5 Reasons Why SAST + DAST with Fortify Makes Sense

The combination of static (SAST) and dynamic (DAST) application security testing methodologies provides a more comprehensive view of an application's risk posture. Here are 5 reasons why SAST + DAST with Fortify makes sense.

The combination of static (SAST) and dynamic (DAST) application security testing methodologies provides a more comprehensive view of an application's risk posture. Static analysis tools give thorough feedback early in the SDLC, while dynamic analysis tools can give security teams a quick win by immediately discovering exploitable vulnerabilities in either production or pre-production environments. Testing in both ways yields the most complete view of the risk posed by weaknesses and vulnerabilities within the application.

1.

A unified taxonomy across testing methods enables a complete view of vulnerabilities. The Fortify Software Security Research (SSR) by OpenText™ group is a team of experts in the application security industry. This team writes the rules which drive our static, dynamic, and runtime products. When researching new vulnerabilities, the team works together to identify the best and most efficient modality for detection. By leveraging a unified taxonomy across all three testing methods, Fortify can detect a weakness in source code with Fortify Static Code Analyzer (SCA) by OpenText™, then identify that same finding using dynamic analysis with Fortify WebInspect by OpenText™ in running environments where the weakness becomes a real vulnerability. Where static and dynamic can both detect a vulnerability, a rule is provided for each technology while maintaining a focus on accuracy and speed.

Customer Value

Static and Dynamic application security testing are complementary technologies in their ability to identify vulnerabilities across the entire SDLC, from development, to QA, to production. When these two technologies are unified across a common taxonomy, they augment one another to deliver a comprehensive solution. Customers see a more complete view of the vulnerabilities that threaten their organizations.

Real-World Example

Consider a basic weak SSL cipher vulnerability. While static and dynamic testing can both detect this weakness, the finding is heavily tied to the application's implementation in production. Static testing modalities will commonly return limited results for instances where SSL is configured from within the application. However, dynamic testing will provide a view of the web server configuration for instances where SSL is terminated outside of the application. By employing tools that leverage a shared taxonomy, Fortify is able to provide an extremely accurate analysis of the vulnerability's real security risk.

1. A unified taxonomy across testing methods enables a complete view of vulnerabilities.
2. Consistent remediation guidance enables collaboration and remediation
3. Powerful prioritization reduces the noise
4. Layered defense provides a safeguard
5. Unified vulnerability management creates feedback loops

2.

Consistent remediation guidance enables collaboration and remediation. By leveraging a unified taxonomy across both static and dynamic testing methods, developers are presented with results that share recommendation advice and security mappings.

Customer Value

By using software that uses developer-friendly language, developers won't need to spend as much time training to understand the reports. This allows them to spend less time researching vulnerabilities and more time remediating them.

Real-World Example

With DevOps methodologies becoming more and more prevalent, application security is becoming a team sport. Development, operations, and security teams require that the tools leveraged at various stages of the SDLC provide consistent vulnerability detail. By leveraging Fortify static and dynamic testing technologies, underpinned by a common vulnerability taxonomy, teams can collaborate on vulnerabilities in a clear and concise manner.

3.

Powerful prioritization reduces the noise. All vulnerabilities are not created equal. A weakness which is identified via source code analysis may be mitigated outside of code, leading to a lower net risk score. By layering dynamic analysis on top of static analysis, customers gain a valuable additional risk metric which allows them to see a more complete real-world risk picture.

Customer Value

It is not realistic to remediate all findings. Modern application security professionals are faced with difficult decisions when deciding which issues to fix, and which to defer. By leveraging a unified taxonomy across both static and dynamic testing, customers can gain an additional metric that allows them to choose which findings should be remediated first. Overall security posture is enhanced, and developers are able to use their time more efficiently by focusing on the most important findings first.

Real-World Example

Modern application security programs use a wide range of technologies and practices to mitigate risk. While static analysis does a great job of identifying a deep and broad set of vulnerability categories, it cannot account for production application context. An organization protecting XSS via a WAF may rightfully place a higher priority on remediating a non-WAF-protected vulnerability, like unsafe deserialization.

4.

Layered defense provides a safeguard. Static analysis provides excellent coverage, but it cannot be run against production environments where configurations and deployment options may have an enormous impact on the applications overall risk posture. Dynamic analysis allows for identifying issues later in the SDLC and into production where they pose the greatest risk.

Customer Value

By leveraging static analysis to identify vulnerabilities early in the SDLC and dynamic analysis to identify externally facing vulnerabilities later in the SDLC and into production, security teams can implement a layered approach which delivers greater security, because DAST acts as a safety net for vulnerabilities that aren't identified by SAST.

Real-World Example

It is true that DevOps cycles drive shorter release cycles that provide more opportunities to identify and remediate security defects, but the constantly accelerating churn of more releases also introduces more opportunities for mistakes. Dynamic testing can efficiently identify vulnerabilities that slip through the cracks due to developer mistakes, deployment errors, or environmental nuances.

5. **Unified vulnerability management creates feedback loops.** Security and Development teams need to consider a wide range of factors when identifying and remediating risk. The Fortify by OpenText™ tools eliminate one of those factors by providing these teams with a unified vulnerability management platform that allows them to easily analyze findings.

Customer Value

Teams are being overwhelmed by security information from point solutions which focus on their individual niches. A unified application security vulnerability management platform is not only critical in terms of the simplified prioritization and triage workflows that it introduces, but also in terms of the patterns that can be gleaned from the data.

Real-World Example

The most profound benefit to leveraging a unified vulnerability management platform centers around the data. A very basic example of this value can be seen in trending of vulnerability patterns. While it is important to identify vulnerabilities early in the SDLC using technologies like static analysis, it is critically important to create feedback loops that can identify when those findings surface in running environments via a DAST scan. An organization that identifies findings like XSS early in the SDLC and continues to detect those issues in production, can focus their training and development resources on addressing systemic problems.

Learn more at
www.microfocus.com/en-us/cyberres/application-security

Connect with Us
www.opentext.com

About Fortify Static Code Analyzer

Fortify Static Code Analyzer (SCA) pinpoints the root cause of security vulnerabilities in the source code, prioritizes the most serious issues, and provides detailed guidance on how to fix them so developers can resolve issues in less time with centralized software security management.

About Fortify WebInspect

Fortify WebInspect is a dynamic application security testing (DAST) tool that identifies application vulnerabilities in deployed web applications and services.

opentext™ | Cybersecurity