Bright

2024 Guide to Application Security Testing Tools

Bright Security White Paper

www.brightsec.com

Version 2.0 06/12/23

Introduction

Welcome to the world of 2024, where generative AI technologies such as ChatGPT are rapidly advancing and cybersecurity continues to be a critical priority. In this chaotic environment, developers are inundated with new technologies and demands, making it essential to establish a strong foundation for your code to ensure long-term success.

Cybersecurity is a never ending chase for a perfect state of security that does not exist. And yes, it is true that, no matter how hard you try, security gaps are unavoidable.

> “If you think you know-it-all about cybersecurity, this discipline was probably ill-explained to you.”
>
> - Stephane Nappo

Even so, continuous education is key and so we have developed this guide to give you an all in one resource about AppSec tooling that will help you secure your organization in 2024 and beyond.

Indeed, our comprehensive guide will help you fully understand the advantages and disadvantages of various application security testing tools in the modern environment. By the end, you'll have a clear understanding of the direction you may want to take.

So, without further ado, let’s get into it!

MAJOR RECENT SOFTWARE BREACHES AND HOW THEY HAPPENED

The key to prepare your organization for potential attacks is to first understand your adversaries. How do they operate? What are the weaknesses they are looking for? What kind of damage can they inflict? Understanding the answers to these and similar questions is the first step in adapting and planning your software security strategy.

Hence, we have come up with a list of 5 notable and recent attacks with some color around the attack.

T-Mobile Data Breach (May 2023):

T-Mobile has suffered yet another data breach, this time affecting around 800 of the telecom provider's customers. According to recent reports, customer contact information, ID cards, and/or social security numbers were scraped from PIN-protected accounts, as well as other personal information pertaining to T-Mobile customers. Unfortunately, this is the company's second data breach of the year. The first one, which took place in January, affected 37 million customers. T-Mobile was also breached in December 2021 and November 2022.

Sharp HealthCare Data Breach (February 2023):

Sharp HealthCare, which is the largest healthcare provider in San Diego, California, has notified 62,777 patients that their personal information was exposed during a recent attack on the organization's website. Social Security numbers, health insurance data, and health records belonging to customers have all been compromised, but Sharp says no bank account or credit card information was stolen.

Twitter Data Breach (July 2022):

A hacker that goes by the alias 'devil' posted on hacking forum BreachForums that they had the data of 5.4 million Twitter accounts for sale. The data stolen includes email addresses and phone numbers from "celebrities, companies, randoms, OGs, etc". The vulnerability itself was discovered months earlier and allows any party without any authentication to obtain a twitter ID of any user by submitting a phone number/email.

Indian Council of Medical Research Data Breach (October 2023):

Around 815 million Indian citizens may have had their COVID test and other health data exposed to a huge data breach. A US security firm first alerted the Indian authorities in mid-October after a threat actor going by the name of "pwn0001" claimed to have the names, addresses, and phone numbers of hundreds of millions of Indians for sale.

Optus Data Breach (October 2022):

Australian telecoms company Optus – which has 9.7 million subscribers – has suffered a "massive" data breach. According to reports, names, dates of birth, phone numbers, and email addresses may have been exposed, while a group of customers may have also had their physical addresses and documents like driving licenses and passport numbers accessed. The attackers are thought to be a state-sponsored hacking group or some sort of criminal organization and breached the company's firewall to get to the sensitive information. Australia's Information Commissioner has been notified.

[1] https://www.cshub.com/attacks/news/54-million-twitter-accounts-reportedly-on-sale-in-hacking-forum

THE IMPORTANCE OF APPLICATION SECURITY TESTING

From the above descriptions of attacks, it should be clear that prevention is better than attack remediation. Indeed, the possible costs of a cybersecurity attack can be very high and the incidence rate is growing rapidly. According to a recent IBM report titled Cost of a Data Breach Report 2023, the average cost of a data breach is $4.45 million and 51% of organizations are planning to increase their security investment:

$4.45 million

Average cost of a data breach; a new record

51%

Percentage of organizations planning to increase security investments as a result of a breach

With a growing number of malicious internet users, online applications and user data are increasingly at risk of theft and exploitation. The reality is that every internet user could potentially be monitored, making it crucial to take precautions against attackers who are constantly lurking in the shadows, waiting for the perfect opportunity to strike.

More than 99% of technologists say applications in production contain at least four vulnerabilities according to a 2023 report by Contrast Security. Additionally, half of security professionals report that developers fail to identify 75% of vulnerabilities according to avGitLab Global DevSecOps survey.

4

Average number of high risk vulnerabilities in each released application

$8 Trillion

The total cost of cybercrime in 2023 according to a eSentire study

It's important to note that many of the biggest app breaches last year could have been prevented with application security testing. This type of testing is crucial in identifying and addressing vulnerabilities within software applications.

Indeed, lack of the necessary tools and processes to prevent these breaches from being exploited is inexcusable. However, for an application security solution to be effective, it is critical that it quickly and efficiently identifies and addresses vulnerabilities before they become issues.

An effective application security strategy not only protects against potential breaches, but also helps maintain the reputation of a brand, increases customer and investor trust, prevents disclosure of important information, and minimizes threats from both internal and external attacks.

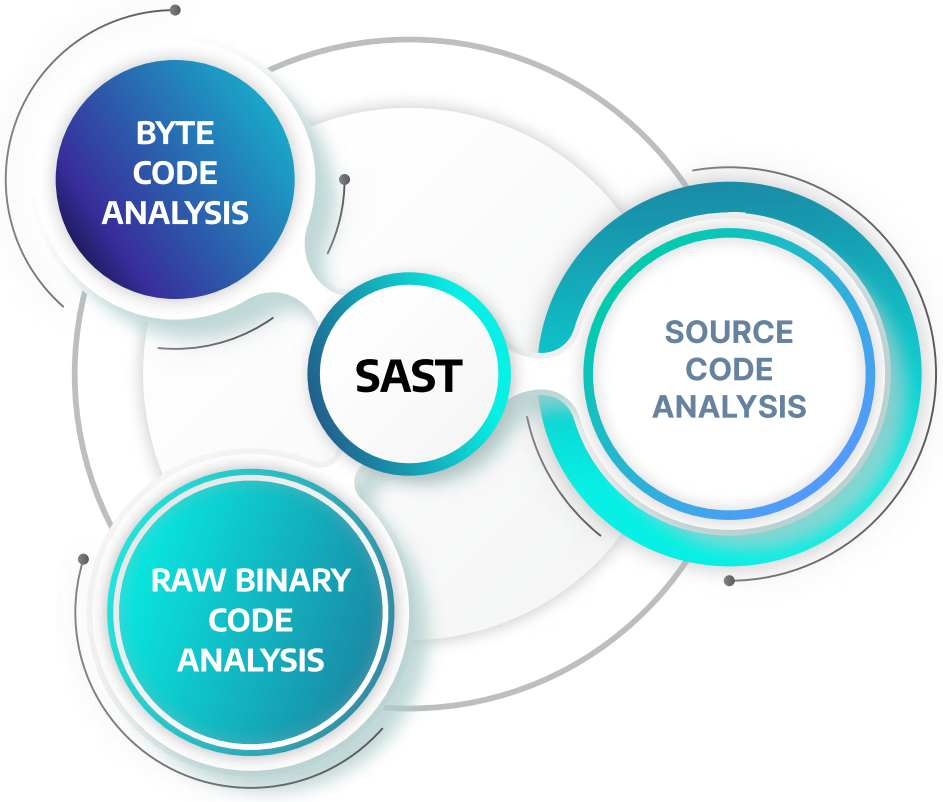
Types of Application Security Testing Tools

A. STATIC APPLICATION SECURITY TESTING (SAST)

Static Application Security Testing (SAST) is a white box testing technique that scans your source code for potential vulnerabilities. SAST tools are popular because they can detect vulnerabilities early on in the development process. It's a fast and automatic way to test your code, capable of scanning through thousands of lines of code in just a few minutes. This makes it an ideal choice for developers who want to ensure their work is secure and up-to-date.

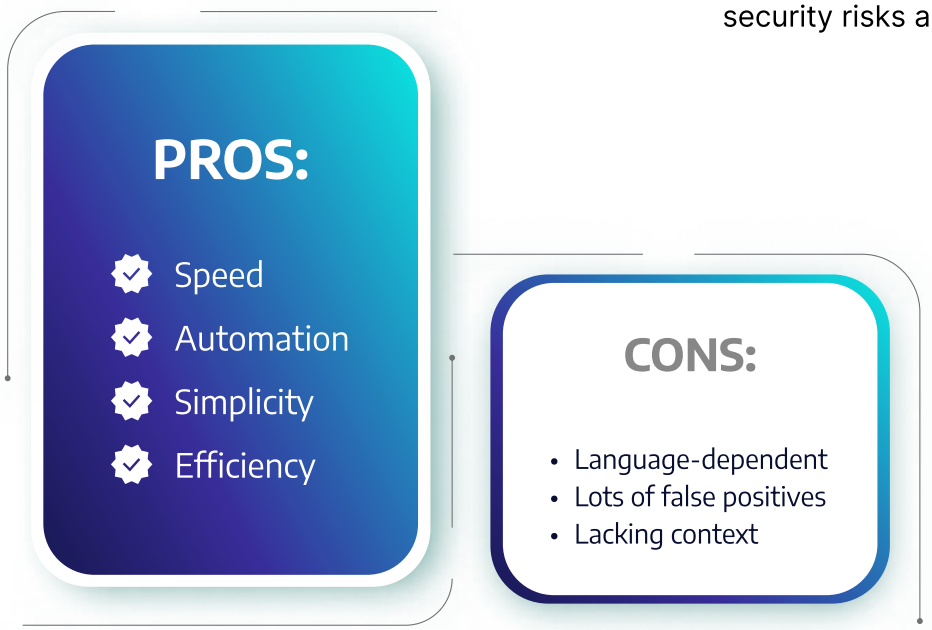
SAST is particularly helpful for developers who are just starting out, as it can identify poor code quality and risky coding practices that could save time and money in the long run. Although issues like these may have a lower priority compared to publicly visible vulnerabilities, they are still a crucial aspect of security.

The main reason SAST is considered developer-friendly is that you can use it during QA (quality assurance), add it to your CI/CD pipeline, as well as utilize it in your IDE (integrated development environment). This means that you could find potential vulnerabilities before even plugging the code into the version control system.



Unlike DAST, static application security testing needs full access to your source code (making it a white-box security testing solution) in order to run properly. However, this also means that there is a big flaw in the SAST system - it doesn't work well in dynamic environments. This is becoming a bigger problem with each passing day, considering the fact that most applications nowadays are built dynamically.

This creates a whole world of problems. The first, and perhaps the most important one, is that SAST is missing context. A lot of apps you may encounter have a separate frontend and backend. Given that SAST cannot analyze those two together means that an abundance of false positives may arise as SAST will recognize an issue on the frontend that's already been solved on the backend and vice-versa.



False positives are SAST's most significant pain point, and you'll often find that it's misleading in its pursuit of perfection.

Another issue is that SAST isn't universal, meaning that it's a code-dependent tool. Generally, there are a ton of SAST tools for the most popular languages such as Python, Java, etc. However, it waters down quickly, and the more you go into the alternatives, you'll have a much harder time in finding a proper SAST solution for your application.

SAST TOOLS

High-performance SAST tools are an essential component of the Shift Left approach to test code as early as possible to save time, effort, and potentially avoid catastrophic security flaws in the future.

Veracode

One of the key features of Veracode's SAST is the extensive and quick feedback that is sent to developers which is a consequence of the tool's integration with well-known IDEs, CI/CD pipelines and work tracking tools. Prior to application deployment, thorough policy scans give developers explicit instructions on identifying, prioritizing, and addressing problems while giving leadership and security teams organization-wide views of application security risks and program performance.

However, just as many other SAST tools, Veracode's SAST tends to generate a high number of false positives which is also indicated by some of the reviews on popular websites such as Gartner, TrustRadius and PeerSpot. Apart from that, the tool might not support some newer languages and frameworks and it is recommended to check the list of supported languages on their website.

PROS:	CONS:
Identification of security vulnerabilities in code	High number of false positives
Detailed reports	Long scanning time
Extensive integration with IDEs and pipelines	Not so friendly UI

Price:

Veracode didn't officially publish their pricing, but according to some user reviews online, it can be costly. However, they do offer different license models to suit different customers. As of now, no free version or free trial is available.

Checkmarx

A pioneer in software security solutions for contemporary software development, Checkmarx provides a complete software security platform that integrates with DevOps by scanning uncompiled source code for security vulnerabilities early in the development life cycle in order to mitigate and address the risk from software vulnerabilities. This static analysis security testing tool provides both source code and open source scanning.

Checkmarx requires a SQL database on which it stores analysis information. Therefore, it isn't very container-friendly and requires a big deployment with a full database server. Apart from that, users generally approve of the UI and developer friendliness of the product. That being said, the pricing seems to be higher than that of competitors.

PROS:	CONS:
User and developer-friendly	High false positives
Analysis of code without compilation	Difficult setup
Language coverage	Pricing

Fortify

Fortify uses a variety of algorithms and a dynamic intelligence base of secure coding procedures to examine an application's source code for any potential risk of harmful or hazardous threats. It enables users to swiftly construct safe and secure software. The solution will also rank the most pressing issues and provide instructions on how users may address them.

Fortify has a very flexible deployment and apart from that, when they write code, users get an interactive manual that gives risk assessments and expected results. While the pricing can be costly, Fortify does offer multiple licensing models.

PROS:	CONS:
Flexible deployment	False positives
Language coverage	Slow at times

HCL Appscan

Another enterprise-level SAST tool, HCL Appscan is nominated as a “Leader” in Gartner Magic Quadrant 2022, and rightfully so. With the help of this potent DevSecOps tool, application vulnerabilities may be quickly fixed during the early stages of software development.

Described as having one of the lowest pricing in the market, the tool also offers a free trial on their website. Despite being susceptible to false positives, it has a very easy setup and offers unique integrations and features, such as QR code scanning.

PROS:	CONS:
Free trial	False positives
Easy setup	Scanning time

Open source SAST

If you still want to prioritize security but are low on budget, there are many open-source SAST’s available. Being open-source offers several benefits, including the constant ability to go in and make changes and, most frequently, a strong community to work with.

LGTM.com scans for common vulnerabilities and exposures (CVE), and the way the data is gathered and displayed is distinctive and powerful. On the other hand, Insider CLI was created to monitor, recognize, and repair the top 10 OWASP-listed web application security issues. There are many other SAST’s that specialize in a specific language or framework, i.e. Bandit for Python or Brakeman for Ruby on Rails.

B. DYNAMIC APPLICATION SECURITY TESTING (DAST)

Dynamic Application Security Testing (DAST) is a form of 'black-box' testing that tests the running application from outside-in. Here, we are attacking the application similarly to a malicious attack, analyzing the front end to find vulnerabilities. DAST has many benefits, such as the test being independent of the application, the ability to find exploitable vulnerabilities, and access to the source code is not required. This limited knowledge approach to testing, without access to internal knowledge of the application or the source code, allows an organization to understand how a malicious actor would approach the application with the same knowledge and information.

By performing a DAST scan early in the Secure Software Development Lifecycle (SDLC), vulnerabilities can be identified and remediated before the code gets released to the public. Human error is inevitable, and without proper testing before deployment, your organization is left at risk of a data breach. However, by finding security bugs early on, organizations can benefit from savings both financially and in terms of brand reputation. After all, the earlier a vulnerability is found, the cheaper it is to fix.

Another way of thinking of DAST is by imagining a security guard for a building. Of course, the first step to ensure the security of the building would be to ensure all locks are working and enabled at the end of the evening. With DAST, we take this one step further. Here, the security guard would attempt to bypass these security measures by picking the locks or using brute force, impersonating someone trying to break into the building. If these attempts are successful, the security guard would alert their superior that a particular lock leaves the building at risk, so they can change the lock or add additional security. By using the same techniques as a malicious actor, organizations can better understand their security posture to protect themselves in the future.

DAST TOOLS

Bright Security DAST

Bright is a developer-first Dynamic Application Security Testing (DAST) scanner empowering Application Security & Development teams to find and fix vulnerabilities at every step in the SDLC, without slowing them down. With Bright, you can secure your applications & APIs for both technical and business logic vulnerabilities at the speed of DevOps, with minimal false positives. The scanner integrates into DevOps environments and enables organizations to run scans as part of their SDLC flows to identify a broad set of technical (9,000+ payloads) and business logic security vulnerabilities early in the development cycle so the vulnerabilities can be remediated early. Bright's DAST scans for multiple protocols across Web, mobile & API and is built for developers to provide compliance on every build by providing remediation guidelines for every vulnerability identified.

The overarching difference is that legacy AppSec solutions were created for the security team, whereas Bright's DAST was built from the ground up to be used by developers, QA and DevOps professionals so while security professionals benefit from the solution as well, it enables organizations to liberate application security and have developers, QA & DevOps professionals use it and shift security to the far left.

Invicti (formerly Netsparker)

This is an effective, automated web application security analyzer that may be used to scan any web application no matter what technological stack or development framework is being utilized. Both a desktop and cloud version are available.

The scanner increases automation and scalability by producing a proof of exploit after discovering a vulnerability that verifies it as not a false positive.

A comprehensive variety of third-party integrations make it simple to integrate this solution with your current SDLC infrastructure.

Invicti offers three plans, Standard, Team, and Enterprise and their Sales needs to be contacted to receive a quote for pricing details. A demo is also available on request.

PROS:	CONS:
Easy set-up and friendly UI	Expensive and limited number of plans
Integration into current workflow	Long scanning time
Excellent customer service	Desktop version consumes a lot of resources

Acunetix

Created by the company Invicti, both a desktop and cloud version are available. Acunetix will scan your website, find vulnerabilities, and let you address them before they are exploited. 6500 vulnerabilities, including SQL Injections and XSS, can be found by Acunetix and all varieties of Single-Page Applications (SPAs) with a lot of HTML5 and JavaScript can be scanned with it as well.

Invicti hasn't officially released the pricing for Acunetix, but according to Trustradius, the price depends on the number of websites that will be scanned and ranges from $4500 to $26000. That being said, reviewers argue that the pricing has been increasing in recent years and that the product is costly.

PROS:	CONS:
Low rate of false positives	Longer scanning time
Easy scan scheduling	Number of concurrent scans
Regular updates	Pricing

Burp Suite

Known for providing a variety of security tools and having the capacity to find the most recent vulnerability, Burp Suite includes a web vulnerability scanner, scheduled and repeat scan functionality, and CI integration. It offers tools for organizations, testers, and developers and helps in improving security testing and penetration testing skills.

Three pricing plans are available: Community (Free), Professional ($399 per user per year) and Enterprise ($3999 per year), the latter two having free trials.

PROS:	CONS:
Developer-friendly	Weak reporting area
Cost effective	Not so friendly UI

Indusface WAS

You can be guaranteed that no OWASP Top 10 vulnerability or malware will go undetected with the Indusface WAS solution. The solution offers thorough web app malware and vulnerability detection and also detects business logic vulnerabilities. Remedial advice is included in the scan reports so the developers may put fixes into place right away.

Three pricing tiers are available for Indusface WAS: Basic (Free), Advance ($49 per app per month), and Premium ($199 per app per month).

PROS:	CONS:
Detects business logic vulnerabilities	Dated user interface
Easy setup	Longer scanning time

C. INTERACTIVE APPLICATION SECURITY TESTING (IAST)

IAST (Interactive Application Security Testing) is a security tool that combines the security function of Static Application Security Testing (SAST) and Dynamic Application Security Testing (DAST) into one security tool.

IAST and DAST both concentrate on application behavior during runtime. However, IAST analysis is based on a mix of internal application flow analysis, scanning, and black-box testing, and is therefore able to link findings similar to those in a DAST to the source code. It does this by analyzing the code run in tests and then pinpointing the exact place where the vulnerability was found in code. Being a fairly new approach to application security strategies, it does have its drawbacks, one being that it's dependent on the programming language as well as it is known to be slowing down the CI pipeline.

Despite this, some key features make IAST a worthwhile investment. SAST tools tend to generate a lot of false positives since one line of code may indicate a security exploit that was already addressed somewhere else. IAST, on the other hand, detects security flaws while the program is running in real-time and scans code both in production and in development. Scanning code in development moves security checks to the left in the SDLC, where it will be less expensive to remedy them. As mentioned above, it also offers fast remediation by linking lines of code with vulnerabilities making this a key difference with DAST.

Apart from being language or technology dependent, IAST also requires you to build and execute applications which makes it very time intensive and it can slow down processes. IAST also doesn't offer complete code coverage because it only analyzes code run in tests which can miss some key parts of the software that might be executed in production.

IAST is far from a perfect application security strategy and before selecting it, it is crucial to assess your technological stack and operational procedures as with every application security testing technique.

IAST TOOLS

Contrast Assess

Contrast automatically identifies vulnerabilities while developers type code, gets rid of false positives, and offers context-specific how-to instructions for quick and simple vulnerability mitigation. By doing this, application and development teams are able to work more successfully and innovate more quickly.

Contrast's IAST offers in-depth visualization of application components, code trees, and dataflow. It automatically generates simple diagrams that illustrate the application's major architectural components. It also provides developers a mapping of the URL and routes of their software that are executed during the testing phase.

Contrast Security has not provided pricing information, which is a common practice. In order to obtain the pricing, the sales team needs to be contacted.

PROS:	CONS:
Easy set-up	Performance issues
Low false positives	Lack of support to some programming languages
Visualization of application components	Integration with DevOps pipeline for larger enterprises

Synopsys Seeker

With Seeker, users may view the security posture of web apps, which also reveals vulnerability patterns in comparison to compliance criteria like the OWASP Top 10 and the CWE/SANS Top 25. It enables security teams to detect and track sensitive data, ensuring that it is handled securely and is not kept in databases or log files with insufficient or no encryption.

Fast interactive application security testing at DevOps speed is made possible by Seeker's integration into CI/CD workflows. This solution analyzes hundreds of thousands of HTTP(S) requests, finds vulnerabilities, and cuts down false positives to almost zero using unique techniques.

PROS:	CONS:
Low false positives	Low language coverage
Easy setup and simple to use	Pricing model not flexible

Jit.io

Jit is an open source platform that safeguards the full software development lifecycle and combines multiple open source solutions into one single interface. One of the integrations it offers is with the SAST solution Bandit used for Python applications. Jit integrates Bandit into the CI/CD and automatically runs it with every new PR, making it a useful open source IAST tool.

D. RUNTIME APPLICATION SELF-PROTECTION (RASP)

Earlier in the guide, we talked about SAST and its inability to properly contextualize vulnerabilities and issues. Well, this is one problem that runtime application self-protection (RASP) certainly does not have.

Unlike some of the previous testing methods, RASP isn't utilized before code deployment, but rather in production, making it an excellent complementary solution to SAST & DAST. Runtime application self-protection monitors your application, trying to detect malicious behavior and putting an immediate stop to it. All of this happens in real time, providing your application with a consistent protection method.

All of this requires RASP to be embedded within your application. This is exactly what allows RASP to monitor and react toward any potential hostile acts. On the other hand, it does create some drawbacks as well, the biggest of which is that including RASP on your server could very well affect the speed and performance of your application, and some companies even reported performance loss of up to 50% while using runtime application self-protection.

You might be thinking that RASP is everything you'll ever need and more in terms of software security, despite the performance shortcomings - but not so fast! Just like everything in the cybersecurity industry, RASP isn't to be taken at face value. The biggest reason for this is that this solution is primarily focused on common and well-known vulnerabilities.

Despite offering a ton of positives, RASP has a major red flag for a lot of businesses - it's very demanding & difficult to set up. Setting up and maintaining RASP has proven to be a daunting task even for larger IT teams, not to mention the extraordinary cost that comes along with it.

RASP TOOLS

Metasploit

Rapid7 refers to Metasploit, an open source network security program, as the most popular penetration testing framework in the world. It was created to assist security teams in managing security assessments, doing more than just identifying vulnerabilities, and raising security awareness.
Metasploit updates databases of exploits regularly, has segregated workspaces for different projects and has a very intuitive user interface. However, its reporting features need some work and the dashboards aren't adapted for individuals with limited AppSec knowledge.

Nmap

Nmap is a free, open source network discovery, mapper, and security auditing software. According to TrustRadius, Its core features include port scanning, identifying unknown devices, testing for security vulnerabilities, and identifying network issues.
Although being open source, its features and capabilities are very extensive. The scanning is very configurable with many options, and given that it's a command-line utility it can be easily scripted to fit your needs. However, commands can take a long time to complete unless you have comprehensive knowledge of the product and know how to properly utilize the different configurations.

Aircrack-ng

Aircrack-ng is a free wireless network scanner used for network administration, hacking, or penetration testing. It has the ability to show network signals and traffic on WiFi networks. It consists of around 20 utilities and comes pre-installed on Kali Linux making it the perfect tool for penetration testers.
As a system designed for experienced experts, Aircrack-ng is an outdated system with a subpar user interface. There is only a command-line system available for Aircrack-ng; there is no graphical user interface. As a result, the tool is challenging to use and is simple for other systems to defeat.

Choosing the Right Application Security Testing Tool

A. FACTORS TO CONSIDER

Given the wide variety of AST tools, choosing the right one for your team can be a challenge. In this section, we will explore the factors that should be considered when selecting an application security testing tool, helping you to make an informed decision that meets your organization's needs. We will cover key considerations such as effectiveness, false positives, ease of use, compatibility, and team empowerment, providing practical guidance that will help you to select the tool that is right for you.

Integration with existing workflows and tooling. It's essential to evaluate how the security testing tool can integrate with your existing workflows and tooling. For example, if you use a continuous integration/continuous deployment (CI/CD) pipeline, can the tool seamlessly integrate with it? Does the tool support integration with the most popular issue tracking systems?

If you use a specific collaboration tool, does the security testing tool support integration with it? Ensuring that the tool can easily integrate with your existing workflows and tooling can save time and increase efficiency.

Scalability. As your organization grows, your security testing tool needs may also grow. It's important to evaluate how well the tool can scale as your application portfolio grows. Can the tool handle testing a large number of applications simultaneously? Does the tool provide an API that can be used to automate the testing process for a large number of applications? Can the tool handle the load of concurrent users and support high availability?

False positives. False positives can be a significant problem for development teams, as they may cause confusion and distract from genuine security threats. They can also slow down the development process, as developers must spend time investigating each alert. Therefore, it is important to choose a tool that has a low rate of false positives, ensuring that the alerts received are accurate and actionable. By minimizing false positives, teams can focus on genuine security threats and reduce the risk of vulnerabilities going undetected.

Support and community. What kind of support and resources are available from the vendor? Do they offer training and documentation to help you get the most out of the tool? Do they have a strong community of users and contributors who can provide additional insights and support? What is their track record in terms of resolving issues and providing timely updates and patches?

Cost. Finally, cost is always a factor when making any investment. What is the total cost of ownership for the tool over its lifetime, including licensing fees, maintenance and support costs, and any additional resources required to operate and maintain it? Are there different pricing models available, such as subscription or usage-based pricing, that may be more suitable for your organization?

In summary, when choosing an application security testing tool, it's crucial to consider effectiveness, false positives, ease of deployment, compatibility, team empowerment, integration, scalability, and cost. Evaluating these factors can help you make an informed decision on which tool is the right fit for your organization. Remember that the right tool can help you identify and fix vulnerabilities early in the software development lifecycle, reduce risk, and improve your overall security posture.

EVALUATION CRITERIA CHECKLIST

Considering the large volume of information you need to gather and digest, we have come up with a practical checklist that you can use to evaluate a potential security tool with a step-by-step approach. Use this checklist to compare each vendor and technology to make an optimal selection that meets your requirements.

Ease of Deployment and Use:

1) How easy is it to deploy the tool on a scale of 1 to 10?

2) Is it user-friendly and intuitive, or does it require extensive training to use effectively?

☐ Yes ☐ No

3) How quickly can a user master the tool in total hours?

Compatibility:

1) Does the tool work with the technologies, frameworks, and libraries that you use in your applications?

☐ Yes ☐ No

2) Does the tool integrate with your development and testing workflows?

☐ Yes ☐ No

3) How easy is it to deploy the tool on a scale of 1 to 10?

- Source code managers
- Development software
- CI/CD
- DevOps tools

Effectiveness:

1) How effective is the tool in detecting vulnerabilities and threats in your applications on a scale of 1 to 10?

2) Is the detection rate high and can the tool identify a range of vulnerabilities, including hidden and unlinked files?

☑ Yes ☐ No

False Positives:

1) What is the rate of false positives identified?

2) Does the number of false positives significantly slow down remediation?

☐ Yes ☐ No

3) Is there a way to set up rules to minimize the incidence of false positives?

☐ Yes ☐ No

Reporting:

1) How easy is it to understand and act on the reports generated by the tool on a scale of 1 to 10?

2) Does the tool provide clear and actionable reports and analytics that can help identify trends and prioritize vulnerabilities?

- [ ] Yes
- [ ] No

3) Do the reports have a user-friendly UI?

- [ ] Yes
- [ ] No

4) How much customization can be done with reporting on a scale of 1 to 10?

fd

Support and Training

1) Does the vendor provide adequate support and training resources to help your team get up to speed and use the tool effectively?

- [ ] Yes
- [ ] No

1) Does the number of false positives significantly slow down remediation?

- [ ] Knowledge base/documentation
- [ ] Email support
- [ ] Phone
- [ ] Chat
- [ ] Custom training workshops
- [ ] Certification program

Team Empowerment:

1) Does the tool support modern development methodologies such as DevSecOps?

- [ ] Yes
- [ ] No

2) Does it encourage collaboration between developers and security teams?

- [ ] Yes
- [ ] No

3) Does it provide detailed and actionable vulnerability reports that can help developers improve code quality?

- [ ] Yes
- [ ] No

Cost:

1) What is the cost of the tool?

1) How does the cost compare to other vendors?

- [ ] Hight
- [ ] Average
- [ ] Low

3) Does the tool fit within your budget and provide good value for money compared to other tools on the market?

- [ ] Yes
- [ ] No

3) Is the above true when you consider the total cost of ownership, including licensing, training, and support costs?

- [ ] Yes
- [ ] No

C. 2023 APPSEC TRENDS TO TAKE INTO ACCOUNT

In recent years, Application Security (AppSec) has entered the spotlight as countless news headlines showcasing data breaches have left organizations unsure of how to protect themselves. AppSec is critical to increasing security posture and preventing malicious actors from gaining access to business-critical data. But at times, it can be challenging to know where to start.

As technology advances, the cyber attack surface grows with it. Malicious actors are finding new, innovative ways to penetrate applications, leaving all organizations at risk. Although we would love to tell you there is a magic solution to put an end to this trend, this is not the case. Primarily, the growing trend of having a distributed workforce has boosted the attack surface growth as multiple systems and plugins, the utilization of numerous access keys, tokens, machine accounts, and automation can be difficult to regulate without proper measures in place.

With the overhanging, constant threat of a data breach, in 2024 organizations will put pressure on vendors to deliver tools with minimal false positives and that help prioritize efforts. At times, it can be difficult to sift through copious amounts of data to find which vulnerabilities are business-critical. Of course, this is not a good use of company hours as your development and AppSec teams have plenty on their plates as it is. Instead, adopting tools that give your team the freedom to focus on releasing code will enable your organization to gain a competitive edge.

Another way organizations will improve their security posture in 2024 is through a sharp “Shift Left.” As many of us know, shifting left is the philosophy behind introducing security as early as possible in the software development life cycle. As previously mentioned, the earlier a security bug is found in the SDLC, the cheaper it is to fix. After all, having your team drop what they are doing to analyze code from weeks prior looking for vulnerabilities is labor intensive, not to mention the entire sprint gets delayed. Luckily, by shifting left, organizations have found a way to keep up with the speed of DevOps and ensure the code they send to market is free of vulnerabilities. By now, many organizations have integrated security testing throughout the SDLC and are already seeing the benefits. However, in 2024 we will see the remaining organizations jumping on this trend.

Best Practices for Application Security Testing

A. SECURE SOFTWARE DEVELOPMENT LIFECYCLE (SDLC)

The Secure Software Development Lifecycle (SDLC) is a well-known process which enables high-quality software development, at a low cost, in a shortened time. Whether your organization is selling software directly to customers, or developing it to run in your operating system, protecting the bottom line is critical to ensure the security of your organization. The SDLC brings security into each development phase to ensure by the time code hits production, it is ready for market.

These phases are as followed:

1. Planning and requirements
2. Architecture and design
3. Test planning
4. Coding
5. Testing and results
6. Release and maintenance

Despite security making it to the forefront of recent headlines, many still see security as an afterthought. The problem here is that the later a security vulnerability is found and fixed in the SDLC, the more costly it becomes to the organization.

But why does this happen? Put plainly, developers see security as a bottleneck forcing them to re-work code from weeks prior and preventing them from releasing cool new features into the market.

However, if that cool new feature is open to exploitation, it will cause more harm than good. We all know that insecure software puts a business at risk, so why are organizations continuously falling into this trap of pushing insecure software to production?

One of the simplest reasons is that organizations are not equipping themselves with the proper tools to seamlessly integrate security into their pipelines. Security doesn't have to be a burden. By providing developers with the tools they need and allowing the AppSec teams to provide the governance, your organization can find vulnerabilities early in the SDLC with time to remediate prior to reaching production.

B. REGULAR TESTING AND UPDATING

Security updates are important because they correct products' known vulnerabilities, which hackers might use to compromise your devices. It is more difficult for attackers to successfully compromise your devices when new security measures are in place.

Threat actors can manipulate outdated systems using malware, and they can also steal data. Malware has the ability to encrypt files, documents, and other programs, rendering them useless. To defend a device from attacks, security patches close these open doors in the software.

Sharing a network also requires extra caution from users since a compromised device has the potential to unintentionally distribute malware to network users, including coworkers, acquaintances, and family.

So, despite the inconvenience of regular software updates, consider them a precaution for your online security. Next time an update notification appears, don't put it off; click the "Install Now" option right away.

C. TRAINING AND EDUCATION

At the end of the day, you can have all the tools in the world set up by the biggest experts in the industry to protect your application, but the key factor still isn't' there - the people. Human beings are the last line of defense, and they're often the most vulnerable link of your cybersecurity chain.

This is why the importance of cybersecurity training and education cannot be overstated. And it's not just a one-time thing either; security on the web is constantly evolving, meaning that you have to regularly update your employees on the latest cybersecurity trends in order to keep both them and your company safe.

The Compliance Trap

One of the most popular security demands for tech companies nowadays is getting all sorts of different compliance certifications. These certifications have strict rules that enforce certain security measures upon your company, automatically rising your cybersecurity standards in the process.
However, they can often paint a false picture and lull you into a false sense of security. This is due to the fact that most companies think that having a compliance certificate is enough to keep their data safe - it is not! While it does help in solidifying your foundations, compliance will only get you so far, and it's up to your whole organization to make a unified effort in creating a secure environment.

Raising awareness

As we mentioned earlier, enforcing cybersecurity standards isn't a one–time event. In fact, it's a long-term process that involves consistently reminding your organization of the security measures expected of them.
This is why raising awareness about cybersecurity company-wide is crucial, as it empowers every individual to become a stronger member of your organization. It should also be noted that raising awareness doesn't just include grandiose meetings, presentations, and etc. Perhaps even more important than that is doing the "boring" stuff consistently. Things like reminding your coworkers about potential phishing attacks, 2FA, and similar everyday issues can go a long way into making a significant change within your organization - saving you insurmountable time and money in the long run.

Empowering Cybersecurity Champions

Superman, Batman, Ironman move aside - the new superhero is in town! We often hear about discrepancies and misunderstandings between the AppSec team and developers, but that has come to an end. Cybersecurity champion is somewhat an abstract role that took prominence in the past couple of years. In the most simple terms, a cybersecurity champion is a bridge between developers and the security team.

This role requires a lot of knowledge and experience. But even more than that, a security champion has to be someone trustworthy, reliable and someone that the developers can freely communicate with. It's certainly not a role suited for people with heavy schedules or crucial decision-makers. The idea of a security champion is that it's someone involved in daily running of the development, making him omnipresent and able to notice and react to any potential shortcomings in security implementations during the development.

Conclusion

Whether we like it or not, cybersecurity threats are all around, and the issue isn't going away anytime soon. It is on us to constantly grow and adapt to changing environments. Having a single perimeter of defense isn't enough, and there is no magic tool that will miraculously patch up your organization. Rather, it's a joint effort between every single piece of your organization, all working in harmony to create a safe and protected environment.

The old saying that the chain is only as strong as its weakest link holds true in cybersecurity perhaps more than anywhere else.

SUMMARY OF KEY POINTS

It's been a long read, and if you made it all the way to this section, you're on the right track! As an award, we'll summarize the whole whitepaper into the key points you can go back to if you ever want to refresh your memory on all things application security testing.

01

Pick security tool based on your needs

02

Enforce security on all levels of your organization

03

Constantly update & adapt to changing security requirements

04

Empower security champions

05

Security results don't always have trackable ROI, but are crucial to long-term success

FUTURE OF APPLICATION SECURITY TESTING

Technology is changing rapidly these days, and there's no telling what's the next big thing that takes over the internet. This gives you all the more reason to dive deep into application security testing to secure the future of your organization.

With AI running riot, it's perhaps the main point of contention amongst AppSec professionals. Will AI help us enforce security standards or is it going to wreak havoc?

While there isn't a certain answer to this question, AI is certainly the next step of application security testing. In a world where different solutions are based on the involvement of humans, AI might be able to bridge the gap and process the data like a computer, but also interpret the results like an AppSec professional. Even though we're a long way from this, it is worth keeping in mind that it might happen someday and fully change our understanding of cybersecurity.

Sources

Why Is Application Security Testing Important and 5 Essential AST Tools
The importance of application security testing
Why Is Web Application Security Testing Important?

Document to reference:

- https://docs.google.com/document/d/17BTfpd1vfWkbSNDu5gOdRT9ftoBM-jugLgrPzYcIwdw/edit?usp=sharing

Popular SAST Tools (break by paid and open source)

- SAST Tools : 15 Top Free and Paid Tools (2024 update)
- Top 10 Static Application Security Testing (SAST) Tools in 2022

Popular DAST Tools (break by paid and open source)

- 15 Best Dynamic Application Security Testing(DAST) Software in 2024 [Reviewed]
- DAST Tools : 23 Best Free and Paid Tools (2022 update)
- 10 BEST Dynamic Application Security Testing (DAST) Software
- Interactive Application Security Testing (IAST)
- What Is IAST? Interactive Application Security Testing
- Interactive Application Security Testing (IAST)
- What Is IAST: Interactive Application Security Testing

Popular IAST Tools (break by paid and open source)

- Top 5 IAST Tools for 2022
- Best Interactive Application Security Testing (IAST) Software
- Interactive Application Security Testing (IAST) Tools

Penetration Testing
Penetration Testing
What is penetration testing? | What is pen testing?

Popular Penetration Testing Tools (break by paid and open source)

- Top 10 Penetration Testing Tools in 2022
- 17 Best Penetration Testing Tools/Software of 2024 [Reviewed]
- 10 free pen tester tools we highly recommend
- 11 FREE Online Penetration Testing (Pentest) Tools to Test Application Security

How to choose the right application security tools
Which application security tools should you choose?
How to Choose the Right Application Security Tool for Your Organization
How to choose a DAST solution: An 8-step evaluation checklist
WEB APPL ICAT ION SECU R ITY SCANNER EVALUAT ION CR ITER IA
How to Evaluate and Select Application Security Testing Vendors
What is Secure SDLC?
5 reasons software updates are important
Is Application security training essential for any organization's security policy?
3 benefits of AppSec training for your long-term security strategy
Deciding on Investing in Application Security Training for Your Developers?