Banking, Financial Services & Insurance: Shifting Application Security Left, Successfully

Bright Security White Paper
Introduction

What is Shift Left testing?

Shifting Left: the benefits

The advantages of Shifting Left

Fast, efficient and secure software delivery
Empowering developers
Prevent interrupted sprints
Reduce the cost and time to fix
Maximize developer productivity
Reduce security and technical debt

Benefits of Shifting Left for Security Teams
More secure applications and APIs
Reducing the reliance on periodic testing
Maximize attack surface coverage
Faster scan times
Enhanced prioritization of fix
Evolving role of application security experts

Shifting Left - why you need to start now

The matter of shifting left is not a case of ‘if’, but ‘when’.

Are you ready to Shift Left? Yes, you are!

Start now and mature Later
Leveraging QA
Security testing as part of the CI/CD Pipeline
Benefits of deploying a developer-first DAST tool in the pipeline
The ultimate goal...Unit Security Testing

Why DAST, Why Bright?

Summary
Introduction

As highly regulated industries, much is at stake for banking, financial services, and insurance organizations, which depend on client trust, privacy, and risk management. It is imperative to protect the sensitive data clients entrust from constant cyber attacks and data breach attempts. Data breaches can result in financial and reputational loss, as well as regulatory penalties.

The BFSI industry has undergone a digital transformation, driven by digital-first organizations and facilitated by regulations such as the Open Banking Directive (PSD2) in the EU and the increased reliance on APIs globally. This has led to a significant increase in the attack surface and security vulnerabilities, as developers ship new software and features at an unprecedented velocity.

Traditionally, application security testing, particularly Dynamic Application Security Testing (DAST), was performed during later stages of the development pipeline. Legacy DAST solutions were intended for security professionals and were used mainly by internal security teams or third parties on a periodic basis to supplement manual testing.

“Frequent and thorough verification by developers as early as possible in the software development life cycle (SDLC) is one critical element of software security assurance”.

The National Institute of Standards and Technology (NIST) Guidelines on Minimum Standards for Developer Verification of Software, Oct’21

These legacy practices, coupled with delays in the developer feedback loops and the many resulting bottlenecks, have not kept up with organizations’ demands in a world of agile development, CI/CD and DevOps. They prevent the guiding themes of automation and ever-increasing development velocity.

The ability to scale security testing and truly reduce your risk exposure, is directly linked to your ability to eliminate these manual processes and empower developers to identify and resolve the real vulnerabilities in applications and APIs that occur in production.

It is therefore imperative to develop securely from the start, by enabling developers to identify and fix security defects much earlier in the SDLC. This is known as Shifting security left.

We currently assess the creditworthiness of banking organizations using credit rating agencies such as Moody’s, Fitch, and Standard & Poor. As data breaches against BFSI organizations become increasingly common, the security of applications and the overall security posture of the organization are likely to also become important factors for customers when choosing a financial institution.
What is Shift Left testing?

Shifting security left means introducing security as early as possible in the Software Development Life Cycle (SDLC). For testing automation and checks, the earliest possible phase is during development.

With the ratio of security to developers, particularly in BFSI organizations, being 1:500, the responsibility for security must be shared across the organization. While the AppSec team can provide governance across what should be tested, when and how, the actual work needs to be spearheaded by development teams who are empowered to test and secure what they build while maintaining their rapid release cycles.

Developer-focused security tooling provides developer autonomy to ensure products are intrinsically secure by design, while being managed and governed by the security team. Automated testing on every build means security issues are detected and fixed earlier, more often, and faster. As a result of removing manual reviews and delivering consistent testing policies, code can be pushed to production faster.

In order to effectively and successfully Shift DAST Left:

Security tooling needs to be built with developers in mind
DAST scanners were traditionally built for security experts. This meant you had to be uniquely trained to use them as they were complex and difficult to use and configure.

Developers must be able to initiate scans automatically & quickly
Scans must be controlled by developers via the Command Line Interface (CLI), without leaving their existing toolset.

Tooling needs to be integrated throughout the SDLC pipeline
This enables developers to automatically test every build, PR or merge, in the CI/CD.

DAST security scans should be deployed in Unit Testing
This is the earliest time to deploy DAST for lightning fast scans so that developers can detect and remediate vulnerabilities 60 X faster.

A feedback loop directly to developers of any security bugs detected is required
Immediate remediation can be performed while the developer still has the relevant context and understanding of the code they have written.
Accuracy is paramount to remove the noise and alert fatigue of false positives, which in turn will:

- Provide developers with clear and actionable results, delivered in real time.
- Remove the need for costly and time-consuming manual validation of findings by the AppSec team which is already stretched too thin.
- By implication, reduce both, security and technical debt.

Security, from the CISO to the AppSec team, needs complete visibility

Understanding the extent of testing coverage, which security tests were performed, and what vulnerabilities were identified and fixed, will ensure adherence to security policies and compliance requirements. It will also reveal the need for additional secure coding training and the implementation of best practices for the engineering teams.

Shifting Left: the benefits

Regular testing on every build/commit or at least daily allows for a focus on making informed security decisions as part of a streamlined DevSecOps strategy. This results in the delivery of software with speed, efficiency, and security.

Organizations relying solely on manual testing struggle to keep pace with accelerated development timelines, leaving vulnerable applications and APIs in production. The success of this strategy depends on development teams having access to user-friendly, accurate, and seamlessly integrated technology.

Traditional, legacy Dynamic Application Security Testing (DAST) tools are not built for this regular cadence of security testing that demands speed, accuracy and even earlier deployment in the SDLC. Modern DAST solutions, like Bright’s, are leading the charge.
The advantages of Shifting Left

Fast, efficient and secure software delivery

The primary goal of shifting security to earlier stages is to deliver secure software at scale. DevOps has revolutionized the way unit and integration testing are incorporated into release cycles, resulting in the early detection and quicker remediation of functional bugs for higher quality code releases.

By applying the same principles to security testing, developers can detect and fix security bugs early and efficiently. This allows for the scaling of security efforts to support large-scale development while maintaining the security of applications and APIs. No longer a hindrance, security can now be integrated as a seamless part of the development process.

Empowering developers

Empowering developers to own security testing means organizations can distribute ownership of delivering secure applications across a much larger team and not rely on a very small AppSec team. This also allows developers who ultimately will remediate the issues to spend less time dealing with security bugs.

Traditionally, security prevents developers from actually focusing on their job of delivering product, with security gating the continuous development process instead of merging into it.

With developer-centric security testing automated across and integrated into your SDL, the debilitating domino effect of sidelining security is eliminated.

Prevent interrupted sprints

There is no need to put other work on hold to prioritize fixing what has now become a legacy security issue, or risk found in production. Security bugs need to be fixed on-the-go by the same developer who wrote the code, instead of the average 280 days this takes with legacy practices.

Security vulnerabilities should be handled in the same manner as any other bug, rather than being solely the responsibility of the security team to address in isolation.

Early detection and remediation of vulnerabilities in the software development life cycle results in a 60 times reduction in both time and cost compared to fixing vulnerabilities that have been released into production.
Reduce the cost and time to fix

Early detection of security bugs results in easier and less expensive fixes. Automated security testing integrates feedback and remediation into the continuous DevOps processes, enabling organizations to scale their efforts and improve productivity.

As software development progresses, the cost of fixing any undiscovered bugs increases, often dramatically. Research from the Ponemon Institute found that fixing security vulnerabilities detected early in the development process can cost around $80 on average, while the same vulnerability found in production could cost up to $7,500 to remediate, not including the potential costs of a security incident or data breach caused by the bug being exploited by a malicious user.

Maximize developer productivity

By implementing security testing in the SDLC, developers can treat security bugs the same way they do functional bugs.

With security testing that runs in parallel with both build and integration testing in the CI/CD, every new finding detected is raised with the developers immediately.

Faster remediation is possible as the developer who is familiar with the code and understands the context of the feature they are creating can handle the fix. This significantly reduces the time needed to fix security bugs, enhancing productivity, lowering costs, and improving the security posture of the organization.

DevOps and the “Shift Left” Principle

Fixing quality & security issues rises costs significantly as the development cycle advances

- Coding $80/defect
- Build $240/defect
- QA & Security $960/defect
- Production $7,600/defect

Reduce security and technical debt

All of the above efficiencies mean the continual compounding effect of security debt and the resulting technical debt can now be addressed and mitigated by putting security testing into the hands of developers.
Benefits of Shifting Left for security teams

While shifting security left allows developers to be more efficient, it also comes with many advantages for the security teams.

Regardless of your organization's size, developers can outnumber security by 500:1 and with almost 4 million open cyber security jobs globally, the problem is only set to get worse.

Automation is the only way to bridge the cyber security jobs shortage. By shifting left, security is scalable by adopting consistent and efficient security testing across your pipelines, managed and governed by the security team.

More secure applications and APIs

While maximizing efficiency across your engineering and security teams, shifting left also means your applications, microservices and APIs are going to be intrinsically more secure.

Moving security testing to earlier stages in the development process reduces the delays in testing and releases caused by blocked testing, ultimately improving engineering velocity and product delivery.

In addition, security testing in production carries greater risks. Not only is it too late to detect vulnerabilities, but it also leaves the door open for malicious actors to exploit them first. Finally, there are other security limitations that can be addressed by shifting security testing left in the development process.

BY SHIFTING LEFT, THE SECURITY TEAM CAN NOW:

- Implement mandatory security scanning for every build/commit to streamline the process of identifying, verifying, and addressing security vulnerabilities before they reach production.

- Adopt a consultative approach to enhance your organization’s security posture, prioritizing remediation efforts and minimizing risk, while providing guidance to developers to address vulnerabilities early in the SDLC.

- Direct efforts towards complex security vulnerabilities that require specialized expertise while relying on automated tools to identify and address low-to-medium risk issues in the pipeline.

- Reduce reliance on manual testing, both in-house and third-party, to minimize costs and increase efficiency.
Reducing the reliance on periodic testing

Unless security testing is baked across your pipelines, security testing in production takes time and is typically carried out monthly, at best, or annually as a compliance measure.

Maximize attack surface coverage

Testing in production results in coverage limitations. Depth of testing is harder to achieve because the tester is not the developer and in many cases, is instead a third party penetration testing company with a limited knowledge of the application.

Overcoming security measures such as adding the scanner to approved lists in a WAF, OTP requirements, CAPTCHA, and other proxies can be challenging for security testing. The manual configuration of these tools requires expertise from security professionals and can be difficult to achieve full coverage.

By incorporating security testing earlier in the process, these security measures are not yet implemented, making it easier to configure scans and maximizing coverage, resulting in more efficient and secure outcomes for all involved.

Faster scan times

Testing in production requires the scanning of a large number of entry points, with test times taking 7+ days. This is not conducive to agile development and can cripple your rapid release cycles, but also results in far more work for the security team.

Shorter, scope-defined tests, on every build or as part of your unit testing, means faster scans and immediate feedback on security issues in the pipeline.

Enhanced Prioritization of Fix

With potentially hundreds of security findings detected in production, prioritization and remediation of findings is a constant challenge and results in friction between the engineering and security teams, leading to compounding of security debt.

Detecting security bugs early and often removes this, as both teams work together in real time to continually remediate security findings. Developers can treat and fix security issues like their functional bugs, with no requirement for security intervention and protracted meetings that would be essential if a bug was to be remediated in production.

Evolving role of AppSec experts

By detecting security vulnerabilities earlier in the development process, the perception of security as a hindrance is transformed into a collaborative effort between security and development teams. This shift to the left positions security as an enabler, working closely with the dev teams as a trusted advisor and consultant.

By doing so, security can focus on addressing more complex security issues and escalate to R&D for resolution, reducing friction between security and development, and ensuring the delivery of secure software.
Shifting Left - why you need to start now

The matter of shifting left is not a case of ‘if’, but ‘when’.

Hands-on responsibility for application security design and testing tasks is shifting to development and engineering teams. Gartner research reveals that more than half of software engineering leaders are directly responsible for application security, and another third share that responsibility.

It is inevitable that implementing security testing early and often in the CI/CD has to be achieved to deal with the large number of builds and iterations in software we see on a daily basis. Security as we know it cannot keep up - this has to change and is now not just on organizations' wish lists, but a minimum required standard in software security.

Included in these minimum requirements are:

1. Automated testing for consistency and to minimize human effort
2. Recommended Minimum Standard for Developer Testing includes:
   - “Frequent and thorough verification [testing] by developers as early as possible in the software development life cycle (SDLC) is one critical element of software security assurance”
   - Do dynamic security testing analysis:
     1. Run the program with built-in checks
     2. Create code-based (structural) test cases. Add cases as necessary to reach at least 80% coverage
     3. Use test cases that were designed to catch previous bugs
     4. Correct the “must fix” bugs that are uncovered and improve the process to prevent similar bugs in the future, or at least catch them earlier
   3. Verification must be automated in order for thousands of tests to be accurately performed and for the results to be precisely checked. Automation also allows verification to be repeated efficiently and often.

NIST - Guidelines on Minimum Standards for Developer Verification of Software

The National Institute of Standards and Technology (NIST) have released their publication on ‘Guidelines on Minimum Standards for Developer Verification of Software’, in response to the May 2021 Executive Order (EO) 14028 on Improving the Nation’s Cybersecurity. This is in addition to NIST’s Secure Software Development Framework (SSDF).

Its goal is two-fold: (1) to establish minimum standards for software testing by highlighting best practices that developers should already be adopting and (2) to serve as a foundation for future, mandatory regulations.
With the exponential and continually growing threat of cyber attacks, relying on reactive defensive measures is not sufficient. Organizations have a duty to their users to ensure their software and the PII of their clients is secure. We have already discussed that delayed periodic testing is no longer viable, whether operationally, financially or from a security perspective.

As per NIST’s guidelines, security testing needs to be driven by developers, carried out early and often - the time to start is now.

Are you ready to Shift Left?
Yes, you are!

Implementing security testing automation into CI/CD pipelines can be made easily with the use of modern DAST tools like Bright, but it requires a certain level of maturity and cooperation across DevOps teams to ensure successful integration.

Shifting left does not have to be an all or nothing response and it is important to start NOW, to ensure that progress is continually made and the organization works to optimize the Shift Left methodology as processes and teams mature.

Hands-on responsibility for application security design and testing tasks is shifting to development and engineering teams. Gartner research reveals that more than half of software engineering leaders are directly responsible for application security, and another third share this responsibility.

Start now and mature later

Irrespective of your maturity levels, the process of Shifting Left can be started today, by initially enabling smaller teams to carry out security testing. Any issues can then be worked out before scaling it up by rolling it out to other teams.

The goal is to have automated scanning on every build or sooner, as part of your unit testing, but shifting left can grow as your teams and processes do too to achieve this ultimate goal.
Leveraging QA

QA can be a vital bridge between development and security and can be used to spearhead your security transformation as you Shift Left.

QA have an intrinsic understanding of the applications and APIs and their existing functional scripts can be leveraged to carry out security testing too.

Supplementing the existing security testing, such as authentication and authorization tests and load testing, with more comprehensive security tests can be achieved by empowering QA teams. They can perform security regression tests, ensuring that previously fixed security bugs do not reappear.

This not only streamlines processes and introduces early security testing, but also cultivates a culture of security best practices by creating security champions within the QA team.

Security testing as part of the CI/CD Pipeline

The ability to run automated tests in the CI/CD enables security testing to keep up with the rapid release cycles and immediately deliver organization-wide efficiencies, cost savings and improved security posture.

Deploying a developer-first DAST tool in the pipeline has a number of benefits for both the engineering and security teams:

- Faster, scope defined tests that do not delay the CI pipeline
- Seamless integration with CI tools
- Enables specific scan policies per environment, target or development team for full control and visibility
- Ease of testing with no requirement to bypass OTP, WAF, CAPTCHA and other aforementioned scan limitations
- Immediate developer and engineering manager feedback of detected issues integrated with ticketing management allows for immediate triaging, prioritization and remediation
- Snapshot understanding of your security posture, with reports continually fed into dashboards to complement compliance
- Having regular snapshots of all of your applications allows analysis of patterns and trends over time, allowing your security team to measure process and effectiveness of tools, training and strategies
- Dramatically reduces the reliance on and cost of manual testing

**BENEFITS INCLUDE:**

- Daily or weekly security scans
- Earlier developer and engineering manager feedback
- Alert on regression tests and rollback mistakes
- Ease of testing with no requirement to bypass OTP, WAF, CAPTCHA and other aforementioned scan limitations when testing in production environments
The ultimate goal...Unit Security Testing

As discussed, Shifting Left is a process, but how far left can comprehensive security testing be implemented?

Software Composition Analysis (SCA) and Static Application Security Testing (SAST) have been at the forefront of early security testing, but they have their limitations, mainly around context.

SAST specifically cannot differentiate between code that is or is not run during execution and regular use of the application and is fraught with false positives.

The traditional single dimensional view of security posture fails to capture the complexity of modern technology and microservices, leading to a lack of context and insight into interactions between the wider application and its microservices and APIs.

SCA alerts for insecure open source libraries often result in false positives and cause developer fatigue, as the code may not actually utilize the vulnerable library method.

While DAST scans are necessary, they can be time-consuming and impact productivity if carried out in the later build phases or by a third-party security team on production.

Shifting Left and enabling developers to perform quick DAST scans in the early stages of development can enhance the security program, as developers can identify and address vulnerabilities in seconds to minutes, instead of hours to days.
UNIT SECURITY TESTING PROVIDES MANY BENEFITS, NAMELY:

- Earlier testing in the SDLC, added to existing frameworks
- Optimized, scope-defined tests on every newly created component / feature is only achievable with unit security testing
- Lightening quick scans against each function
- Reduces complexity of testing authenticated part of the target later in the pipeline
- Immediate feedback to developers while they code
- No ticket opening is required as the developer gets feedback and guidelines immediately
- Removes need for triaging, validation of finding and prioritization of fix
- Ability for developer to easily and quickly re-test after remediation on-the-go
- AppSec intervention is completely eliminated (apart from remediation assistance or advice)
- Developer context switching is eradicated
Why DAST, Why Bright?

DAST, or Dynamic Application Security Testing, is a method of evaluating web applications or APIs for vulnerabilities through simulated attacks. DAST tools, also known as web application scanners, simulate attacks from the outside-in perspective, mimicking the actions of a malicious attacker.

Once a DAST scan is complete, it reports any vulnerabilities it found so they can be addressed. DAST is a critical piece in developing, running, and maintaining secure applications and APIs.

Bright’s DAST solution, designed with developers in mind, sets a new standard in security testing by integrating into unit testing processes. This integration enables organizations to incorporate security testing into their development and CI/CD pipelines, reducing the risk of security and technical debt and promoting early and frequent scans by engineering teams.

With low/no false positives, manual validation of security findings is no longer necessary, freeing up valuable time and resources for both the security and development teams and allowing for faster and more secure releases.

Bright offers a user-friendly and efficient solution for testing applications and APIs (SOAP, REST, GraphQL) that aligns with modern technologies and architecture. With its innovative Business Logic Security Testing, organizations can identify complex security vulnerabilities and reduce the need for manual testing, improving the overall security posture while maintaining full visibility and compliance.
Security testing automation is now considered a necessity, especially in industries such as banking, finance, and insurance. Moving security testing earlier in the development process, known as "Shifting Left," is crucial for adopting a best practice approach to building secure software at scale.

Traditional DAST solutions no longer meet the demands of DevSecOps, which requires a more efficient and automated approach. Bright Security offers a cutting-edge DAST solution specifically designed for DevSecOps, enabling organizations to shift security testing even further left, beyond what was previously possible.