Feature

Alamy

SECURITY > APPLICATION SECURITY

The State of Software Security Testing Tools in 2022

Organizations have begun to ramp up software security testing. Learn about the types of tools on the market, how to choose the right tools, and more.

Karen D. Schwartz | Jul 22, 2022

Supply chain attacks, injection attacks, server-side request forgery attacks – all these threats, and more, prey on software vulnerabilities. Vulnerabilities can range from misconfigurations to faulty design and software integrity failures. Overall, applications are the most common attack vector, with 35% of attacks exploiting some type of software vulnerability, according to Forrester Research.

The focus on software security, along with the proliferation of software security testing tools, has grown over the past few years, thanks in part to supply chain attacks like those on Stuxnet and SolarWinds. And as organizations expand their web presence, there is more risk than ever. Finally, the move toward DevSecOps has encouraged more organizations to include security testing in the software development phase.

Keeping software attacks at bay requires increasing efforts around testing — and not only at the end of development. For those developing software in house, software should be tested early and often. Doing so can reduce delays and extra expenses that occur when software must be rewritten toward the end of a production cycle.

In the case of software developed externally, the wisest approach is to test via multiple methods before putting it into full-scale production.

"It's always easier to prevent problems than it is to find issues during production, so baking in security testing from the beginning makes a lot of sense," said Janet Worthington, senior analyst for security and risk at Forrester.

Mountains of Software Security Testing Tools

One of the most important testing tools to prevent the escalation of threats is static analysis testing.

Also called static application security testing (SAST), this type of testing analyzes either the software code or its application binaries to model the applications for code security weaknesses. It's especially good at rooting out injection attacks. SQL injection attacks are a common attack vector that inserts a SQL query through the input data from the client to the application. It is often used to access or delete sensitive information.

SAST tools also can help identify server-side request forgery (SSRF) vulnerabilities, where attackers can force servers to send forged HTTP requests to a third-party system or device. SAST tools can help catch these vulnerabilities before they reach production.

Another critical testing tool is software composition analysis. These tools help block malicious components from entering the pipeline altogether. They look for known vulnerabilities in all components, including those in open-source and third-party libraries. Vulnerabilities like Log4J have contributed to the popularity of this type of testing tool. Forty-six percent of developers now use software composition analysis tools for testing, according to Forrester.

Other important types of software security testing tools include:

- **Vulnerability Scanning**: While these tools focus on finding application security vulnerabilities across the board, there are also specialized versions for finding weaknesses in web applications. They are particularly useful for finding threats like SQL injections, path traversal, insecure server configuration, command injection, and cross-site scripting

teams plan to use DAST before software releases.

- **API Testing**: APIs are everywhere today. While APIs aren't always a top concern, they aren't immune to security threats. Yet Gartner finds that unmanaged and unsecured APIs create plenty of vulnerabilities, managed only by API security testing and API access control.
- **Interactive Application Security Testing (IAST)**: This method tests software for vulnerabilities during execution, using sensor modules to monitor software behavior during the testing phase. If IAST detects a problem like SQL injection or cross-site scripting injection, it sends an alert. As a newer type of testing, IAST is often done by teams that already perform static and dynamic testing. It tends to have lower false positive rates than other types of testing.
- **Penetration Testing**. Also known as ethical hacking, pen testing involves testing applications for vulnerabilities and susceptibility to threats, usually by an external party. Pen tests can uncover many things, from software bugs and configuration errors to supply chain attacks.

Depending on the type of threat, the platform, and other factors, organizations may choose to employ various types of testing tools. Some applications may also need testing tools that aren't on the list above. For example, an application that includes cryptographic signing will probably require a cryptographic analysis tool. That's why today, more than ever before, it's important to use more than one type of software testing tool.

"If you want to be as thorough as possible, you'll want to do SAST testing to find vulnerabilities in source code, SCA for open-source components and DAST to test the running web application," said Ray Kelly, a fellow at Synopsys, which provides software security and testing tools. "It's really about finding the right tools for your specific situation."

How to Choose Software Security Testing Tools

There is no shortage of tools, and it can be confusing to sift through the options. Overall, there are open-source tools, best-of-breed tools from vendors, and proprietary software testing platforms.

Open-source tools tend to be very tactical in nature, focused on one thing. Examples include OWASP ZAP, a free web application security scanner; Snyk's free code quality and vulnerability checker; SQLmap or Metasploit for penetration testing; SonarQube for code security; and FOSSA for open-source dependency testing.

There are, of course, many best-of-breed tools available for a fee from various vendors.

In most cases, organizations would do best to blend different types of tools from different sources, said Aaron Turner, a vice president at Vectra AI, a threat detection and response vendor. “If you combine a software testing platform with select best-of-breed testing tools, whether open source or proprietary, you can be sure to hit all of your marks, because there is no one platform that can do everything.”

If budget is an issue, Worthington recommended starting with the free version of a testing tool, which many vendors now offer. For example, Snyk, which is known for its software composition analysis tool, has a free open-source version. After the tool has proven valuable, the organization can decide whether to pay for the full-featured version.

Advice From the Experts

Know your team and its capabilities before diving into software security testing, Kelly advised.

“In many cases, software development [or evaluation] teams are overwhelmed by features, product requests, and agile deployment methodologies,” Kelly explained. “Often, they are shipping a new product every week, if not every day, and sometimes security takes a backseat. It’s worth taking the time to really analyze what applications are actually running in your environment today, what their risks are, and what the threat landscape is. Take the time to take that inventory and get a baseline.”

And before committing to any testing tool or methodology, make sure you’re considering the relative importance of the software in your environment. “If you’re a natural gas pipeline operator and you rely on a specific piece of software to keep the pipeline running, you’ll probably spend a lot more time and effort testing that piece of industrial control software than you would testing Wordpress, which runs your website,” Turner said.

Finally, it’s important to keep up with developments in software security. That means not only subscribing to relevant blogs and podcasts, but staying on top of government advisories (e.g., via the Cybersecurity and Infrastructure Security Agency) and NIST’s National Vulnerability Database.

About the author

Karen D. Schwartz is a technology and business writer with more than 20 years of experience. She has written on a broad range of technology topics for publications including CIO, InformationWeek, GCN, FCW, FedTech, BizTech, eWeek and Government Executive.

0 COMMENTS

RECOMMENDED READING

Meet BLOOM: The 'Most Important AI Model of the Decade'

Third-Party Attacks on the Rise as Organizations Struggle with Security

US Investigating 'Significant' Breach of American Court System

Let's Unpack the 10 Immutable Laws of Security Administration