and functional.

Run and managed by the U.S. Department of Housing and Urban Development (HUD), the 232 Healthcare Portal plays a pivotal role in HUD's ability to fulfill its mission.

Administered by the Office of Residential Care Facilities (ORCF) for HUD, the portal is the primary vehicle by which Section 232 insured healthcare facilities submit quarterly operator financial information and receive loan applications and loan closing documents from FHA lenders. In short, it is vital to the conduct of critical and sensitive HUD business and manages both electronic data and physical documents through a single point of contact, making it a potentially attractive target for hackers.

As a result, while the 232 Healthcare Portal eases the administration of these programs for the ORCF, it also introduces potential risk. The 232 Healthcare Portal is a browser-based web application maintained, hosted in the Amazon Web Services (AWS) FEDRAMP-certified cloud environment. Peniel Solutions (PSL) was thus contracted to prepare a business service application Static and Dynamic security plan.

PSL's scope of work included:

- Run dynamic scans while the web app remains in operation and static on code in a non-runtime environment.
- Through those scans, identify any vulnerabilities or weaknesses.
- Resolve those issues as soon as possible.
- Provide supporting documents to stakeholders.

The Challenge

Outdated, inconsistent, non-standard code doesn't just introduce potential security vulnerabilities, it also impacts performance and can make the entire app more difficult to maintain over time. It's especially challenging to manage when the code was originally built by a separate team, only to be passed to a new vendor. The PSL team inherited the source code for the 232 Healthcare Portal, and it quickly became clear that a lot of the code didn't meet current coding standards for using .NET language.

In fact, when PSL first started running scans, they would catch at least 40 or 50 different kinds of vulnerabilities on a monthly basis.

The Solution

The best way to identify potential vulnerabilities and risks is to actively scan for them.

For its static scans of source code in a non-runtime environment, PSL uses SonarQube, a highly rated open-source tool that can comprehensively detect any errors or coding quality issues and make recommendations for improving code formatting. PSL runs static code scans any time changes are made to the code.

For its dynamic scans of all the servers for the portal (eight in total), PSL uses Qualys, an industry-leading and FEDRAMP-certified scanning tool. The dynamic code scans run on a fixed monthly scheduled.

Based on SonarQube's and Qualys' output, PSL can then implement or recommend any appropriate code changes. That might include removing any malicious code, changing any piece of code that could render the application vulnerable to attack, updating code to meet .NET coding standards, or applying any new patches issued by Microsoft or other vendors.

Any time the scans turn up any performance-impacting bugs, PSL creates a trouble-ticket with the appropriate team at HUD, and PSL will collaborate with them to re-scan after the bugs have been fixed.

The Results

1: Vulnerability Control

Over the three years that PSL has been running scans, the number of vulnerabilities discovered through dynamic scans have plummeted from dozens monthly to no more than four or five per month. Most of those are security patches or updates that need to be applied. For example, in a recent month, all vulnerabilities found were directly related to security updates just released by Microsoft that still needed to be applied.

Meanwhile, the number of vulnerabilities discovered on the static coding side have plummeted from close to 100 (most of them medium-severity threats) at the initial scans to zero vulnerabilities discovered in three recent consecutive months.

2: Performance Benefits

Especially for the static code scans, once PSL has remediated identified issues (coding standards met and bugs fixed), portal performance has improved.

3: Peace of Mind

In the three years that PSL has overseen the vulnerability scanning project, the 232 Portal has had no incidents with compromised code. Knowing that all servers are being scanned, vulnerabilities fixed, and all security updates applied means peace of mind and allows ORCF and HUD leaders to focus on mission-critical duties.

4: Security Planning

Once patches have been applied and vulnerabilities fixed, PSL generate comprehensive reports. This enables PSL and HUD's security office to collaboratively produce and maintain a working system security plan for the 232 Portal detailing all controls and measures put in place to secure the application. HUD stakeholders have historically been heavily engaged in this process, so the robust reporting allows them to stay current on vulnerabilities identified.

Peniel Solutions is a recognized leader in providing complete, end-to-end technology enabled business solutions. By fostering an environment of continuous education, we can deliver the latest technical advancements and trends in the areas of Cybersecurity, Data, Development, Cloud and Advisory Services. We seek to understand the unique needs and goals of our clients and with that information we work to improve the performance of our client's critical business systems and processes.

Cloud Solutions

Capitalizing on AWS's GovCloud, PSL's partner status enables us to support customers with Cloud Application Migration activities, application maintenance and DevOps services.

Security

PSL uses best-in-class solutions to support Identity & Access Management, Secured Remote and Local Access, Protected Data Transmission, Secured Cloud Services while Protecting Identity and Preventing Theft.

Big Data Analytics

By leveraging TransAccess, Big Data and related technologies our customers are empowered to capture, analyze, and manage their information real time and from anywhere they are located.

Software Development

Based on ISO and CMMI Development-Certified processes, PSL delivers mobile and traditional software solutions of the highest-quality based on scrum, agile and DevOps methodologies.

Advisory Services

We help customers navigate evolving technology and digital transformation with industry-leading expertise that ensures regulatory compliance without sacrificing effectiveness.

Corporate Office
3885 Crestwood Pkwy, Ste. 275
Duluth, Georgia 30096

Washington, D.C. Office
1001 G. Street, NW, Suite 800
Washington, D.C. 20001

(866) 878-2490
info@penielsolutions.com
www.penielsolutions.com

References

[i] https://eng.umd.edu/news/story/study-hackers-attack-every-39-seconds
[ii] http://www.gtcybersecuritysummit.com/2015Report.pdf
[iii] https://www.dellemc.com/en-us/collaterals/unauth/analyst-reports/solutions/dell-bios-security-the-next-frontier-for-endpoint-protection.pdf
[iv] https://www.accenture.com/us-en/insights/security/cost-cybercrime-study
[v] https://www.ibm.com/security/data-breach
[vi] https://securityboulevard.com/2019/10/60-of-breaches-in-2019-involved-unpatched-vulnerabilities/
[vii] https://www.darkreading.com/vulnerabilities---threats/unpatched-vulnerabilities-the-source-of-most-data-breaches/d/d-id/1331465
[viii] https://www.zdnet.com/article/vulcan-cyber-study-finds-serious-problems-with-vulnerability-management/
[ix] https://www.tomshardware.com/news/websites-outdated-insecure-javascript-libraries,33885.html
[x] https://www.infosecurity-magazine.com/news/misconfiguration-error-cloud/
[xi] https://www.techrepublic.com/article/cloud-misconfigurations-cost-companies-nearly-5-trillion/
[xii] https://krebsonsecurity.com/2019/08/what-we-can-learn-from-the-capital-one-hack/
[xiii] https://www.ciodive.com/news/0-unemployment-rate-and-5-other-numbers-you-need-to-know-about-cybersecuri/566779/