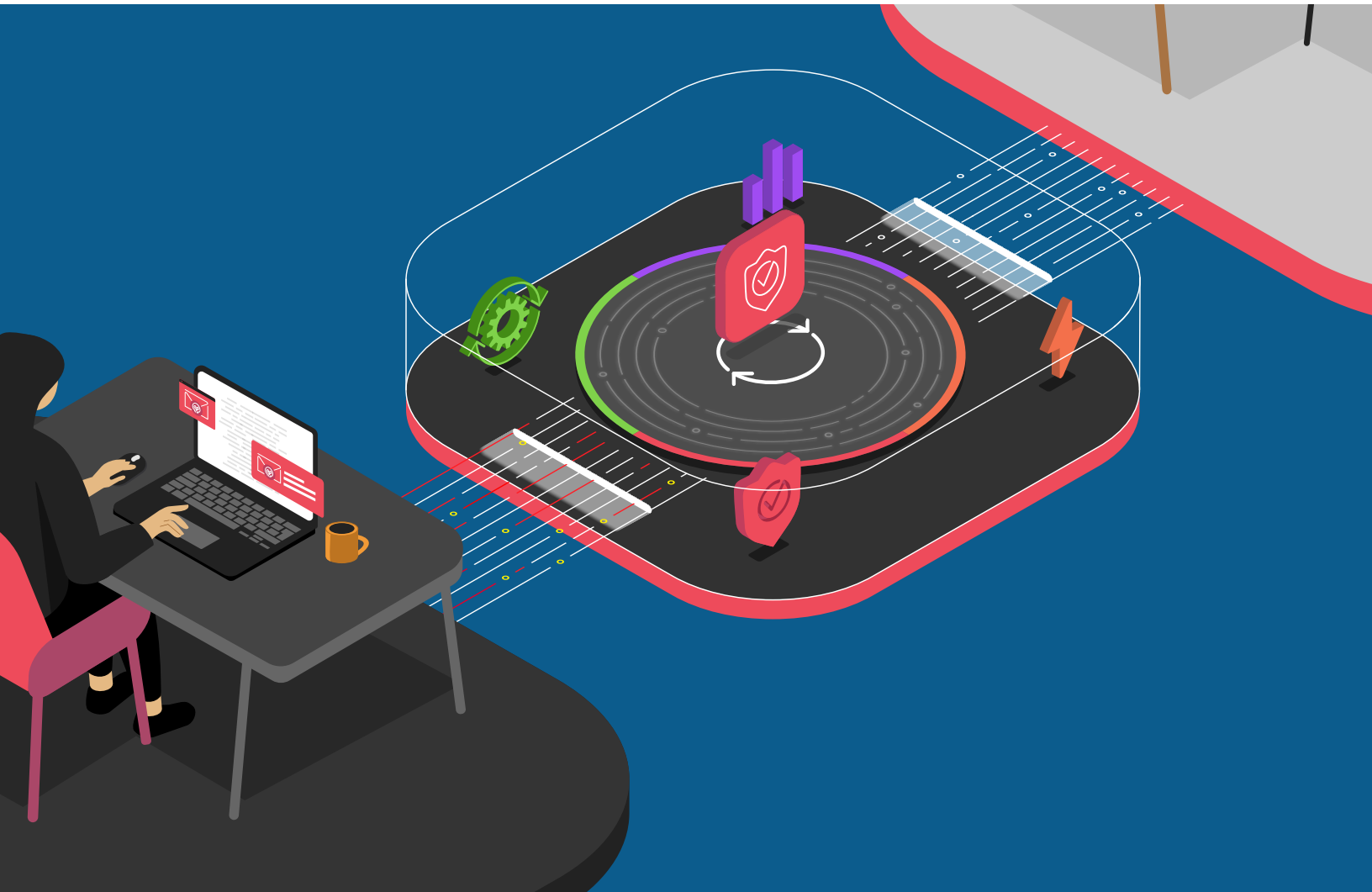
SOLUTION OVERVIEW

Welcome to the Application Era

Today, apps not only enable digital transformation in a business, they are the business.



KEY BENEFITS

Shift Left
Application security is intrinsically integrated into the application development lifecycle, regardless of architecture, cloud, or framework.

Highest Real-World Security Efficacy
F5 protects the most apps with the highest real-world security with less friction—in the network, in the cloud, and in the application architecture—to accelerate digital transformation and protect strategic business outcomes.

Automated Protection
F5 solutions adapt and maintain full efficacy even as attackers retool and evolve to overcome countermeasures without compromising the overall user experience.

To maintain agility while accelerating speed to market, organizations have adopted agile development and entrusted AppDev and DevOps teams to deliver on strategic business imperatives. A developer, with the click of a button, can automate the build, testing, deployment, operation, and monitoring of new code that may change the world.

But What About Security?

Application development has been transformed and is largely automated, but security remains a manual effort. Developers and DevOps practitioners outnumber security professionals by as much as 100 to 1. Time-to-market pressure has caused friction between application and security teams and created the perception that security is a bottleneck. This is a serious dilemma that often results in poor testing, process shortcuts, and ineffective oversight.

At the same time, proliferation of architectures, cloud, and third-party integrations has dramatically increased the threat surface for many organizations. Application vulnerabilities like cross-site scripting (XSS) and injection have been prevalent since the dawn of application security more than 20 years ago, yet attackers continue to discover and exploit them at an alarming rate. Critical application vulnerabilities are released daily, and attackers quickly weaponize them in automation frameworks that continuously scan the internet to discover and exploit them for monetary gain. Open source software in particular is plagued with vulnerabilities—some of which are planted by attackers.

F5 Labs reports that the growing decentralization of application architecture—whether we understand it in terms of third-party widgets, serverless computing or containers, microservices apps, or the increased importance of APIs—raises significant challenges to visibility and security controls that simply weren't there before.[1]

ATTACKERS DO NOT NEED TO UNDERSTAND SYSTEM INTERNALS WHEN EXPLOITS ARE PUBLISHED ONLINE, AND AUTOMATION FRAMEWORKS THAT SCAN FOR NEWLY PUBLISHED VULNERABILITIES CAN BE EASILY UPDATED AND LAUNCHED.

To effectively manage the growing complexity of securing applications across architectures, clouds, and developer frameworks, organizations need consistency.

Open Web Application Security Project

The Open Web Application Security Project (OWASP) was founded in 2001 to persuade business executives and corporate boards of the need for effective vulnerability management. A disciplined approach, including security vendors and community feedback, has resulted in the OWASP Top 10—a list of the most prevalent and critical application vulnerabilities.

KEY FEATURES

- Native integration in application development frameworks improves time to market and reduces friction
- Protection from well-known and emerging threats reduces risk and accelerates digital transformation
- Support for all architectures, form factors, deployment modes, and compliance mandates provides flexibility to support all applications
- Out-of-the-box protection from application vulnerabilities reduces risk and remediation costs
- Dynamic signature feeds to block emerging threats allows security to adapt to the threat landscape
- Automated policy deployment improves effectiveness by implementing and stabilizing security controls earlier in the software development lifecycle
- Integration into the CI/CD pipeline reduces friction between development and security teams
- Declarative API simplifies policy deployment and maintenance by abstracting complexity and reducing developer overhead
- API-driven deployment and maintenance simplifies policy management and change control across multiple architectures and clouds

XSS and Injection have been in every OWASP Top 10 list since its inception. XSS vulnerabilities are present in two-thirds of all applications.[2] And although SQL injection gets a lot of attention, there are many other forms, including LDAP, XPath, OS commands, XML parsers, SMTP headers, Expression Language, and ORM queries. Legacy code is particularly vulnerable.

A growing threat that OWASP is monitoring is the pervasiveness of open source software. Although open source software significantly speeds development, it also changes risk management because controls that are common in custom software developed in house, such as static code analysis (SCA), are not possible with open source software.

In 2017, attackers exploited an insecure deserialization vulnerability in unpatched instances of the open source software Apache Struts. While insecure deserialization vulnerabilities in general are not easily exploited, due to the need to intimately understand system internals, they can result in remote code execution with devastating consequences.

The Rise of Automation and Cloud

The rapid evolution of technology is changing the way that organizations do business—and the steps they must take to keep their businesses safe and secure. Today, 80% of organizations are executing on digital transformation—with increasing emphasis on accelerating speed to market—and 73% are automating network operations to boost efficiency.[3]

The explosion of applications and speed to market is also driving fundamental changes to risk management. Network engineers may not be deploying infrastructure. DevOps teams can easily create virtual and ephemeral infrastructure using emerging architectures like serverless computing in a mature cloud environment—automating everything from code build to service deployment. These changes in roles, responsibilities, and ways of working in the application development cycle can often leave security behind.

At the same time, attackers are getting creative in their methods by leveraging readily available tools and frameworks to automate and scale their attacks.

Additionally, many businesses want to adopt one or more cloud providers or a favorite vendor for risk management and business continuity. The challenge is that cloud providers lack universal security. This often leaves those in charge of security confused about what is secured or not, and the nuances can result in a vulnerability—commonly a security misconfiguration (for example, see the AWS shared responsibility model).

F5 Labs discovered a threat campaign where attackers leverage automation to find injection vulnerabilities (for example, CVE-2011-4107 and CVE-2013-3241 in open source PHP) and exploit weak authentication portals and/or outdated MySQL databases to steal sensitive data and set up a beachhead for further attacks.[4]

F5 Labs and the Verizon Data Breach Investigations Report show how threat actors exploit a vulnerability to gain access to cloud-based email servers, send internal phishing emails linking to fake login forms, and harvest credentials for use in credential stuffing attacks.[5]

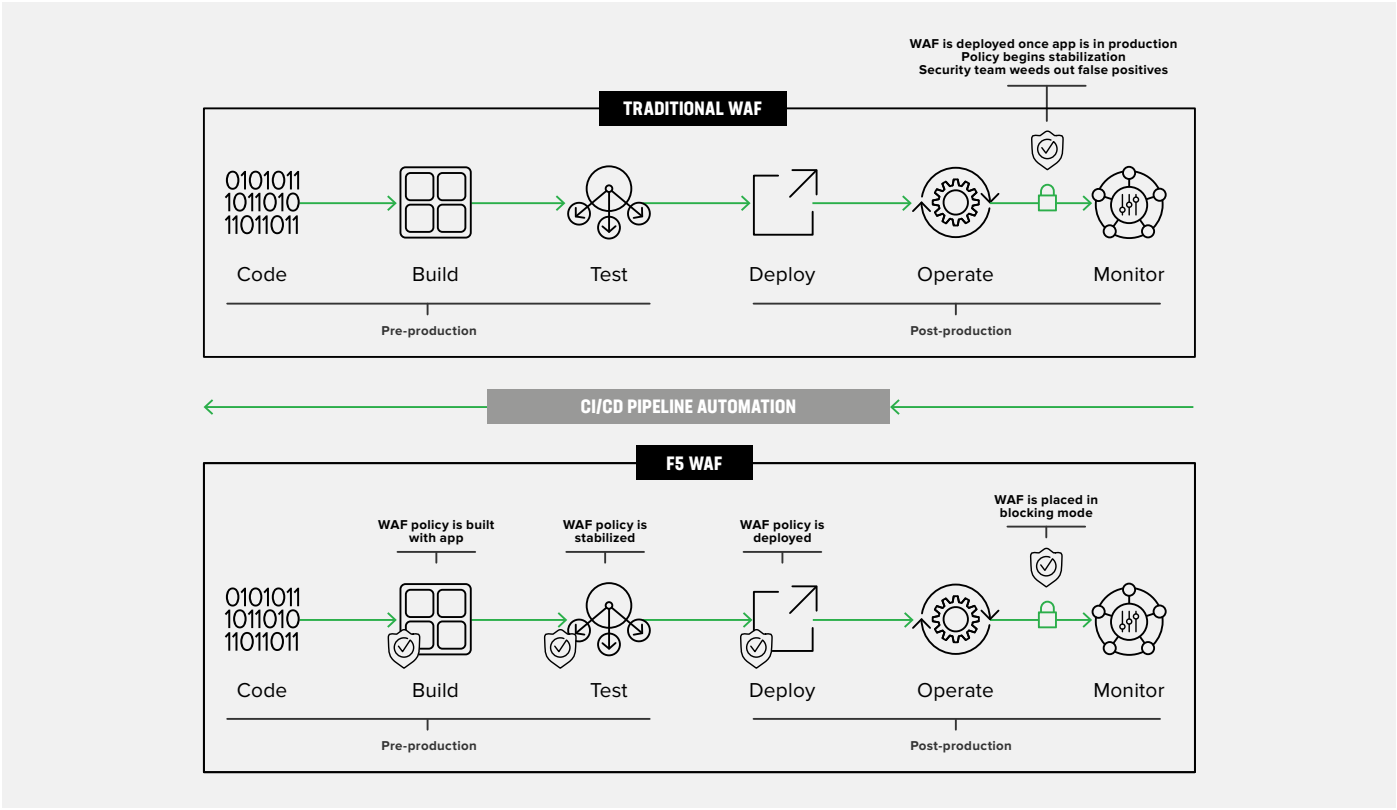


Figure 1: CI/CD pipeline automation can improve time to market while reducing risk and producing better business outcomes

APPLICATION SECURITY NEEDS TO SHIFT LEFT AND BE INTRINSICALLY INTEGRATED INTO THE APPLICATION DEVELOPMENT LIFECYCLE, REGARDLESS OF ARCHITECTURE, CLOUD, OR FRAMEWORK.

The Need to Shift Left

Risk is shifting due to the way applications are built and deployed. Therefore, security needs to shift to stay ahead of the curve of application vulnerabilities. Visibility into attacks is as important as ever, given the creativity of attackers who may initially exploit a vulnerability but eventually progress to sending a well-crafted phishing email that tricks a user into revealing credentials. Attackers can later use that information in a large-scale credential stuffing attack, which can result in unauthorized access, a data breach, an account takeover, and fraud.

Virtual patching and dynamic application security testing (DAST) are also as important as ever, but organizations need a paradigm shift in the way application security is implemented. Instead of constructing a security policy after an application is launched and vetting out false positives to stabilize the policy, application security needs to shift left and be intrinsically integrated into the application development lifecycle, regardless of architecture, cloud, or framework.

The most effective application security is automated and integrated. Automation can lower operational expenditures (OpEx) and reduce strain on critical security resources during application release, deployment, and maintenance. Automated policy deployment can improve effectiveness by implementing and stabilizing security controls earlier in the software development lifecycle (SDLC), leading to higher efficacy with less human intervention.

Native integration into application development frameworks and continuous integration/continuous delivery (CI/CD) pipelines can improve time to market while decreasing risk, reducing friction between development and security teams, and leading to better business outcomes.

Integration into developer tools, through API-driven deployment and maintenance, simplifies policy management and change control across multiple architectures and clouds by abstracting complexity and reducing developer overhead.

Conclusion

F5 APPLICATION SECURITY – PROTECTION WITHOUT FRICTION THAT ADAPTS AS APPLICATIONS AND ATTACKERS EVOLVE.

Today, apps are the business, which makes threats to applications the biggest risk to the business. Modern, decentralized, application architectures have expanded the threat surface, automation has increased attacker effectiveness, and the fallout from cybercrime continues to grow.

The solution is clear. Rather than delaying the release of an application, shift security to the left so that it occurs earlier in the development lifecycle and is integrated into the process. That is the best and most effective way to proactively mitigate app vulnerabilities, increase app security, and reduce the risk to the business.

F5 protects apps by effectively protecting against various threats with less friction to accelerate digital transformation and protect strategic business outcomes.

To learn more, contact your F5 representative, or visit F5.

Endnotes

[1] 2019 Application Protection Report, found at
https://www.f5.com/labs/articles/threat-intelligence/2019-application-protection-report

[2] OWASP Top Ten 2017 A7:2017-Cross-Site Scripting (XSS), found at
https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A7-Cross-Site_Scripting_(XSS)

[3] The State of Application Services in 2020, found at
https://www.f5.com/state-of-application-services-report

[4] Application Protection Report 2019, Episode 1: PHP Reconnaissance, found at
https://www.f5.com/labs/articles/threat-intelligence/application-protection-report-2019--episode-1--sensor-networks-r

[5] 2019 Application Protection Report, found at
https://www.f5.com/labs/articles/threat-intelligence/2019-application-protection-report

300SEC | 557236648