A Theoretical Versus a Practical Approach Toward Application Security

Why DAST remains a primary pillar in a holistic AppSec program

Modern software development practices such as agile and DevOps have increased the complexity and speed of applications being released in the world. But as applications become smaller and more numerous—e.g., microservices, mobile apps, APIs, single-page applications, dynamic front ends, etc.—understanding and managing their security risk often requires more time than business needs allow.

The rift between development and security teams is not new, but the conflict between delivering complex software quickly and delivering it securely has only exacerbated it. A recent survey performed by the SANS™ Institute and commissioned by Black Duck® found that a lack of developer buy-in and an increase in organizational silos between development and security teams are ongoing barriers to implementing DevSecOps.[1]

Shifting security left by introducing application security testing (AST) tools earlier in the software development life cycle (SDLC), while successful in preventing some vulnerabilities from getting into production, presents cultural challenges, such as

- Lengthy scan cycles
- Duplicate findings and false positives
- Proliferation of tools and scans[2]

Another important consideration, and one frequently overlooked in the rush to “shift left” and place more security responsibility into the hands of developers, is that **security risk always comes from the right**—that is, from the production side of the equation. A developer who writes risky code hasn’t put the organization at risk until that code is deployed in the production environment.

This eBook examines the persistent need for testing applications in their running or deployed state, key differences between static application security testing (SAST) and dynamic application security testing (DAST) and why they should complement each other, and the ways that DAST benefits developers and software security professionals. Lastly, the eBook highlights the unique features of Black Duck Continuous Dynamic™ that enable organizations to build trust in their software by testing applications at DevOps speed and enterprise scale.

> The sum of an application is much different than its pieces, and how those pieces interact with each other may also be different when deployed versus when in development. Thus, the perimeter of an application can only be known in its deployed state. And its security risk can only be known by testing the application in its deployed state.

What works in theory may not work in practice

Software composition analysis (SCA), SAST, and other tests run in the SDLC help prevent vulnerabilities from being released into production. But such “white box” tests cannot help the security team determine, understand, or manage their organization’s risk posture. SAST and SCA findings are only as complete as the vulnerabilities they’re checking against. They rely on enumerated exploits, and they only test pieces of an application: SAST tests proprietary code; SCA tests open source and third-party dependencies.

The sum of an application is much different than its pieces, and how those pieces interact with each other may also be different when deployed versus when in development. Thus, the perimeter of an application can only be known in its deployed state. And its security risk can only be known by testing the application in its deployed state.

DAST approaches this problem by utilizing “black box” testing. Regardless of whether there are known vulnerabilities in the code or if the code is proprietary or open source, DAST focuses on understanding how the application behaves in response to tests. It delivers payloads to exploit vulnerabilities that may have slipped through prior scanners or were introduced by a new update, such as misconfigurations, authentication issues, and insecure programming practices.

The analogy of a racecar is useful here. Engineers can test the safety and integrity of every component in a racecar, but that doesn't alleviate the need for test laps. A DAST-centric approach to application security recognizes that the thousands of components that make up the racecar are working together for the first time in an environment loaded with variables—the track, the weather, the skill of the team that assembled the car, the driver's experience, etc.

Until driving the racecar in its intended environment, all one can say is that it is theoretically safe. No one knows for certain until it is tested in real-world conditions.

Defense-in-depth

No security scanner is perfect. There's going to be a deficiency somewhere, maybe in the approach, the technology, the process, or the people that surround it. The answer to imperfect (or incomplete) testing is defense-in-depth, and it's why 70% of organizations utilize 11 or more AST tools at any given time.[3]

SAST, SCA, and DAST are the three pillars of a solid application security program. SAST, for example, can find many OWASP Top 10 vulnerabilities and eliminate them before they get into production. But SAST will never eliminate all vulnerabilities. Neither will SCA—or DAST, for that matter.

Perfect security exists only in theory. In practice, security teams need to understand their risk profile, manage it based on the policies of their organization, and prioritize their triage and remediation efforts accordingly.

Defense-in-depth is a great methodology for managing risk and mitigating gaps in security. As development organizations are under increasing pressure to release new features as quickly as possible, it's essential that they prevent insecure code from making its way into the CI/CD pipeline. Organizations must also ensure that their deployed code is constantly checked for vulnerabilities introduced by endless updates. Finding documentation on old code can be difficult and may limit an organization's ability to scan with SAST or SCA solutions. DAST provides the final layer of safety in adherence to a defense-in-depth methodology.

DAST also lets CISOs and their teams know the risk position of their web applications and APIs, as well as which vulnerabilities are crucial to mitigate. White box tests are good security hygiene, but they don't reveal what risks your organization is exposed to right now.

Compliance requirements

Many compliance frameworks, including the payment card industry data security standard (PCI DSS), require DAST because DAST simulates attacks as a hacker would perform them. It crawls a website, delivers payloads, and brings back evidence of real-world vulnerabilities or weaknesses, such as cross-site scripting and SQL injection, that can lead to data breaches. Industries such as retail and eCommerce, financial services, and healthcare require this type of testing for any applications that collect sensitive data.

DAST also provides context for a vulnerability or weakness. It lets the security team know what is exploitable—and in what ways and what damage can be caused.

Common compliance frameworks

- PCI DSS
- HIPAA
- ISO/SAE 21434
- NIST

Continuous Dynamic exceeds the requirements of PCI DSS by providing ongoing, verified vulnerability assessments for internal and public websites. Integrations with web application firewalls support the creation of virtual patches to fix vulnerabilities while providing reports needed for auditor inspections.

The three-legged stool of DAST

Any great DAST solution has three requirements: discovery, attack engine, and authentication. Each one provides an essential element to cover all checkpoints for a scanner. But just like a three-legged stool, if one element is weak or broken, the entire structure falls apart.

Discovery

A DAST tool can do several things to discover what it's going to attack. These involve some kind of input to tell it what to do.

- Crawling
- Probing (clicking buttons on a single-page application, entering data, etc.)
- Documentation (API specs, API doc)
- Functional tests
- Listening to API gateways

Attack engine

Once the tool knows what it's up against, it attempts to simulate an attacker. This is the bread and butter of any DAST solution, and it's where a tool's efficacy, accuracy, and coverage (e.g., OWASP) come into play.

DAST attacks by

- Manipulating requests
- Adding dangerous payloads then looking for responses to determine weakness or vulnerability

Authentication

Usually, the most valuable data on a site requires authentication to access it, and one of the most-overlooked aspects of a DAST solution is making authentication is easy for the scanner. If a scanner can't authenticate itself, it can't reach applications that live behind the authentication, leaving the organizations vulnerable for a potential breach. Some of the authentications methods include

- Basic username and password credentials
- Multifactor authentication
- SMS pushes
- Time-based passwords

When the three elements are equally strong, DAST provides protection where it's needed most, allowing organizations and their security teams to

- Reduce business risk
- Protect brand reputation
- Minimize windows of exposure
- Achieve PCI compliance
- Provide attestation
- Protect against zero-day threats

For most scanners on the market, configuring the scanner to authenticate against the site is time-consuming and complicated. Continuous Dynamic reduces the burden of setting up authentication and maximizes coverage.

Continuous Dynamic is fully secure

Continuous Dynamic has two SOC2-compliant data centers: one in the United States that also complies with the California Consumer Privacy Act and GDPR, and an AWS data center in Germany.

Best-in-class DAST

Security teams should ask five crucial questions to their potential DAST solution providers. This is how Continuous Dynamic answers them.

Does your DAST solution support a cloud-first strategy?

A software-as-a-service (SaaS) solution for DAST such as Continuous Dynamic requires no hardware or scanning software to be installed. Continuous Dynamic deploys quickly, is scalable to fit any environment, and matches any pace of development. The benefits of this type of SaaS solution include

- Cost-effectiveness (no maintenance or operational costs)
- Ease of use
- Scalability
- Backup recovery
- Ability to focus on building applications and managing assessment results

By comparison, on-premises solutions require longer lead times to implement, internal resources to manage, costly hardware to run, and maintenance and upgrade costs.

Is your DAST solution production safe?

Production-safe DAST solutions such as Continuous Dynamic don't require a separate test environment, saving time and cost without causing downtime. By using single-thread, low and slow, benign injectors instead of code, there is no threat to internal- or external-facing websites.

Does your DAST solution provide unlimited, continuous, and concurrent scanning?

Continuous Dynamic constantly scans websites as they evolve. Continuous scanning minimizes the time between when a vulnerability is introduced and the time it is discovered. This "always on" approach provides

- Automatic detection and analysis of code changes to web applications
- Alerts for newly discovered vulnerabilities (e.g., Log4j)
- Ability to onboard and scan more than 10,000 websites concurrently
- Ability to retest a vulnerability without having to test from the beginning

Does your DAST solution provide business logic assessments?

Continuous Dynamic comes with a team of security professionals that can specifically look for business logic flaws that can't be detected via most tools. Similar to penetration tests, business logic assessments can find issues in the context and logic of running applications that can produce unintended functionality of production websites or deployed code.

For instance, a behavior in an application may not present evidence of a vulnerability but still results in the application not performing as expected. The business logic assessment provides context to show the potential avenues a hacker could take to reach a vulnerability.

Continuous Dynamic uses a unique combination of security experts augmented by machine learning and artificial intelligence to eliminate false positives. The Continuous Dynamic "Ask a Question" feature provides direct access to security experts via the portal for remediation guidance and to confirm flagged vulnerabilities are indeed vulnerabilities.

Business logic assessments analyze user-application interactions such as

- Account profile/settings
- Account transactions/history
- Checkout
- Contact us
- Forgotten username or password
- Shopping cart processes (adding/removing items)
- Site administration

Does your DAST solution eliminate false positives?

Nothing infuriates developers more (or is more costly) than receiving dozens of vulnerabilities from the security team only to realize the majority of them aren't vulnerabilities at all. With Continuous Dynamic, the security team can ensure that developers get verified vulnerabilities with near-zero false positives, meaning developers can worry less about triaging findings and tracking down bugs. This increases their productivity and allows them to focus on what they love: writing code and creating features.

Conclusion: Better together

With so much emphasis on "shifting left" in recent years, it's tempting to forget that security risk starts on the right. Risky code is only a risk once it's deployed.

Organizations get breached in production. That's a fact. And it could be why Forrester's 2022 Security Survey shows that organizations implementing DAST for their production websites were the least breached.[4]

This doesn't mean white box testing like SAST and SCA aren't necessary. There are plenty of real-world benefits to catching security flaws early in the SDLC and preventing them from reaching production. But no test is perfect. The only way to assess and manage an organization's risk posture is to scan its actual attack surfaces. DAST scans an application's perimeter and answers the questions on every CISO's mind: What's my risk? What vulnerabilities do I have right now? Are they manageable?

The most mature application security programs employ a layered approach. Utilizing defense-in-depth, they realize the best of both white box and black box testing. They use SAST and SCA to ensure they're not introducing known vulnerabilities into production, and they use DAST to monitor their risk posture.

As developers get better at secure coding practices, they can eliminate many risks in production—but organizations will always need DAST to know what risks remain and what new risks are introduced.

Discover how Continuous Dynamic can help secure your web applications.
Request a demo today.

Resources

1. Chris Edmundson and Kenneth G. Hartman, "SANS 2022 DevSecOps Survey: Creating a Culture to Significantly Improve Your Organization's Security Posture," The SANS Institute and Synopsys, September 2022.
2. Dave Gruber, Cracking the Code of DevSecOps, Enterprise Strategy Group, June 2021.
3. Ibid.
4. Janet Worthington, "A Modern Approach to Application Security," Synopsys webinar, October 13, 2022.

BLACKDUCK®

About Black Duck

Black Duck® offers the most comprehensive, powerful, and trusted portfolio of application security solutions in the industry. We have an unmatched track record of helping organizations around the world secure their software quickly, integrate security efficiently in their development environments, and safely innovate with new technologies. As the recognized leaders, experts, and innovators in software security, Black Duck has everything you need to build trust in your software. Learn more at www.blackduck.com.