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 uni ed 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 e cient modality for detection. By leveraging a uni ed 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 nding 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 uni ed 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 nding is heavily tied to the application's implementation in production. Static testing modalities will commonly return limited results for instances where SSL is con gured from within the application. However, dynamic testing will provide a view of the web server con guration 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.

2. Consistent remediation guidance enables collaboration and remediation. By leveraging a uni ed 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 identi ed 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 ndings. Modern application security professionals are faced with di cult decisions when deciding which issues to x, and which to defer. By leveraging a uni ed taxonomy across both static and dynamic testing, customers can gain an additional metric that allows them to choose which ndings should be remediated rst. Overall security posture is enhanced, and developers are able to use their time more e ciently by focusing on the most important ndings rst.

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 con gurations and deployment options may have an enormous impact on the applications overall risk posture. Dynamic analysis allows

5. Uni ed vulnerability management creates feedback loops. Security and Development teams need to consider a wide range of factors when identifying and remediating risk.