CyberRes

White Paper

Application Security Top Trends: 2021

There’s the ongoing shift where modern development is more dynamic than ever, with increased velocity and complexity. These top trends fit into a modern development framework where security is developer-driven and focused on actionable results that enable digital innovation.

Top Application Security Trends

Application Security is evolving to fit modern development environments. This list is a mix of larger trends and cool new features. What would you add to this list?

We see the migration to APIs, microservices, IaC, and are committed to shrinking our security tech stack while increasing capabilities.

Introduction

In any annual trend list, there should be few surprises. Most trends are a continuation of what was important just a month or year ago, after all! At Fortify we have a holistic AppSec vision that is based on being excellent on foundational elements. This includes broad and accurate language coverage; an integration ecosystem that fits AppSec with as little friction as possible into the tools our customers are already using; and end-to-end Application Security with a hybrid ability between SaaS or on-premises.

There's the ongoing shift where modern development is more dynamic than ever, with increased velocity and complexity. We see the migration to APIs, microservices, IaC, and are committed to shrinking our security tech stack while increasing capabilities. Many things on this list reflect what our customers are asking us to focus on and deliver. And most of these topics deserve more than a few paragraphs. We'd love to hear what trend or detail you would have included.

With that, here is our list of key Application Security trends for 2021.

1. AppSec Tooling Becomes Embedded in the Devops Toolchain
AppSec teams will have less influence in SAST tooling in the DevOps toolchain. Development organizations have pushed back for years and now some commercial vendors offer hyper-convenient scanning. This embedded security scanning discovers a fraction of the vulnerability issues a more robust AppSec tool can find, but delivers convenience and cost savings while helping organizations check the compliance box.

Cloud platform vendors are also offering security tools that make it hyper-convenient for development teams to avoid using AppSec's tools. For instance, Google Cloud offers "Web Security Scanner" that can be kicked off via API and is free to use within certain usage thresholds. Once again, this checks a low-threshold compliance box at a low price.

2. Container Security Is the Battleground for Securing the Software Supply Chain

There's heightened awareness of the software supply chain with the Solarwinds hack last year, and with Equifax and Struts in 2017. Containers are now the ultimate battleground here as multiple software supply chains converge when building and deploying containerized apps. A 2020 report from the Cloud Native Computing Foundation noted that 92% of surveyed organizations used containers in production, up from 84% the previous year.

Security and risk management practitioners have to deal with container security issues around vulnerabilities and compliance. Based on emerging best practices, here is a 4-pillar strategy around shifting container security left to complement post-deployment runtime solutions.

1. Secure Base Image, with centralized base image selection, CVE scanning, and hardening, which is key to enterprise scale and consistency, and particularly important with the rising threat of supply chain attacks. The hardened images are then pushed to an enterprise registry with approved images.
2. Secure the application itself using standard AST methods like SAST, SCA, and DAST, as with any other app.
3. Build images with the ability to scan Dockerfiles, which define the layers used to build the containerized app and are maintained within the app's source code. Developers incorporate best practices from the CIS benchmark, such as no opening up SSH in a container, or forgetting to ensure the container runs as a non-root user. You can also detect traditional software vulnerabilities such as hard-coded passwords which are unfortunately a common mistake in Dockerfiles.
4. Orchestration enables a team to build application services across multiple containers, schedule containers across a cluster, scale the containers, and then manage the containers' health. Orchestration may include code in Kubernetes Helm charts, Docker Compose files, and cloud-native container engines.

3. IaC Security Adoption Grows

Infrastructure as Code (IaC) is the process of managing and provisioning computer data centers through machine-readable definition files, rather than physical hardware configuration or interactive configuration tools. IaC technology is being adopted more and more by organizations to allow for rapid provisioning and cloud deployment of environments.

With the power of IaC comes the responsibility of managing security risks, which can be introduced if best practices are not followed. Poor security decisions when utilizing IaC technologies will result in the rapid and automated deployment of an insecure production environment ready for compliance violations and data breaches! Attention to detail becomes crucial when nearly everything about infrastructure deployment is automated. Manual security assessments performed on a regular basis are no longer sufficient.

With the power of IaC comes the responsibility of managing security risks, which can be introduced if best practices are not followed.

Today, IaC templates are used to provision compute and containerized instances by including base images stored in trusted registries. This presents an opportunity to detect and address any known vulnerabilities in such base images early and dramatically reduce the cost of remediation. Hardcoded secrets or credentials is a common malpractice that involves storing plain text credentials, such as SSH keys or account secrets, within source code. This risk can enable unauthorized privilege escalation and lateral movement during a breach. It is very difficult to trace and contextualize hardcoded secrets in runtime environments. Unfortunately, provisioning and managing infrastructure through code makes it easier to hardcode secrets within it.

APIs (Application Programming Interfaces) are a key part of digital transformation strategies, and securing those APIs is a top challenge.

Integrating both static and dynamic analysis of IaC templates into the CI/CD pipeline alongside the rest of the application, can provide a more complete view of the actual risk. However, delivering this new experience to developers requires the right guardrails separating what developers can and cannot do.

4. Vulnerability Management Takes a Step Forward

Vulnerability management provides a comprehensive view of risk from applications and their supporting infrastructure. By aggregating all vulnerabilities from different tools and parts of an organization into a single view, organizations can provide more holistic AppSec analysis and reporting. This can improve prioritization and aid proving compliance requirements are met.

Development org leaders and executives mainly care about the risk of the environment. Tools that aggregate information from multiple sources and present that risk in a rollup view have an advantage over tools that offer one perspective about a focused area of the software. AppSec tools will face pressure to natively offer this functionality at enterprise scale.

5. SAST and DAST Become Truly Integrated

SAST and DAST already complement each other. 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. While it is important to identify vulnerabilities early in the SDLC using technologies like static analysis, it is critically important to create feedback loops that can identify when those findings surface in running environments via a DAST scan. An organization that identifies findings like XSS early in the SDLC, and continues to detect those issues in production, can focus their training and development resources on addressing systemic problems.

True SAST and DAST integration means SAST and DAST tools integrate into a single developer-centric platform with a single management console. Unified vulnerability management creates feedback loops. A unified application security vulnerability management platform is not only critical in terms of the simplified prioritization and triage workflows that it introduces, but also in terms of the patterns that can be gleaned from the data. More intelligent scanning means DAST validation of SAST findings, and DAST tuning by SAST results.

6. Cloud Native App Security Requires a Continuous Application Security Approach

The modern software stack includes cloud-native elements of the architecture. CNFC defines cloud native as "cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach."

AWS, Azure and GCP dominate the market. Each offers similar functionality and SDKs for interacting with the cloud infrastructure components, along with a further level of abstraction with serverless functions. While cloud computing offloads many tasks to the cloud provider, the end user organization retains responsibility for securing the data that goes into the cloud.

This requires a continuous application security approach that fits a dynamic environment.

- SAST Rulepacks should detect vulnerability categories specific to the cloud-provider's frameworks and cloud-native apps out of the box.
- Cloud native apps are increasingly incorporating Infrastructure as Code (IAC) to programmatically define the cloud infrastructure upon which they operate.
- A trend is to expand IaC coverage to seamlessly detect not only configuration issues (with ARM, AWS CloudFormation and other templates), and more complex structural, control flow, and data flow issues beyond the capabilities of basic IaC scanners.

Application security needs to be embedded earlier in the software development lifecycle and must produce less friction to development teams.

7. API Security Remains a Top Challenge

APIs (Application Programming Interfaces) are a key part of digital transformation strategies, and securing those APIs is a top challenge. APIs are a rapidly growing attack surface that isn't widely understood and can be overlooked by developers and application security managers.

Modern cloud native apps typically employ a distributed architecture, services/microservices, and serverless functions. These components communicate with each other, and end users, with APIs, creating the need to assess security at the component and system levels. At the component level, interservice communication can use a variety of protocols, ranging from HTTP to RPC. At the system level, an API gateway is typically used to consolidate individual service APIs into a unified business app API, based on HTTP, usually REST.

The first step in securing APIs is to incorporate SAST in the DevSecOps pipeline for each independent component. Then API security incorporates DAST scanning at both the component and the system level APIs where HTTP is utilized. The first step is attack surface discovery, which means providing the endpoints and parameters that constitute the API attack surface (the "what"). In addition to the "what," proper discovery also incorporates how the API is used (the "how"), which is important for the business logic workflows and sometimes complex authentication at the system level of the API gateway.

8. DAST with Functional Testing

How about another AppSec acronym? Functional Application Security Testing (FAST) utilizes DAST with functional testing to create seamless, fully automated dynamic testing as part of DevSecOps pipelines.

The FAST capability integrates into the CI/CD pipeline and detects dynamic vulnerabilities based off the customer's functional tests, whether that be Selenium, Cucumber, or any other technology that leverages http to test an application. FAST technology will allow customers to test the most critical portions of their applications with sub 5-minute scan times, without the complexity of setup and configuration.

9. Susceptibility Analysis to Focus on Exploitable Open Source vulns

OSS is a critical part of DevSecOps. 1.5 trillion OSS download requests were expected in 2020, while there was a 430% YOY growth in cyber attacks targeting open source software projects (Sonatype's State of the Software Supply Chain Report). To succeed in this environment, teams have to focus on those issues that are not only vulnerable but exploitable.

Susceptibility Analysis means quickly illustrating vulnerable components that are directly or indirectly being invoked and thus exploitable. Being able to prioritize open source issues saves time on investigation of known issues, and even more time spent upgrading a library that has almost zero security benefit.

Susceptibility Analysis means quickly illustrating vulnerable components that are directly or indirectly being invoked and thus exploitable.

At Fortify we partner with Sonatype to accomplish this. The way that we collect methods and function signatures is based on the requests that we receive for Sonatype indications of known components. So as you request that Sonatype scan various open source components, we understand that for any of those particular known vulnerabilities that have had updates, meaning that they have been patched, we'll generate a signature for that function or method so that we can see that the function is actually in your own custom code and that you are utilizing that vulnerable component of the dependency. This means you know not just that you have the dependency on your class path but you've actually used it in a way that makes you susceptible to this particular vulnerability.

10. Hacker Level Insight (HLI) Moves DAST into Risk Assessment Posture

DAST scanning will evolve from classic vulnerability detection into more of a risk assessment tool. Hacker Level Insight (HLI) is a technology set that presents developers and AppSec teams with the same set of data that a hacker would typically use to perform reconnaissance and targeting.

Adding dynamically detected component and library reporting to DAST enables DevSecOps teams to use this information to "see what the hacker sees" and in turn prioritize their resources toward the most critical gaps in their application security posture. Developers can use hacker-level insight to better inform their threat model. A foundation to holistic application security is fostering collaboration.

Final Thoughts

The complexity of software and frequency of releases continues to increase. Fortify's Software Security Research team finds that a vast majority (79% in our latest AppSec Risk Report) of web of applications have at least one critical or high severity issue.

Application security needs to be embedded earlier in the software development lifecycle and must produce less friction to development teams. The trends discussed above fit into a modern development framework where security is developer-driven and focused on actionable results that enable digital innovation.

We'd love to hear from you. Which of these trends interests you the most? Are there other AppSec innovations you are watching closely?

About Micro Focus Fortify
Fortify lets you build secure software fast with an application security platform that automates testing throughout the CI/CD pipeline to enable developers to quickly resolve issues, strengthening their cyber resilience. Fortify static, dynamic, interactive, and runtime security testing technologies are available on premises or as a service, offering organizations the flexibility needed to build an end-to-end software security assurance program.

Learn more at
www.microfocus.com/appsecurity

Contact us at **CyberRes.com**
Like what you read? Share it.

CyberRes
A Micro Focus Line of Business