WHITE PAPER

Changing the DAST Game *with* Netsparker IAST

netsparker

Executive Summary

Interactive application security testing, or IAST, aims to bridge the gap between static and dynamic application testing methods. Netsparker Shark is an additional, easy-to-deploy module that brings IAST functionality into Netsparker, adding an extra dimension of valuable information to the broad web application testing coverage of this industry-leading DAST solution.

Netsparker Shark works hand in hand with the main vulnerability scanning engine to provide truly interactive security testing. This sets it apart from many other products that are also marketed as IAST but are, in fact, separate testing tools with little or no interaction. The Netsparker engine directly interacts with a local Shark sensor for PHP, Java, or .NET to gather additional information about scan targets and identified vulnerabilities.

This document outlines the benefits of using Netsparker with its Shark IAST module in an enterprise setting.

Highlights from this white paper include:

- An overview of IAST in general and the Netsparker Shark approach in particular
- The advantages of truly interactive application security testing with Netsparker Shark
- The enterprise benefits of combining Netsparker's advanced dynamic application testing with the added depth provided by Netsparker Shark

A True IAST tool has to actively communicate with a core security testing platform while it probes each corner of the application environment.

Introducing True IAST and Netsparker Shark

As the simple web applications of the early 2000s got more complex and started processing ever more valuable data and business logic, it became clear that security testing needed to cover more than just checking the static source code. At the same time, early dynamic (runtime) security testing tools were harder to automate and provided very limited information about the issues that were found. The inherent limitations of both approaches gave rise to the idea of developing tools to cover the middle ground between static and dynamic security testing, with the first commercial product appearing in the late 2000s.

Fast-forward over a decade and we still see broadly similar categories in application security testing. Static application security testing, also called SAST or white-box testing, is built around source code analysis, so it cannot find runtime issues or test external dependencies. Dynamic application security testing (DAST), or black-box testing, probes the entire running application, so it can test the entire attack surface and find all the vulnerabilities that an attacker could. Even so, DAST still has no access to the source code, so it cannot truly pinpoint identified weaknesses.

A variety of tools exist to either add a dynamic element to static testing or provide more accurate issue information for dynamic testing. Despite major differences between such products, anything that combines white-box and black-box testing in this way is called gray-box testing or interactive application security testing (IAST) – an umbrella term popularized by Gartner in the 2010s.

Because this is a catch-all category, calling all gray-box testing "interactive" is, in fact, a serious misnomer. In reality, most tools marketed as IAST are simply standalone products that attach to a running application and provide additional insight into its internal workings. Depending on the product, they can be triggered by test suites during the build process or by a dynamic scanner (so-called DAST-induced IAST). In both cases, these are separate tests, usually delivering separate result sets.

To be called truly interactive and integrated, an IAST tool should actively exchange information with a comprehensive security testing solution throughout the testing process. This is the approach used by Netsparker Shark – the True IAST module for Netsparker.

> While a new feature in Netsparker, Shark is based on mature technology and built with the help of the same team that implemented the first DAST-based interactive application security testing tool over a decade ago.

What Makes IAST Different from SAST and DAST

Beyond the acronyms, application security testing products take different approaches to achieving the same basic goal: finding vulnerabilities. Each approach has its merits and shortcomings, so finding the right balance of usability and effectiveness is crucial for web application security. Netsparker's Shark IAST module adds inside information to the outside-in view provided by DAST to give you the best of both worlds.

SAST

REACTIVE SCANNING | **PROACTIVE SCANNING**

APPLICATION: STOPPED | APPLICATION: RUNNING

OUTSIDE ACCESS | **INSIDE ACCESS**

SAST tools perform static analysis to check the application source code or bytecode for insecure constructs. They can pinpoint issues accurately but are prone to false positives. They are also language-specific and cannot identify runtime vulnerabilities.

vs

DAST

REACTIVE SCANNING | **PROACTIVE SCANNING**

APPLICATION: STOPPED | **APPLICATION: RUNNING**

OUTSIDE ACCESS | INSIDE ACCESS

DAST tools probe a running application to safely simulate the actions of real-life attackers. Modern products such as Netsparker can find runtime vulnerabilities and often confirm them to eliminate false positives. While they have no access to the application source, they are language-independent and can test the entire application environment.

vs

IAST

REACTIVE SCANNING | PROACTIVE SCANNING

APPLICATION: STOPPED | **APPLICATION: RUNNING**

OUTSIDE ACCESS | **INSIDE ACCESS**

IAST tools attach to a running application to see what is going on inside. They are passive, so their coverage depends on how they are triggered. In the case of Netsparker Shark, the IAST sensor continuously communicates with the core vulnerability scanning engine to add inside information to the broad coverage of DAST.

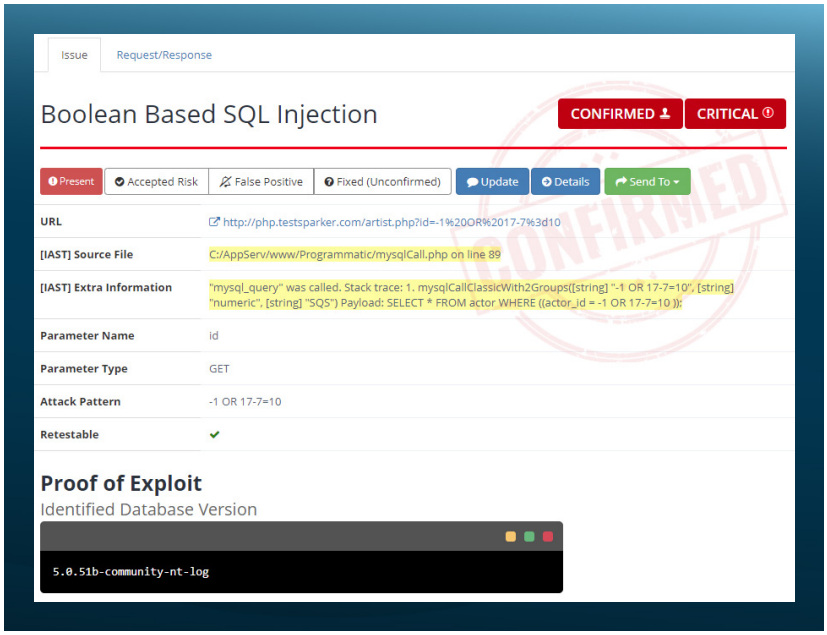
The Advantages of True Interactive Application Security Testing with Netsparker

Netsparker Shark extends the industry-leading DAST capabilities of the Netsparker vulnerability scanning engine. As Netsparker crawls web assets and probes them for vulnerabilities, Shark sensors installed locally in the application testing environment continuously provide additional information about vulnerabilities and the environment itself. This allows Netsparker to pinpoint many vulnerabilities right down to the line number[1], overcoming a limitation of traditional dynamic application testing.

DAST + IAST = Netsparker with Shark

The crucial advantage of the Netsparker approach to gray-box testing is that the IAST module complements what is already an advanced and highly accurate DAST solution. The True IAST concept combines the broad testing scope of Netsparker's dynamic testing engine with the benefits of low-level insight into the runtime environment. All the information provided by Shark sensors is seamlessly integrated into Netsparker scan results with no manual processing required and is available both in Netsparker's intuitive user interface and through the internal API.

Compared to code-level security testing and gray-box products that require source code modifications to build the required instrumentation into the application, Netsparker Shark is very easy to deploy. To add IAST capabilities to Netsparker vulnerability scans, you simply install the right sensor in your application testing environment and start scanning as usual. Shark not only supports Java and .NET applications (like many other IAST products) but also comes with a PHP sensor. Sensors for additional technologies are already in development.



Proof-Based Scanning with Even More Proof

A stand-out feature of Netsparker is its Proof-Based Scanning™ technology that enables the scanning engine to safely exploit and automatically confirm many vulnerabilities, often extracting sample data as proof that the vulnerability is real and exploitable. The Shark IAST module augments these results by isolating the source of the issue, often down to the specific line number. It can also provide confirmation and attack payloads for more vulnerabilities and return additional information about the security of the application environment.

[1]For .NET applications, bytecode-level information is provided.

The Shark sensor uses its access to the application execution environment not only to provide runtime insights but also to identify hidden and unlinked assets that are beyond the scope of traditional crawler-based discovery. This allows security teams to find and fix many issues that might otherwise slip under the radar, such as server misconfigurations or information exposure.

> Netsparker Shark takes an already excellent application security testing solution and makes it even better.

All the extra intelligence supplied by Shark is automatically collated with results from the main scanner to provide security engineers and developers with detailed and actionable vulnerability reports for a wider range of issues, including out-of-band vulnerabilities. This makes it easier to estimate the impact of each vulnerability, resulting in faster and more accurate issue triaging and resolution. For issues that have been automatically confirmed, you can even use Netsparker to automatically create and assign developer tickets containing all the vulnerability details extracted both by the core scanner and by Shark, complete with information about the potential impact and remediation guidance.

TECHNICAL REQUIREMENTS FOR ADDING NETSPARKER SHARK TO YOUR ENVIRONMENT

- You will need Netsparker deployed and configured to match your application environment.
- PHP, Java, and .NET applications are currently supported.
- The application you are testing must be mature enough to be compiled and executed (same requirement as for DAST).
- The application runtime environment needs to allow for the deployment of an IAST sensor. For CI/CD pipelines, the recommended way is to include the sensor in a redeployable environment, such as a suitable Docker image.
- The execution environment should allow for a minor performance overhead that may be added by the IAST sensor. Based on our extensive internal testing, running Shark increases resource utilization by 5% on average, so any SLAs that may be in place for that environment will need to account for that.

Enterprise Benefits of Netsparker Shark

The technical enhancements that True IAST with Shark brings to Netsparker add a new dimension to dynamic testing. This translates into a wide variety of benefits all across the application development and testing workflow, from improving security and internal collaboration to further shortening time to value. Most importantly, the runtime insights provided by Shark extend the ability to confidently automate application security testing for maximum scalability in dynamic enterprise environments.

Improved Application Security

The obvious and also most important benefit of more detailed vulnerability identification is improved security. Shark adds extra depth to the issue information gathered by Netsparker for identified vulnerabilities, helping developers to pinpoint and fix flaws much more quickly. This includes many advanced and out-of-band vulnerabilities which can now be resolved more efficiently than with DAST output alone.

Having detailed information about vulnerabilities and possible attack payloads allows security engineers and Netsparker itself to triage issues faster and more accurately. It also helps developers to understand issues in depth and eliminate root causes instead of deploying superficial patches. This improves security in the long run so vulnerabilities are fixed for good and don't resurface in future releases, paving the way to mature DevSecOps workflows where security is an integral part of development.

> Shark brings additional information to scan results and expands the scope of detectable issues.

Beyond bringing additional information to scan results, Shark also expands the scope of detectable issues. Because the Shark sensor is deployed in the runtime environment, it can identify and test unlinked assets that are not directly accessible to the main vulnerability scanner, such as hidden files or web server configuration files. The ability to find and test these assets is a major boon for security, as human attackers might be able to target them in chained attacks that are normally beyond the scope of automated scanners.

Scalable Application Security Workflows

Netsparker's Proof-Based Scanning™ technology is a key enabler of automation in application security testing, allowing even the largest organizations to scale their vulnerability scanning and remediation capabilities without overwhelming the security team. The addition of True IAST with Shark takes this up a notch to provide automatic confirmation (and often proof) for even more issues so you can send them directly to developers with no risk of false positives. To make this possible and efficient, Netsparker provides out-of-the-box integration with popular issue trackers and collaboration platforms, rounded out by a comprehensive internal API for customization.

True IAST with Shark provides automatic confirmation for even more issues so you can send them directly to developers with no risk of false positives.

By handing developers detailed and fully trustworthy bug tickets complete with attack payloads and sometimes even a specific file name and line number, Shark helps to streamline security issue resolution to keep development workflows running smoothly. This minimizes the risk of lingering bugs, incomplete fixes, lengthy exchanges between developers and security engineers, and all the other problems that often cause security issues to delay releases (or – even worse – be ignored to avoid delaying a release).

Because automatically confirmed vulnerability reports are always real and actionable issues, Netsparker can directly convert them into developer tickets based on predefined severity thresholds. This bypasses the bottleneck of manual verification and triaging by the security team to unlock true scalability in enterprise organizations with hundreds of developers but only a handful of security engineers. In fact, combined with the right internal organization and workflows, Netsparker with Shark can even be set up and used as a fully automated application security platform, allowing companies to shift the entire vulnerability resolution process to the development level.

Better Working Relations Between Teams

Netsparker makes it easy to integrate with popular issue trackers and send automatically confirmed, 100% real vulnerabilities directly to the developers, complete with detailed reports and added insights from the Shark module, often including the file and line number. This is in stark contrast to the traditional approach where security engineers identify vulnerabilities and then manually create, assign, and manage developer tickets. Misunderstandings and insufficient remediation guidance can then lead to friction and an adversarial approach rather than a collaborative one. However, when actionable issues are received from a trusted tool, not a human, the role of security engineers shifts away from pestering developers about fixes and towards educating them about security and helping them find the best ways of implementing fixes.

The additional information provided by Shark also helps to minimize extra work and inefficiencies resulting from false positives – the scourge of less mature application security testing tools. Proof-Based Scanning already allows Netsparker to clearly indicate which vulnerabilities have been automatically proven to be exploitable. For these issues, there is no risk of false positives and you know exactly which scan results can go straight to the remediation stage.

> The IAST module supplements the core vulnerability testing engine to provide additional details and attack payloads, reducing manual verification for the security team.

The Shark IAST module supplements the core vulnerability testing engine to automatically confirm even more issues and provide additional details and attack payloads. This leaves the security team with much less manual verification to do. Better still, because even unconfirmed issues are reported in detail and accompanied by additional insights gathered by Shark, developer tickets created after manual verification are far less likely to be annoying and time-consuming false alarms.

Cost Savings and Shorter Time to Value

Improved security, streamlined development workflows, more efficient collaboration between security engineers and developers – all these benefits can also have a positive effect on your bottom line. The additional vulnerability details provided by Shark can greatly reduce the time to fix, resulting in fewer man-hours spent on hunting bugs and more on developing new features. Combined with Proof-Based Scanning™ and Netsparker's integration capabilities, this can completely change the dynamics of application security testing. Instead of constantly fighting fires, weeding out false positives, and manually verifying suspected vulnerabilities, the security team can now focus on analyzing more complex vulnerabilities and providing guidance to improve organizational security in the long run. Maximum automation also reduces communication overhead and helps everyone work more efficiently.

An obvious, though indirect, financial benefit of improved application security is a decreased risk of costly security incidents, such as data breaches, application outages, fraudulent transactions, and many others. With the average total cost of a data breach estimated at $3.86 million and an average time to identify and contain a breach reaching 280 days[2], seemingly minor security improvements can prevent a lot of pain down the road. Another indirect benefit of more accurate and efficient application security testing is that you can avoid the cost and delays of reworking code at a late stage of development or even having to pull back releases due to security flaws.

> The time from initial deployment to your first scan results, first automatically confirmed bug ticket, and first resolved vulnerability can now be as short as a few hours – and that is your time to value.

Lastly, Netsparker with the Shark True IAST module gives you unprecedented time to value. In the realm of application security testing, getting the first measurable benefits from your investment can take weeks or even months, especially for code-level testing that needs to be manually integrated with the development pipeline. In contrast, adding IAST capabilities to Netsparker is just a matter of installing an additional Shark sensor in your runtime environment.

[2]Source: 2020 Cost of a Data Breach Report, IBM Security

Netsparker Shark Dives Deeper

To be effective, modern web application security testing needs to combine the widest possible test coverage with the accuracy required to efficiently isolate and resolve vulnerabilities. Netsparker has tackled this challenge head-on by adding True IAST capabilities to the broad scope and trustworthy results of its industry-leading DAST solution. Netsparker Shark adds deep runtime insights to vulnerability reports generated using Proof-Based Scanning and supplements them by identifying assets that are beyond the reach of crawlers.

The introduction of True IAST into the enterprise security model helps organizations build a scalable application security program through efficient and confident automation. With Netsparker Shark deployed in your application testing environment, your security and development teams can get more detailed vulnerability information from Netsparker to resolve issues more quickly and unlock efficiencies all across the application security workflow.

ABOUT NETSPARKER

Netsparker is a comprehensive automated web security solution that includes web vulnerability scanning, vulnerability assessment, and vulnerability management. Its strongest points are scanning and crawling accuracy, rapid asset discovery technology, and integration with leading issue management and CI/CD solutions.

The Netsparker scanner can find vulnerabilities in all types of modern and custom web applications, regardless of the architectures or platforms that they are based on. For many classes of vulnerabilities, the scanner can safely exploit the identified flaw and deliver proof that the issue is real and not a false positive, greatly improving automation and scalability.

Netsparker is designed for organizations that require a customizable and scalable solution for complex environments. Netsparker is also available in other variants to suit different customer requirements. Depending on the variant and deployment needs, Netsparker can be implemented as desktop software, a managed service, or an on-premise solution.

Netsparker is part of Invicti Security, a leader in dynamic and interactive application security testing.