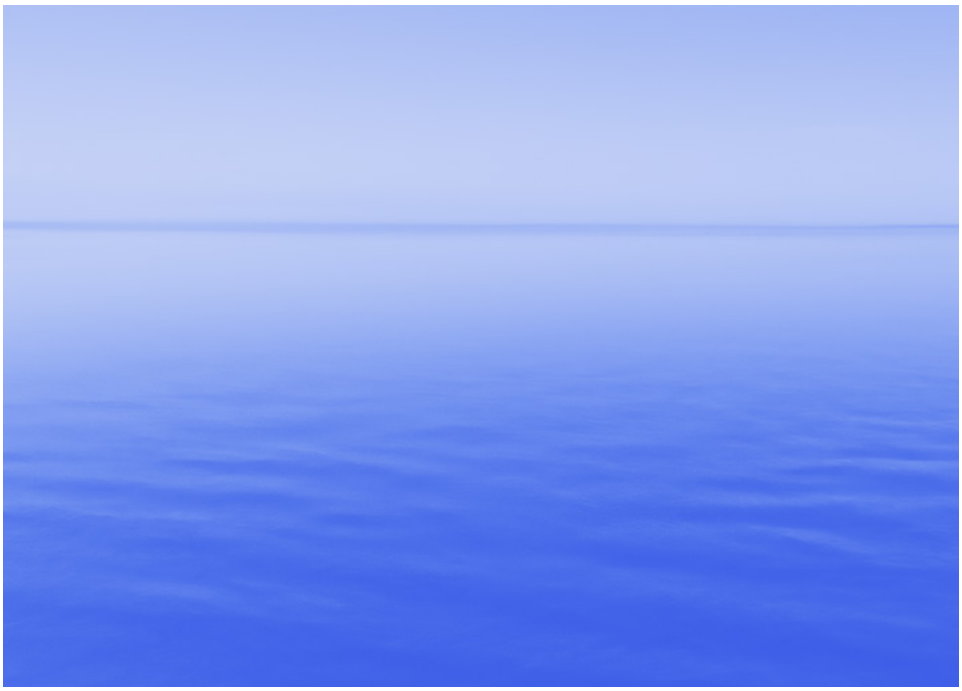
Clearwater

Whitepaper

Why Secure Software Development is Critical for Healthcare Now and in the Future

Table of Contents



Introduction

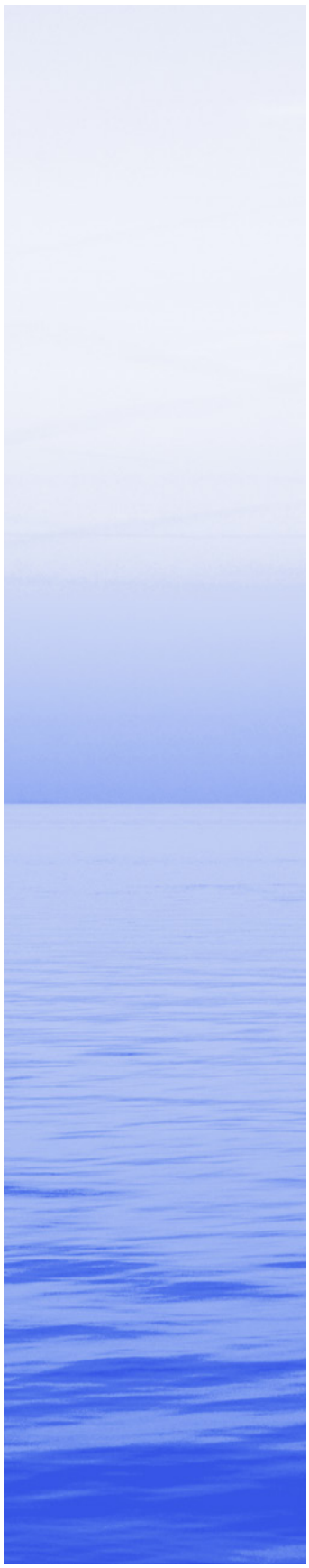
As web application attacks continue to rise, today building security into web applications is more important than ever. In December of 2020, Imperva Research Labs monitored a 51% increase in web application attacks on healthcare targets. Imperva data shows the healthcare industry experienced 187 million web application attacks per month globally, on average, or roughly 498 attacks per organization each month. There are no indications this trend won't continue.

Unfortunately, that is because these attacks are often successful and lucrative for threat actors while resulting in dire consequences for organizations and patients. For example, in August of 2020, a threat actor gained access to the web-based appointment scheduling application managed by Luxottica. The hacker gained access to records of 829,454 patients. In January of 2021, a breach of the Florida Healthy Kids Corp website operated by a vendor, exposed data on 3.5M covered individuals. These two breaches likely generated millions of dollars for the perpetrators, who no doubt sold this information on the dark web or used it for their own fraudulent purposes.

Despite these very public breaches and the subsequent financial, regulatory and reputational impact, healthcare organizations continue to deploy insecure web applications. In so doing, they are needlessly placing confidential patient information at risk along with their organizations' capital, reputation and revenue. In the case of third-party software and service providers, they are also putting at risk the capital, reputation and revenue of their customers, who are increasing their efforts to understand and manage this vendor risk.

Why are web apps targeted?

Some security experts estimate most, if not all, web applications today have some sort of vulnerability, coding issue, misconfiguration, or other security concern that make them susceptible to exploit. When a web application is deployed on the Internet, these vulnerabilities are offered up to threat actors operating everywhere in the world. The value of electronic health, financial, and proprietary research information is causing an increasing number of these threat actors to target these vulnerabilities specifically and the healthcare industry more generally.



There are many sources of vulnerabilities within web applications. A great resource for identifying common vulnerabilities is the industry standard OWASP Top 10.

OWASP's current Top 10 include:

- Injection
- Broken Authentication
- Sensitive Data Exposure
- XML External Entities (XXE)
- Broken Access Control
- Security Misconfiguration
- Cross-Site Scription (XSS)
- Insecure Deserialization
- Using Components with Known Vulnerabilities
- Insufficient Logging and Monitoring

When web applications are publicly accessible through the Internet, any threat actor with Internet access anywhere in the world can search for and attempt to exploit these vulnerabilities.

Threat actors come in all shapes, sizes and nationalities. They range in sophistication from kids experimenting with automated tools to professional criminal organizations and nation states. Over the last year, it has become clear that healthcare organizations are targets for all of these threat actors and need to plan accordingly.

Unlike data from breaches of information in other industries, breached healthcare data often contains far more of an individual's personally identifiable information. The availability of this rich information provides additional avenues of fraud and subsequently value for criminal actors. As a result, according to a Trustwave report, a healthcare data record may be valued at up to $250 per record on the black market, compared to $5.40 for the next highest value record (a payment card).

It's not just personal health information that interests threat actors. The COVID pandemic and organizations doing research on treatments have found themselves the target of the most sophisticated of threat actors as nation states look to gain an edge. China, in particular, seems to be conducting this type of activity. In July 2020, the US Department of Justice indicted two individuals working with the Guangdong

State Security Department (GSSD) of the Ministry of State Security. Another US government report found a Russian hacking group known as Cozy Bear also targeting US COVID-19 research firms.

Why don't we just fix the applications?

This might cause one to ask, "why are there so many issues with apps and web services and why don't we just fix them?" The answer is perhaps found within the complexity of most modern applications and the speed and volume at which they're thought-up, developed, and deployed.

Let's take a look at an example of a website as it relates to application security. Websites often use multiple frameworks, open-source libraries, and plug-ins to facilitate production and deployment speed. Developers assume these components are secure but often don't really know.

Even for web development teams that build sites with custom code, you'll often find they have a hybrid approach where they may use a website framework, for example WordPress, and open-source libraries and plug-ins to implement specific components, like an online storefront, but then they'll write their own code for other components. In this example, the framework, plug-ins, libraries, and custom code are all potential attack vectors.

Even when developers use mostly custom code, we find problems. To speed development, they'll use already-written code segments. If there is a security vulnerability in one of these segments, and that code is reused multiple times, the vulnerability is replicated throughout an application or applications resulting in significant security vulnerabilities that can be time consuming to remediate.

These problems continue to exist because many development teams lack the skilled professionals, resources, tools, and experience to get insight into every layer of every application or web service. Right now, developers outnumber security professionals 100 to 1, and few development teams have integrated security practices across the entire software development cycle.

There's also a general misconception that if developers follow a specific set of standards during software development, the software is secure. But that's not effective, and once deployed, security teams are left to figure out and resolve application security issues.

The challenge for security teams?

One of the key challenges is the often-siloed approach to software development and security, where one team builds the application, and another is responsible for figuring out all the security issues and developing a plan to fix them.

Security teams understand it's important to address vulnerabilities, patching, misconfiguration, and coding issues as efficiently as possible, but the growing adoption of new technologies, applications, and the Cloud create so many issues, they struggle to keep up.

Often, they have so much vulnerability data, they have no idea where to start first, and if a headline-making security issue pops up, they're often called to drop everything and respond, even if there's not a specific, impending threat to their organization. Once behind, every new vulnerability assessment, pen test, and risk analysis adds to the list and makes it ever-more challenging for the team.

That's not to say developers aren't taking steps toward security themselves. Good teams frequently review and test their applications during development, but not many have the abilities or tools to dig into all the layers to find all issues before deployment. Once deployed, those already stretched development teams turn their focus to new apps or other improvements.

Integrating security into the system development lifecycle

One way to reduce the number of vulnerabilities and cost of remediation is by introducing security into the system development lifecycle SDLC or building security into applications from the ground up. This approach reduces the volume of vulnerabilities that security teams are left to manage when applications are deployed as well as reducing the cost of remediation.

There are various development methodologies teams use when developing an application. Three common methods we see are:

Waterfall Methodology

The waterfall methodology is decreasing in use across many organizations because it's just not as productive as other methodologies in our current, fast-paced development and deployment environments. In this method, developers design and develop code for a complete application and then test that design.

Agile methodology

Developers who use the agile methodology generally develop software in short sprints or phases that can last anywhere from a week to a month. In each sprint, the team reviews the development plan, for example, a new feature to add to the application, and will work toward that goal during that sprint. If the development plan calls for several new features, then with an agile approach, the team would plan for and then move into the next sprint for the next feature.

DevOps methodology

DevOps approaches to software and application development are becoming increasingly common. It's similar to working within a cloud environment where you can quickly test, release, and monitor your new solution. This method helps ensure that all of the required steps and processes take place in a continuous manner and sometimes can be effectively automated, like making a change within a code repository, and then that change spans across the development and production environments simultaneously. This method limits the ability for code changes to take place in between and adds a security layer to coding interval deployment processes.

Regardless of the methodology you follow, it is important to embed security into the development lifecycle. Whenever you come up with an idea for a new application or software, or any time you are considering adding new features or attributes, your development team should plan to address known security risks (for example, the OWASP Top 10), and then build in security mitigation and remediation measures throughout the architecture:

- Are you routinely checking your code throughout development?
- Are you looking for known coding issues?
- Are you ensuring your team is fixing coding mistakes before deployment?
- Security testing, like penetration tests, as well as code analysis and validation can help you discover these issues so you can address them before you complete application production.

An important part of this process also involves risk analysis. By doing risk analysis as part of the SDLC you can review risks, determine probability of impact, and evaluate those risks against your organization's risk threshold. This helps in understanding what additional safeguards or controls you should build into the application.

Along with risk analysis, it is beneficial to deploy threat modeling for security. Essentially, during threat modeling we create an architectural diagram of the software, then look at the environments where it will be deployed, along with connected components and interactions. From here, one can better understand threats and security risks.

The Microsoft STRIDE Threat Model, for example, includes six categories that define the risks you can classify the interactions between different entities into, such as:

- Spoofing: Adversary pretends to be someone else
- Tampering: Data modification without authorization
- Repudiation: Denying one did something
- Information Disclosure: Unauthorized access to information
- Denial of service: Attacks designed to prevent service
- Elevation of privileges: Gaining more rights than one is authorized

By addressing these threats throughout the software development lifecycle, you can be well on your way to more secure applications.

Bringing development, security and operations together

Whether you're a healthcare organization developing your own applications or you're working with a software development company or other third-party service, you need confidence that your application meets all of your privacy, compliance and security standards.

This is where adopting a security roadmap can help you on your security journey—giving you complete end-to-end visibility on all your new and modified applications and software.

Two core areas for attention when implementing your security roadmap include:

Compliance

- Healthcare software should be HIPAA compliant
- Needs FDA approval
- Needs to follow strong security validation and verification frameworks
- Meets vendor security requirements outlined during RFP approval

Streamlined security audit

- Complete visibility of software development process
- Managed security process and automation
- Knowledge training

Your application security roadmap helps clarify which teams are responsible for which activities and encourages collaboration across teams throughout the software development lifecycle—from conception to deployment and beyond.

Your roadmap builds that collaboration and also adds value to your business by creating streamlined processes for development and security. This reduces the cost of remediation and the risk of embarrassing flaws after deployment.

Cross-collaboration is key. When you have disparate teams working toward disparate goals, it's challenging to get complete visibility into all of your security issues. You want to build development teams that think about security processes throughout the SDLC and encourage security team involvement in new software and functionality before deployment.

Build continuous and autonomous processes between DevOps and security with an emphasis on risk analysis and risk reduction. This can be done by integrating security into traditional development methodologies or adding security into DevOps known as DevSecOps.

Here are a few key recommendations to help you with this process, listed in order of effectiveness:

1. Code review
2. Architectural risk analysis
3. Penetration testing
4. Risk-based security tests
5. Abuse cases
6. Security requirements
7. Security operations

Software security pillars

When planning for secure application development, three critical security practices you can execute as part of an application security roadmap:

Risk management

- Define all possible threats
- Perform a risk analysis at the architecture level and manage risks through the secure SDLC

Security touchpoints

- Employ security touchpoint at every step of SDLC

Security knowledge

- Maintain a knowledge repository for your code base-related security vulnerabilities
- Continuously add to the knowledge repository so it becomes an iterative process where security standards always grow
- Implement security bugs checklist
- Integrate security in large-scale code base

Finally, if you need help building your healthcare application security program, consider working closely with an advisor that understands security protocols for application development and the healthcare compliance landscape. With a Clearwater partnership, we can help you:

- Stop the bleeding
  - Identify problematic areas and add security
- Establish a foundation
  - Build foundations for future initiatives (such as change control programs, transparency, traceability, and common knowledgebase)
  - Automate software development and testing process through DevSecOps
- Address compliance concerns

- Highlight and document developed organizational capabilities
- Align with compliance standards for healthcare software development
- Bring value by streamlining security process

Clearwater is the leading provider of Enterprise Cyber Risk Management and HIPAA compliance software and consulting services for the healthcare industry. Our solutions enable organizations to gain enterprise-wide visibility into cybersecurity risks and more effectively prioritize and manage them, ensuring compliance with industry regulations. Clearwater's suite of IRM|Pro® software products and consulting services help healthcare organizations to avoid preventable breaches, protect patients and their data, and meet OCR's expectations, while optimizing cybersecurity investments. More than 400 healthcare organizations, including 64 of the nation's largest health systems and a large universe of business associates that serve the industry, trust Clearwater to meet their information security needs.

Have questions regarding this white paper? Engage with Clearwater for a specific discussion about your organization's approach.

- ClearwaterSecurity.com/Contact