A Strategic Approach to Web Application Security

Extending security across the entire software development lifecycle

The problem: websites are the new security perimeter

Today, almost every enterprise conducts business online. As the applications that run online businesses spread out over technologies and platforms, the security risks also increase. No company is immune to attack. In 2012 alone, there were more than 800 reported hacking incidents, and 70% of those were perpetrated through web application flaws. The web is the new perimeter for enterprise IT security, and it is not nearly as easy to lock down as a network.

Most organizations that develop custom software struggle with managing these new risks. That's because application security is a continually evolving field—security standards and rules must evolve as fast as these technologies and the ever changing array of attacks on them. And yet, solutions in the past focused primarily on reactive measures.

Due to the unprecedented increase in the scale and intensity of sophisticated web application attacks, organizations are scrambling to re-architect their development processes to make security a major component of each phase.

Many approaches for testing security after the application has been built have proven costly and caused critical deployment deadlines to be missed. Historically, there was no economical way to continuously test source code for security issues during the development process. Therefore, organizations were forced to consider security only near the end of the development process, when the web application was nearly complete. Then, applications were typically tested by either expensive consultants or tools requiring a high level of security expertise to use them effectively.

Because no solution existed that could provide the continuous security guidance to developers, security was forced to the end of the development cycle. WhiteHat Security has now addressed this issue with Sentinel Source, a radically different approach to application security. Sentinel Source provides security analysis to developers from the moment the first line of code is written, which assures that risk and development costs are reduced over time.

A new security paradigm

Security activities historically revolved only around the most critical web applications. However, recent attacks have proven that attackers can and will target non-critical applications. In today's landscape, attackers no longer need to breach your most critical applications in order to steal valuable information that will cause financial losses or damage your company's reputation. Instead, we now know that in many high-profile breaches the attackers can gain a beachhead by targeting less visible, and therefore less secure, applications such as no longer used or subsidiary websites. From these sites, hackers can now compromise your underlying security infrastructure through attacks such as SQL Injection. The notion that you have full web security coverage when you protect only securing flagship applications is no longer sufficient.

Security throughout the software development life cycle (SDLC)

Web application development remains a relatively new domain. As such, formal processes and best practices for developing web software are still being defined.

Currently most enterprise organizations follow a set of standard steps, which define each phase of software creation. These phases are collectively referred to as the SDLC.

Phases of software development:

- Training
- Requirements
- Design
- Implementation
- Quality assurance
- Production

It became apparent that security was a serious issue and also the most vital missing piece in the development process. In fact, security assurance in the past was relegated to the QA phase of development when it was considered at all. Now, however, forward-thinking organizations are adding security activities to every phase of the SDLC in order to discover flaws earlier and to significantly increase the security of their production applications.

Typical security activities in each phase of the SDLC are as follows:

Training

Everyone involved in web application development should be provided basic security training. Scalability and repeatability are critical aspects of effective security training programs. Traditionally in the past, security training was done live—but as development teams grow, e-learning and computer based training (CBT) are a great way to ensure consistency, reproducibility and regularity of training across your entire IT workforce.

Requirements

As software requirements are defined, the corresponding security requirements should also be defined. For example, if sensitive customer data will be collected and stored, requirements on how the data should be encrypted, both in transit and at rest, should also be established as a requirement.

Design

Once the application requirements are captured, an architecture is created that should correspond to the software requirements. At this stage of development, necessary security controls should also be identified and allotted as part of the application.

Implementation

After requirements have been determined and an architectural design is in place, constructing the software begins. Ideally, developers should receive security feedback while they are coding. This feedback should be done as early and often as possible. Because this phase is often the most labor-intensive, continuously running automated security assessments allows a developer to address issues in near-real time. Otherwise, organizations will develop applications built on faulty code, with security an afterthought, rather than being designed at the start.

Quality assurance

New code needs to be tested before it goes into a production environment to ensure that the code behaves as expected. While most organizations currently test for functional requirements, more are also beginning to test for security requirements in this phase as well.

Production

In the deployment phase, continual testing is vital to maintain security assurance and to protect the application where most attacks occur. Also, updates to production applications can introduce new flaws. Therefore, any code updates should also be subjected to source, QA and production testing.

Making security visible is the key to maximum security

Vulnerabilities are a direct result of source code complexity. Therefore, in order to fully understand your application vulnerabilities and the overall enterprise security posture of your websites, you need to secure more of your source code throughout both the development and production processes.

It's easy to insist, "Why can't developers code securely," but the fact is that web application security is an extremely complicated problem. And with millions of applications and lines of code already existing online, web security is now more than ever an arcane knowledge-set that is continually evolving into new parameters.

Application code may contain a large number of obscure and hard-to-see issues that are further obscured by the ever-increasing complexity of modern web applications. So it's natural that some security issues are nearly impossible to identify in a timely way by developers, security teams and stakeholders.

In the past, most organizations depended on a few highly trained individuals who investigated code for security issues. The problem with this method was scalability: It was virtually impossible for security consultants—or even first generation tools—to review source code continuously.

And without continuous visibility into the security status of its applications, it's very difficult for an enterprise to manage risk effectively.

Source Code vs. Binary Analysis

Why is it better to assess source code than binary code?

Here's how WhiteHat Security views these two approaches.

Analyzing the source code is absolutely the best approach (provided that the source code is available). Because developers create the source code, any vulnerabilities are embedded within it, and the code can then be fixed to address security issues that arise during the development process.

Analyzing source code also allows analysis of the exact vulnerable method calls being made, so that the remediation advice is both pertinent and actionable.

WhiteHat Sentinel Source analyzes source code, providing developers specific remediation guidance.

On the other hand, analyzing source code is better than binary analysis.

Binary analysis perform assessments on compiled code that is not decipherable by developers. For this reason, with binary analysis, it may be difficult to distinguish the code written by internal developers and the code inherent to the platform. This shortcoming results in non-actionable and confusing reports for the developers. However, binary analysis can be a suitable last resort in those situations where source code is unavailable.

The challenge: enterprise SAST solutions that fail to work

Static Application Security Testing (SAST) has continued to play an important security role, building different generations on top of one another. Below is a high-level overview of those generations performing web application analysis:

First generation: standalone tools

The first generation of SAST tools consisted of standalone applications that were installed and run either within an organization or by external consultants. These tools were faster and more comprehensive than a consultant's manual assessment, but these tools still required an expert, or team of experts, to examine the findings, which typically included a high percentage of false positives. Essentially, each finding had to be verified by security experts, or the developers had to determine which findings really needed to be addressed.

Because of these limitations, first generation tools are nearly impossible to scale to meet the needs of a modern enterprise. Furthermore, most enterprises cannot afford either the time or human resources to operate these tools.

Second generation: cloud-based SAST

In an effort to address the shortcomings of early SAST products, a new class of vendors developed cloud-based SAST solutions. Organizations were advised to compile their web applications with special debugging flags, zip up, and submit their application to the cloud. The idea is that testing done in the cloud reduces the need for manual interaction.

This generation of SAST solutions yielded improvements over the previous models; however, there were still several domains in which this solution was not sufficient for actual enterprise use.

These second generation solutions required an organization to hand over its entire source code for analysis. If the organization considers source code to be its intellectual property, that presents a serious problem in having the code analyzed for security vulnerabilities. That's because even though the code is "only" in binary format, it can be easily decompiled to source code within seconds.

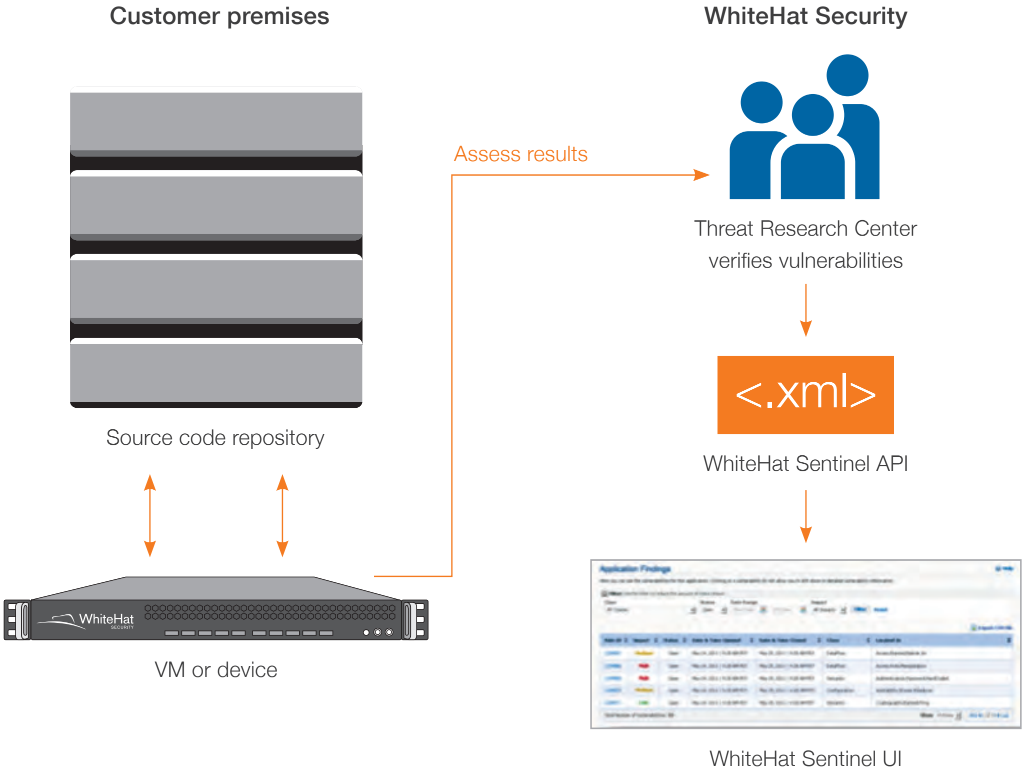
In addition, second generation analysis is not continuous, so the application had to be nearly complete before analysis would be useful. As a result, the security advice reached the developer much too late in the development cycle. Not to mention the effort it will take for developers to configure the code and ship it off to their vendor for testing.

Also, in an effort to find a workaround for multi-language support, some cloud-based security vendors take all binaries, reduce them to a common format and then scan the common format. This method dramatically reduces the depth of the scan. These types of scans only find the "lowest hanging fruit." So they are not a practical solution for time-strapped enterprises dealing with an expanding array of threats.

The solution: WhiteHat Sentinel Source

WhiteHat Security, a pioneer in delivering easy-to-use and scalable website security solutions for production websites, includes WhiteHat Sentinel Source, a SaaS-based annual subscription service for Static Application Security Testing of web applications. Sentinel Source was designed to address the issues that have plagued both first-generation and second-generation tools—creating a new generation truly designed from the ground up to meet the security needs of modern enterprises.

Sentinel Source provides in-depth, high quality assessments, which are free from false-positives, while respecting the privacy and chain of custody of an organization's most important piece of intellectual property—source code.



Why WhiteHat Sentinel Source?

WhiteHat Sentinel Source was architected and designed around four principal areas.

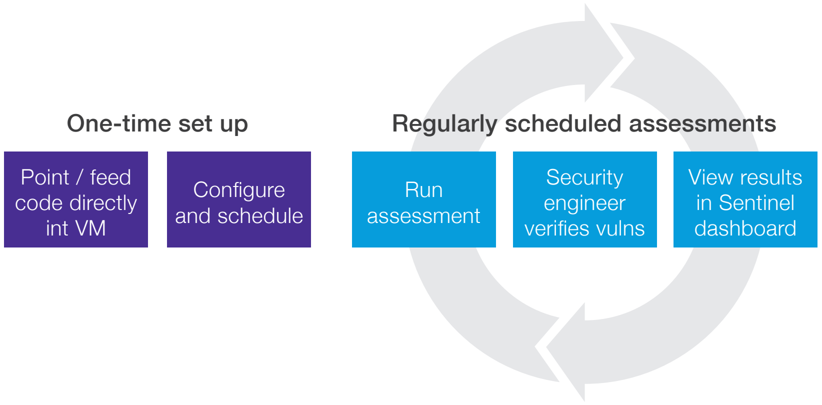
Daily scheduled and on-demand assessments
Developers need to know as quickly as possible if their code has introduced a security flaw. Sentinel Source runs every night, identifying issues as quickly as possible so developers can fix the problems inline and avoid building upon faulty code.

Verified results
Every reported finding reported is verified by a security engineer in the WhiteHat Security Threat Research Center (TRC). This ensures that security reports contain solid, accurate and actionable instructions for developers, as well as detailed advice for remediation.

Protection of intellectual property
Sentinel Source runs only within the organization’s network, ensuring that its source code tree always remains within the organization. No uploading of fully compiled source code or binaries outside of the company is necessary.

Ease of use
The Sentinel Source appliance that is deployed with the service is very easy to configure. With all software contained within the appliance box, there’s no need to install anything on the developers’ machines. All Sentinel Source results are available via the common Sentinel web interface, and the Sentinel Open XML API enables easy integration into Archer, Jira and other applications.

How Sentinel Source Works


One-time set up
Regularly scheduled assessments
Point / feed code directly int VM
Configure and schedule
Run assessment
Security engineer verifies vulns
View results in Sentinel dashboard

Static Analysis vs. Dynamic Analysis

Why should an enterprise use SAST and DAST together?

WhiteHat Security is a pioneer in application security for production and pre-production websites. Now, that knowledge and experience is being applied to the SAST arena. It is vital is to understand that application security is not a single solution problem – but rather a strategic initiative. The threat landscape is constantly changing. With hundreds of millions of websites already launched, the first step an organization should take is to assess the security of their production sites, their most vulnerable point of attack. Once an organization's security posture has been established, static analysis should be implemented to assess new applications during the development process or code changes to existing applications. The goal is to create a security process that tracks an application throughout the SDLC.

Each type of testing has its own advantages. Performing nightly static application security testing on source code will greatly assist developers to ensure that they code securely throughout the development process. Vulnerabilities such as SQL Injection, Command Injection, Improper and/or Custom Cryptography, and a host of other issues that occur primarily on the server are easy for SAST to find and may be eliminated early in the process.

However, certain other types of vulnerabilities can only be seen once the code is live on the web. Therefore dynamic scanning of production applications is equally critical. Issues such as cross-site request forgery, business logic flaws ,and some forms of cross-site scripting are much more easily identified by a DAST solution such as the Sentinel production family. And, with a web application firewall, enterprises can automatically mitigate most vulnerabilities while awaiting developer remediation.

Using SAST and DAST technologies at the appropriate stage during the SDLC is necessary to create a strategic enterprise application security program.

Conclusion

With the advent of Sentinel Source, we stand on the brink of a complete revolution in secure web development. By giving developers continuous feedback during the entire development process, by tracking in near real-time when vulnerabilities are introduced—and when they are remediated—we can make major headway towards eradicating the current global epidemic of website hacking at the beginning of the development process.

About WhiteHat Security

WhiteHat Security has been in the business of securing web applications for 15 years. Combining advanced technology with the expertise of its global Threat Research Center (TRC) team, WhiteHat delivers application security solutions that reduce risk, reduce cost and accelerate the deployment of secure applications and web sites. The company's flagship product, WhiteHat Sentinel, is a software-as-a-service platform providing dynamic application security testing (DAST), static application security testing (SAST), and mobile application security assessments. The company is headquartered in Santa Clara, Calif., with regional offices across the U.S. and Europe. 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