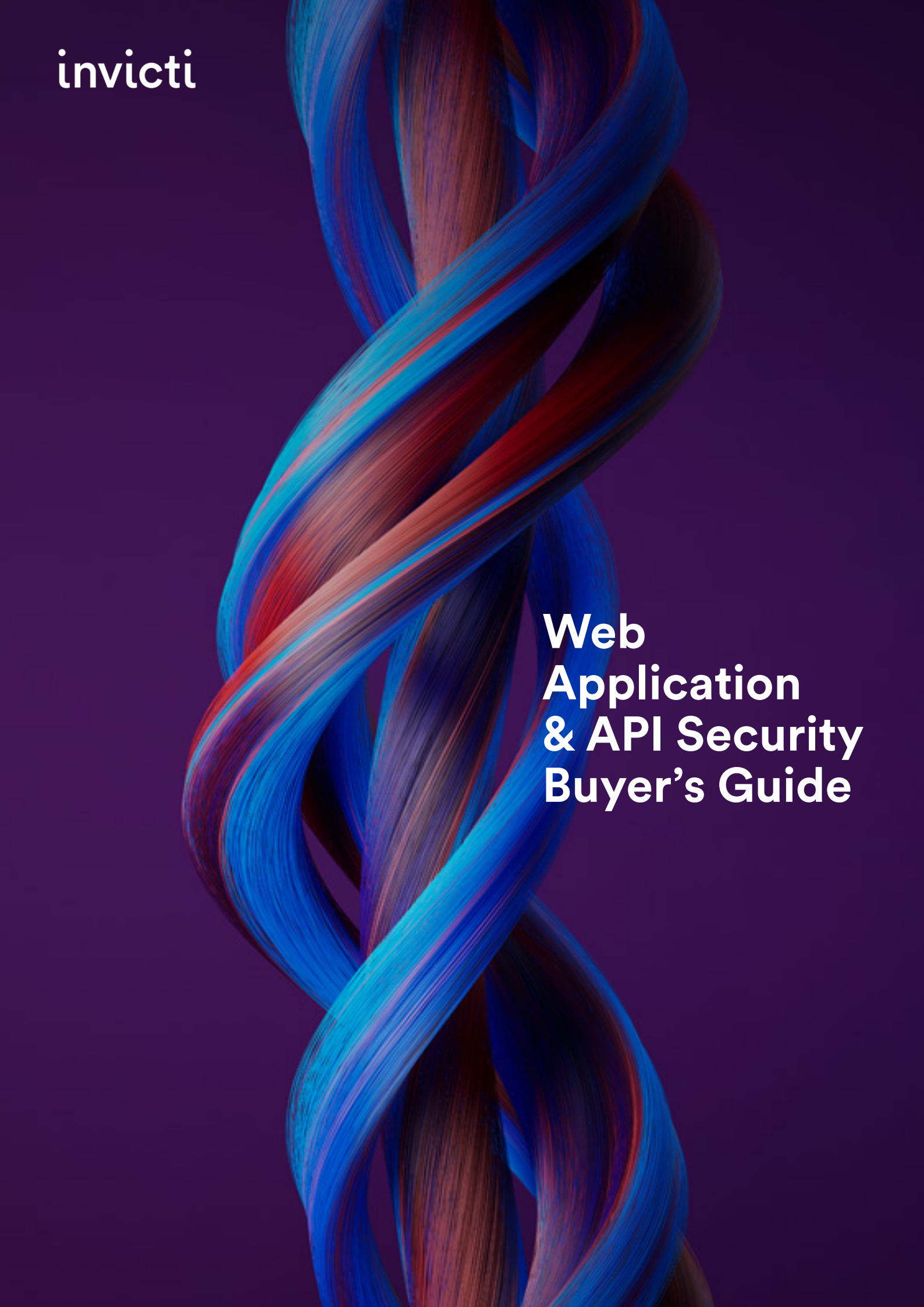
invicti



Web Application & API Security Buyer's Guide

invicti

Who is this guide for?

While Engineering, DevOps, and AppSec Managers are most likely to benefit from this guide, it can be used by anyone involved in selecting solutions to secure websites, web applications, and APIs.

Why do you need it?

We highlighted the features that we believe you should pay attention to when you evaluate solutions. While you likely have selection criteria of your own, please treat this guide as a handy reminder about things that you might want to have on your list.

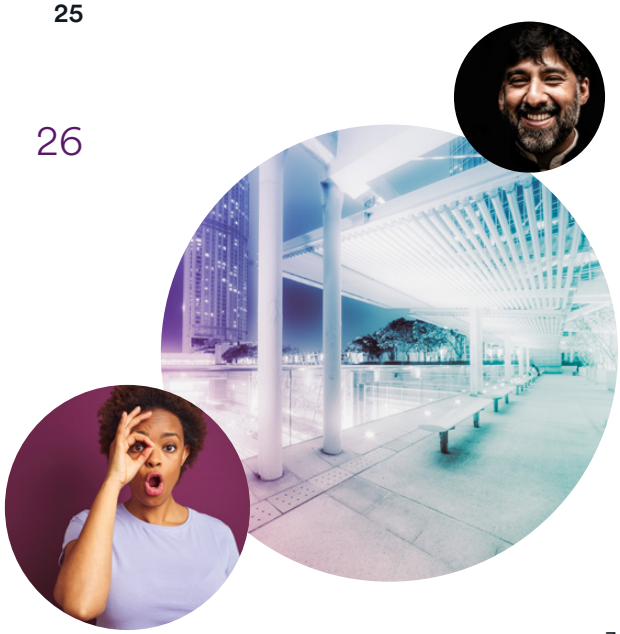
What can you evaluate?

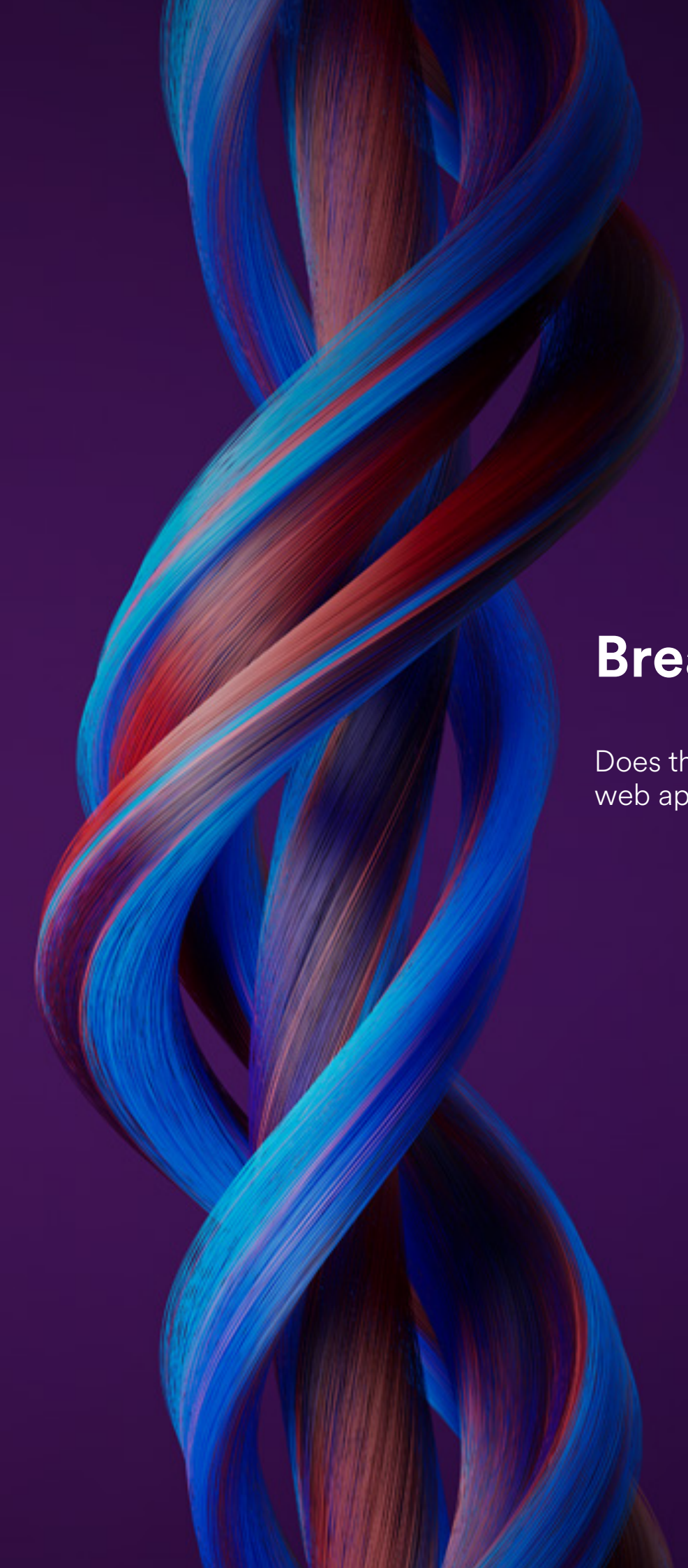
This guide covers primarily dynamic application security testing (DAST) solutions for web applications and APIs. However, if you are looking at web application security testing tools in general, our guide can help.

How and when to use it?

Go through the steps and the checklist presented in this guide to see if your web application security solution evaluation criteria might be missing something. We've also provided tips on finding details that can confirm if the software you are considering meets your criteria.

Table of contents





Breadth

Does the solution cover the entire web application attack surface?

Check for web discovery capabilities

Why you need it

It's likely your organization has websites, web applications and APIs that are not part of your regular security processes yet still belong to you or represent your business. A successful attack on one of these applications may lead to a breach, allow attack escalation, or harm your reputation. For example, even the simplest campaign site associated with your business name may be used in a global phishing attack if it has a reflected XSS vulnerability.

What to look for

Look for a web application security solution that can **find all the websites and web applications that belong to your business**. Technically, this can be as simple as crawling for domain registration information and SSL certificate information. However, the leading tools employ many other techniques to support the discovery process.

How to evaluate

- ☐ **Ask your potential vendors about techniques they are using to discover web assets.**
- ☐ **Request a custom discovery report from the vendor for your business name.**
- ☐ **When you test the software, look for any functionality that lets you automatically find targets to scan instead of adding them manually.**
- ☐ **When you find the discovery function, enter your business name to see what results it provides and how long it takes.**

IMPORTANCE

MEDIUM

Small businesses are less likely to be affected while large organizations should treat this as high importance

RISKS IF REQUIREMENTS ARE NOT MET

- Some sites and applications are left completely unprotected
- Increased risk of a data breach, escalation to other systems, or loss of reputation

Ask about the crawling engine

Why you need it

Your web application security solution must be able to reach as deep into your application as possible. In theory, it should be able to reach every input that exists in the application and test it. However, applications are increasingly complex due to modern JavaScript and HTML5 frameworks. Many crawling engines cannot interpret this complex code correctly and therefore miss many inputs.

What to look for

Look for a web application security solution that uses a **crawler based on a renowned engine** such as Chromium. Crawlers that are developed from scratch have a hard time keeping up with web technology changes. This is because they require a dedicated team that continuously adjusts the crawler to changes in browser engines. On the other hand, Chromium is the engine used by many browsers and, by design, it can interpret all modern web applications.

How to evaluate

- ☐ **Ask potential vendors about the specific crawler that the web security solution uses, and bring skepticism to claims that their crawler supports modern web technologies.**
- ☐ **If you cannot test the solutions, ask each vendor for crawl reports for several intentionally vulnerable applications designed for automatic scanning, for example, Testinvicti. Compare the results to see which crawler found more.**
- ☐ **If you can test the solutions, perform the above tests yourself and compare the crawl results.**

IMPORTANCE

CRITICAL

Eliminate solutions that are not based on renowned crawler engines

RISKS IF REQUIREMENTS ARE NOT MET

- Parts of applications are left unsecured
- Increased risk of attack

Consider crawling from the inside

Why you need it

There are places where even the best DAST crawler can't reach. For example, pieces of the application developed but not yet linked to the main application, directories hidden in the belief in security through obscurity, and more. Such elements should not be left unsecured, especially in the case of hidden assets that might be unlinked but still accessible from the outside.

What to look for

If your environment makes it possible for you to install an additional agent on the server side, look for a **web application solution testing solution that combines DAST and interactive application security testing (IAST) capabilities**. Look for a sensor that can work together with the crawler and combine information from the outside and the inside. This way, you are much more likely to scan the entire potential attack surface.

How to evaluate

- ☐ **Look for vendors who mention DAST and IAST capabilities in their marketing materials. Check if the IAST capabilities are compatible with your environment and application languages.**
- ☐ **If you find information about IAST capabilities, check whether the IAST sensor is active. Most IAST sensors are passive and therefore cannot actively crawl the application from the inside to find unlinked directories and files.**
- ☐ **Ask potential vendors whether their IAST sensor exchanges information with the crawler and the scanner. If the sensor is completely independent, it may generate duplicate warnings that will consume your resources.**
- ☐ **If possible, test the sensor with your specific application to see if it works well in your environment and provides information about unlinked server-side resources.**

IMPORTANCE

MEDIUM

Consider this less important if your developers are trained never to use security through obscurity

RISKS IF REQUIREMENTS ARE NOT MET

- Parts of applications are left unsecured
- Increased risk of attack

Ensure that you can scan your APIs

Why you need it

Web APIs are becoming more and more widespread, from microservices-based applications and mobile application back-ends to IoT devices. In many cases, the only way to evaluate the security of your web solutions will be to focus primarily on the APIs. And while APIs use the same web technologies as websites and web applications, most DAST tools are unable to recognize and test all the endpoints, potentially leaving unacceptable security holes.

IMPORTANCE

CRITICAL

Do not consider solutions that are unable to parse standard API definition formats such as Open API or Swagger

RISKS IF REQUIREMENTS ARE NOT MET

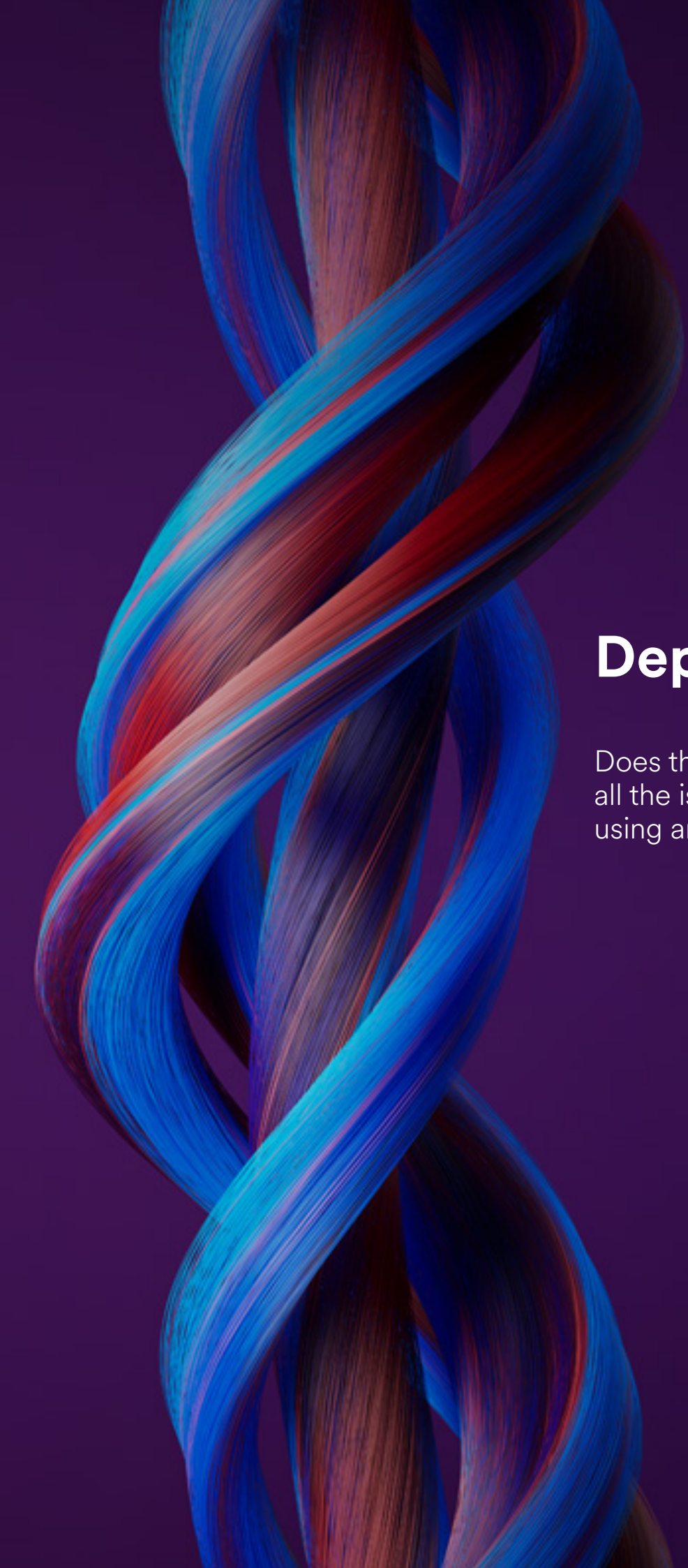
- API endpoints have to be tested manually
- Some API endpoints are left untested
- Increased risk of attack

What to look for

Since a crawler cannot automatically find all the API endpoints (as it does with URLs within a web application or website), your solution must be able to **import API definitions** in the standard that your developers use. Make sure to know the standards that are currently in use in your organization — also consider that these standards might change in the future, and new ones may be introduced. Seek a software vendor that embraces new standards and expands the capabilities of its software.

How to evaluate

- ☐ **Look for publicly available technical documentation to check if the solution supports the API definition standards you use, such as OpenAPI or Swagger. If you have (or plan to develop) GraphQL applications, support for testing GraphQL APIs is also vital.**
- ☐ **If your business uses web services, check if the tool supports importing web service definitions, such as WADL or WSDL files.**
- ☐ **If a Proof of Concept is offered, ask to scan some of your APIs as a test. Otherwise, ask the vendor to provide a sample API scan report.**
- ☐ **Look for or ask about additional import formats to see how versatile the solution is in terms of enabling you to import custom API definitions. For example, see if it supports debugging proxies such as Fiddler or Paros as well as API development platforms such as Postman.**



Depth

Does the solution search for all the issues that may be found using an automated tool?

Look for authentication and business logic support

Why you need it

Many web applications and APIs reach sensitive data and are protected using custom-built authentication measures. Automated scanners cannot enter such areas to test them, leaving them prone to attacks. The same applies to scenarios with multi-page forms where subsequent pages depend on the input on the first page – if automatic scanning tools cannot follow the business logic, they won't get access to parts of the application. Your solution must address these challenges and provide a way to overcome them.

IMPORTANCE

HIGH

Consider this less important only if your web applications have no authentication

RISKS IF REQUIREMENTS ARE NOT MET

- The most sensitive parts of applications are left unsecured
- Increased risk of a data breach

What to look for

The tool you're looking for must have a module or extra functionality that provides a partly-automatic, partly-manual way to **enter custom-authenticated areas** and a second similar module that lets you manually **follow business logic** to reach parts of the application or API that depend on earlier input. These are usually marketed as separate modules using terms such as **authentication and business logic support**. Note that the authentication support module should provide basic automatic discovery that allows for manual intervention if unsuccessful.

How to evaluate

- ☐ **Look through potential vendors' marketing materials to find information about scanning authenticated areas and incorporating business logic.**
- ☐ **Look for an authentication support module that can automatically detect standard authentication patterns and only requires the credentials.**
- ☐ **Look for an advanced authentication support module that lets you manually follow custom authentication flows (macro recorder).**
- ☐ **Look for similar macro recorder functionality that lets you manually follow business logic rules and fully cover, for example, multi-level forms.**
- ☐ **Ask potential vendors for a demonstration (a live presentation) of their authentication and business logic support modules in action.**
- ☐ **If you already have web applications or APIs that contain custom authentication and/or business logic, ask if you can test these modules on your own during a Proof of Concept.**

Evaluate the scope of vulnerabilities found

Why you need it

Certain types of vulnerabilities cannot be detected based on immediate server responses. This includes vulnerabilities that are often called **out-of-band or blind**. In such cases, the attacker sends the payload directly, but the reply is sent to a different place, usually a server controlled by the attacker. Therefore, the web application and API security solution you choose must simulate this type of attack by providing a mock server that receives such replies and proves that the attack is possible.

What to look for

Look for a module that provides support for **out-of-band/blind vulnerabilities**. Such a module is usually a designated server that exchanges information with the scanner. The scanner attempts to send a payload to your web application or API. If there is a vulnerability, the payload bounces back to the auxiliary server, which in turn informs the scanner that the attack was successful.

How to evaluate

- ☐ **Check marketing materials for information about out-of-band/blind vulnerability support.**
- ☐ **Ask about the scope of out-of-band/blind vulnerabilities found by the solution.**
- ☐ **Ask potential vendors to provide a sample scan report with details about detected out-of-band/blind vulnerabilities.**

IMPORTANCE

CRITICAL

Immediately eliminate solutions that cannot find out-of-band/blind vulnerabilities

RISKS IF REQUIREMENTS ARE NOT MET

- Severe out-of-band/blind vulnerabilities are left undetected
- Increased risk of attack

Look for the ability to find more than just web vulnerabilities

Why you need it

When testing the security of web applications, a scanner can find much more than just web vulnerabilities. It can detect web server misconfigurations, discover publicly accessible databases with potentially sensitive data (one of the most common causes of data breaches), and – last but not least – detect insecure third-party modules and libraries.

IMPORTANCE

MEDIUM

While specialized tools do exist for finding some of these issues, a DAST scanner can detect some problems that are undetectable with other methods

RISKS IF REQUIREMENTS ARE NOT MET

- You need to buy, implement, and integrate separate tools to cover these requirements

What to look for

Security solutions get information about different types of issues directly from the scanner and therefore there is usually no specific module or functionality related to this requirement. However, if a potential vendor mentions terms like **misconfigurations, open databases**, and **vulnerable libraries**, there is a good chance that they support the discovery of many different types of web application security issues, not only web vulnerabilities.

How to evaluate

- ☐ **Check marketing materials for mention of misconfigurations and libraries as well as for the term SCA (software composition analysis).**
- ☐ **Ask the vendor for more details on the types of issues related to web application security (other than web vulnerabilities) that its scanner detects.**
- ☐ **When evaluating a scanner, request a scan report that contains information about web server misconfigurations, publicly accessible databases, and vulnerable JavaScript libraries.**

Effectiveness

Does the solution make it easy for security pros and developers to tackle vulnerabilities?

Look for proof of vulnerabilities

Why you need it

If a vulnerability reported by automated software and then automatically assigned to a developer turns out to be a false positive, it has to be reassigned to the security team, manually verified, and then sent back into the queue. This can increase issue resolution time even tenfold, leading to unacceptable delays in development. Every vulnerability that cannot be confirmed with 100% confidence by your software must be verified manually, breaking any development automation and consuming time and security team resources.

IMPORTANCE

CRITICAL

Do not consider solutions that cannot provide confidence and evidence of identified vulnerabilities (depending on the type of vulnerability)

RISKS IF REQUIREMENTS ARE NOT MET

- Every unconfirmed/unproven vulnerability must be confirmed manually
- Major delays in application development

What to look for

The most common terms used to signify that the solution will help you avoid the need for manual confirmation are **confidence and proof**. Some vendors use specialized terms such as **proof of exploit or proof-based scanning** to describe such functionality. Basically, it means that, whenever possible, the scanner performs a mock attack and either provides evidence that the attack was successfully executed or provides a way for you to easily replicate the attack yourself (as proof).

How to evaluate

- ☐ **Ask your vendor for a detailed security scan that includes information about confidence and proof of vulnerabilities, or run a scan yourself using a test version of the evaluated software.**
- ☐ **If the vendor states that it provides proof, look for the following types of proof: contents of a file from your server, information extracted from your database, the result of a command run on your server.**
- ☐ **Note that not all vulnerabilities can be accompanied by evidence from the server but at least remote command execution and directory traversal should be proven.**
- ☐ **If the vendor states that vulnerabilities are provided with 100% confidence, enquire how this confidence is achieved (for selected types of vulnerabilities). If the vendor states that the HTTP request and response are enough to achieve confidence, be wary – tools that only rely on simple pattern recognition in responses are highly prone to false positives.**
- ☐ **If the vendor states that they provide proof of exploit, look for a way to easily replay the mock attack. For example, is there a link that lets you confirm an XSS vulnerability?**

Consider checking for vulnerabilities from the outside and inside

Why you need it

While you may be looking primarily for a DAST solution, additional IAST input can provide several benefits (some of them have already been mentioned when we asked you to consider crawling from the inside). First of all, a small percentage of web vulnerabilities can only be discovered from inside the running application. Secondly, some vulnerabilities can be proven only when you can see the behavior of the application from the inside. And thirdly, having an inside view can make it much easier for developers to fix issues.

IMPORTANCE

MEDIUM

An IAST module gives a clear advantage over scanners that can only see the application from the outside

RISKS IF REQUIREMENTS ARE NOT MET

- Developers waste time manually searching for issues in the source code
- Some vulnerabilities cannot be discovered or proven

What to look for

If your application environment makes it possible to install an additional agent on the server side, look for a **web application and API security solution that combines DAST and IAST capabilities**. Look for a sensor that can align and exchange information with the scanner as well as provide the line of code where a vulnerability is found.

How to evaluate

- ☐ **Look for a mention of IAST capabilities in marketing materials. Check if the IAST capabilities are compatible with your environment and application languages.**
- ☐ **Ask potential vendors whether the IAST sensor exchanges information with the scanner. If the sensor is completely independent, it may generate duplicate warnings that will consume your resources instead of helping.**
- ☐ **If possible, test the sensor with your specific application to see if it works well in your environment and provides information such as the line of code, stack traces, or additional proof of vulnerabilities.**
- ☐ **Look for benchmark information on the overhead introduced by the IAST module – it's commonly acceptable for such a module to introduce a server-side overhead of no more than 5%.**

Make sure vulnerability reports help developers remediate

Why you need it

When dealing with vulnerabilities, fixing an issue takes much more time than finding it. Your chosen solution should strongly focus on shortening the time needed to fix issues by supporting the developers as much as possible. If results from your selected solution are not developer-friendly, they will introduce tensions between teams and major delays in development.

IMPORTANCE

CRITICAL

If the proposed solution does not provide information that helps developers, it will be impossible to truly adopt it in your organization

RISKS IF REQUIREMENTS ARE NOT MET

- Developer frustration, and tensions between development and security teams
- Frequent unsuccessful fixes that require the issue to be re-queued, causing major delays

What to look for

First and foremost, the selected tool must offer **specialist reports for developers** that contain detailed information about the vulnerability. Second of all, vulnerability reports must contain at the very least the exact content of **request and response**, preferably color-coded, with the payload clearly visible so the developer can quickly understand what is happening with the application. If possible (when IAST is used), reports should also pinpoint and quote the **line of code** to help developers quickly find the root cause. Last but not least, the report must contain **links to information** about both the vulnerability and, if possible, **best practices** for fixing that type of vulnerability.

How to evaluate

- ☐ **Make sure that the evaluated solution provides specialist developer reports with complete vulnerability information.**
- ☐ **Check if the evaluated solution shows the content of the HTTP request and HTTP response.**
- ☐ **Check if the HTTP request and response are easy to read, for example, cropped out to contain just the necessary information, color-coded, whether the payload is highlighted, etc.**
- ☐ **If the evaluated solution includes an IAST module, check if vulnerabilities confirmed with IAST show the exact line of code and its content. This enables developers to immediately open the right source file and quickly find the root cause of the vulnerability in their development environment.**
- ☐ **Check several different types of vulnerability reports and see what additional information is provided. The report must contain a short description of the problem, its consequences, its root cause, and the best way to resolve it. It must also contain links to detailed information where the developer can learn about the given type of vulnerability and learn how to resolve the issue.**
- ☐ **If you are not directly involved in software development, don't evaluate this by yourself– involve the developers. Discuss the above information with your developers and ask them if they like the received reports.**

Evaluate user interface and reporting

Why you need it

Your teams will be accessing the user interface and/or the reports of the selected tool regularly. This includes security engineers, developers, management, possibly even executives. While it will be difficult to meet everyone's needs and preferences, they should be taken into consideration. In addition to subjective preferences on the look-and-feel of the tool (both the UI and the reports) you need to make sure that presented information is actually applicable for those who will use it.

What to look for

Your tool should have a **modern** user interface and a **wide** selection of reports, including specialized developer reports as well as trend-based executive reports. If information is presented in an archaic or limited way, it means that you cannot expect dynamic development of the tool in the future. If the vendor pays no attention to current visual trends, user friendliness, and applicability, you may expect that the development of the engine is just as limited.

How to evaluate

- ☐ **Ask future users, for example, security engineers or managers, to evaluate the user interface based on a visual demo or a trial version of the tool. Focus on the opinion of those who will actually be using the tool.**
- ☐ **Ask the vendor to supply you with samples of different types of reports. Ask future report consumers to evaluate these sample reports. Include developers, managers, and possibly executives.**

IMPORTANCE

MEDIUM

While unlikely to cause major problems, subjective perception of the tool is important for its adoption

RISKS IF REQUIREMENTS ARE NOT MET

- Users that do not like the tool may not use it regularly
- If information is unclear/non-applicable, it takes more work to interpret it
- The security team may waste time rebuilding reporting for executives or other stakeholders

Completeness

Can the solution meet every requirement of your web application and API security strategy?

Evaluate usability in the SDLC

Why you need it

To avoid having to replace your web application and API security solution when your development methods change, you need one that is future-proof and fully fit to work in your SDLC, no matter how it is organized. Until quite recently, DAST solutions were perceived as usable only late in the development lifecycle. However, a few current solutions are designed to be just as effective in early development. Since your development organization may change with time, you should select a solution that won't hinder it.

What to look for

First of all, consider how different types of web application and API security solutions **fit in different stages of the SDLC**. If you are not certain whether you should select a SAST, DAST, or IAST solution, check which one can be used both at the earliest stage (i.e. right after the developer makes a commit) and at later stages (in the staging environment or even in production). The primary factors that affect this are scan performance, integration capabilities, report applicability, and a proven track record.

How to evaluate

- ☐ **If you're not set on a DAST solution, carefully consider whether a different type of solution (e.g. passive IAST or SAST) will be flexible enough to use in your SDLC — even if it evolves.**
- ☐ **See whether the solution can be integrated (out of the box or via an API) with your CI/CD solutions (if you use any in your SDLC).**
- ☐ **Ask each potential vendor about its track record of successful deployments in both the early and late stages of the development pipeline. For example, ask for a case study where the solution was used in the earliest stages of development and another showing how it was used in production environments.**

IMPORTANCE

HIGH

If you don't consider a flexible solution that can be used anywhere in the SDLC, changes in your development organization or workflows may require you to replace it later

RISKS IF REQUIREMENTS ARE NOT MET

- Inefficient security testing processes at different stages of the SDLC
- Organizational changes may be impossible due to a lack of software flexibility
- Organizational changes may require a new solution to be selected

Check if you can easily integrate

Why you need it

Standalone security tools consume a lot of human resources and cause major delays. The more integration capabilities a solution has, the more time you will save when setting it up and using it. Even if your organization is still growing, you can expect that in the future the selected solution will have to work together with other products that you buy, both security solutions and development/operations tools.

What to look for

Two things affect your ability to integrate the solution: the availability of **out-of-the-box** integrations and the capabilities of the internal **API**. Both are equally important. Out-of-the-box integrations can save you a lot of time setting up integration with popular products, including issue trackers and collaboration platforms. On the other hand, a robust internal API will let you integrate with any solution, including custom systems and workflows. Look for an API that is well-documented, well-designed, and enables access to all the key functionality.

IMPORTANCE

HIGH

Even if you have few integrations now, it is likely you will need more as your organization grows

RISKS IF REQUIREMENTS ARE NOT MET

- Security silos may develop where a given security tool must be handled separately from others
- You may waste a lot of time and resources both setting up and using an inflexible solution

How to evaluate

- ☐ **Look through the vendor's marketing materials for information on available out-of-the-box integrations. Ask how often new ones are added–current features are just as important as future ones to prove flexibility and growth.**
- ☐ **Look for integrations with issue tracking systems, CI/CD platforms, business communication platforms, etc.**
- ☐ **If you are expecting custom integration work, for example integrating with in-house development or security tools, request access to the internal API documentation. Look through the documentation carefully to evaluate the ease of use and design.**
- ☐ **Ask the vendor for a proven track record of integration support services. Ask them to present cases where they worked together with a customer to develop custom integrations.**

Seek WAF automation for early protection

Why you need it

It often takes weeks or months to fix a vulnerability. The time to fix is the time between the scanner finding a vulnerability and the release of a fixed version of the application. During that time, the application remains defenseless against attacks that target this vulnerability – unless you can somehow temporarily protect it. And that's exactly what web application firewalls (WAF) are for. Your chosen web application security and API solution needs to automatically provide your WAF with the data required to block attacks until the vulnerability is fixed.

IMPORTANCE

CRITICAL

Without WAF rule support, vulnerabilities can remain exploitable for weeks or months

RISKS IF REQUIREMENTS ARE NOT MET

- An attack is almost imminent with severe vulnerabilities out in the open
- Developers may be under extra pressure to quickly fix urgent vulnerabilities, leading to rushed work and internal friction
- Your security team will need to spend time manually creating WAF rules for the most critical vulnerabilities that are found

What to look for

If your proposed solution includes **WAF support**, most likely it can provide suitable data to create rules for your WAF. However, the details may vary greatly. Check if the solution can provide actual **rules**, not just suggestions, and whether it supports the specific web application firewall you are using or planning to buy. You should also check if the solution integrates with your chosen WAF out-of-the-box or via the API. Note that not all WAF products will let you import rules, so these capabilities may be limited not by the DAST solution but by the WAF itself.

How to evaluate

- ☐ **Check if your current/planned WAF supports manual import and API import of rules.**
- ☐ **Look for any evidence of WAF support in marketing materials. See if this includes your current/planned WAF.**
- ☐ **Analyze the documentation for details about WAF support to make sure that you can import rules both manually and automatically (using out-of-the-box integration or via the API).**

Consider your compliance requirements

Why you need it

While web application and API security solution vendors can't directly audit the security of your applications, a DAST tool can be instrumental in achieving and maintaining compliance certification. A vulnerability scanner helps you meet two primary goals: to evaluate what needs to be done before the audit and to provide auditors with proof that you are actively maintaining web application and API security in line with specific compliance requirements. DAST vendors do this by providing built-in reports for various types of compliance.

What to look for

Your selected solution should come with a range of compliance reports, including ones that apply to your line of business (e.g. **PCI DSS, HIPAA, ISO 27001**). Even if you do not need formal compliance, the solution should include industry-standard compliance reports such as OWASP Top 10. Remember that even if your organization does not need to meet compliance standards today, you may need compliance reports in the future. For example, if you seek ISO certification or similar.

How to evaluate

- ☐ **Check the marketing materials and software documentation for the types of compliance reports provided. These should include both reports that you need now and that you might need in the future.**
- ☐ **Ask for an example report or use a trial version of the software to generate a report for one of your web applications. Ask your compliance or legal department to evaluate its viability.**
- ☐ **In the case of industry-specific compliance standards such as PCI DSS, see the potential vendor works with an official auditor (for example, an ASV). If so, you can be sure that the software meets all auditor requirements.**
- ☐ **Ask the vendor for information demonstrating their track record for compliance in your industry.**

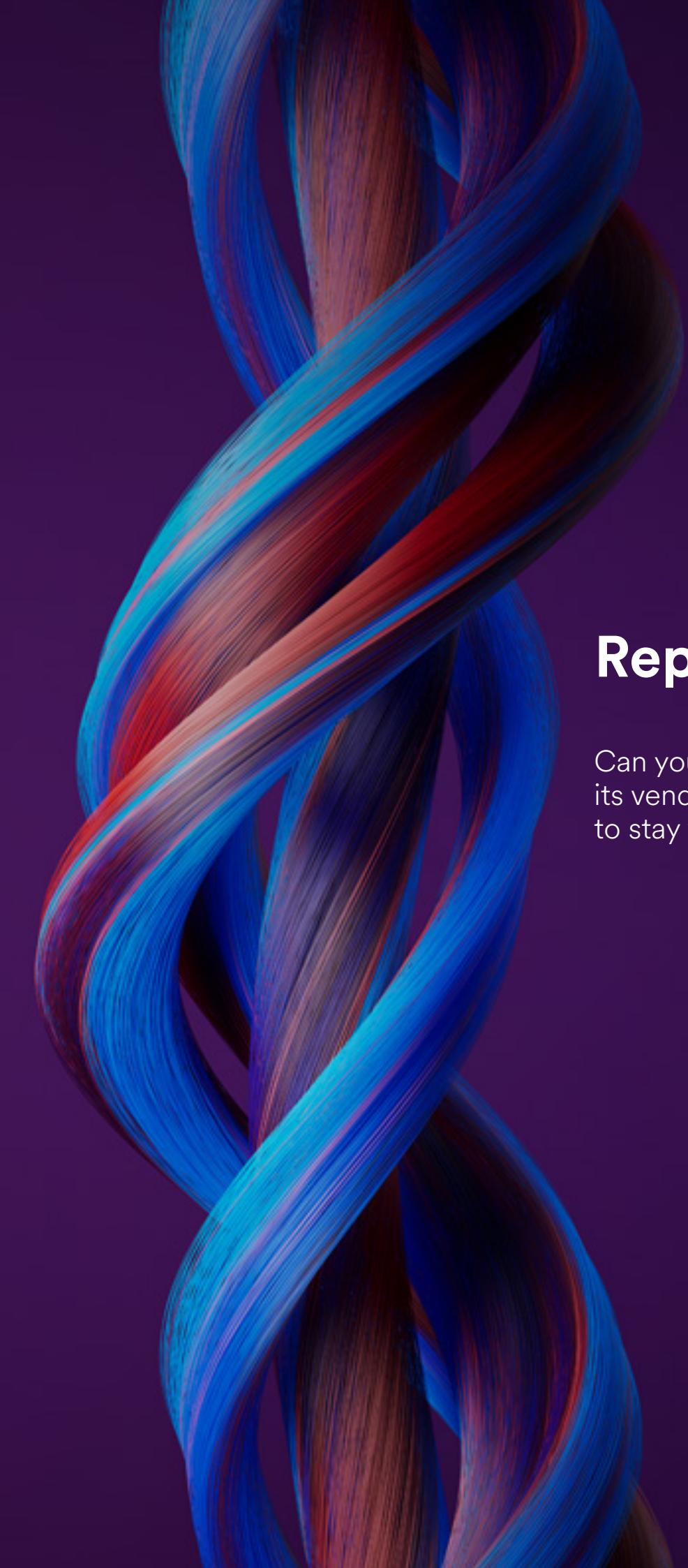
IMPORTANCE

MEDIUM / HIGH

This should be considered high for regulated industries that require formal compliance (such as financial, medical, etc.)

RISKS IF REQUIREMENTS ARE NOT MET

- You may have problems assessing your level of compliance before an audit, which can mean extra work and costs to become compliant
- You will need to manually reorganize data from standard reports from the tool to meet audit requirements
- There might be legal ramifications for compliance violations, including fines or even a custodial sentence



Reputation

Can you trust the solution and its vendor to truly deliver and to stay with you for the long term?

Seek renowned products

Why you need it

vendor to supply your web application and API security solution, you can expect to work together for many years. This is why choosing the right company is just as important as choosing the right product. You want to be sure that the company will not disappear from the market, suddenly change the industry, or discontinue your product, leaving you stranded and forcing you to invest in yet another solution. Market renown should be one of the key factors when selecting a long-term solution.

What to look for

Look for a product that has been on the market for at least **10 to 15 years**. Look for a product that is still supported by the original team that created it. Look for a product that is still in active development and releases new versions often, dynamically expanding its capabilities. This way, you can be sure that the product won't disappear from the market suddenly and that it's not a legacy tool approaching the end of its life. Note that many DAST products on the market do not meet these requirements: they are either complete newcomers with bold claims but no track record or they are old products that are now barely developed or supported.

IMPORTANCE

CRITICAL

Even the best product will be useless to you if its vendor goes out of business in a year and discontinues support

RISKS IF REQUIREMENTS ARE NOT MET

- If the vendor goes out of business, you will have to invest in another solution
- All the time and effort that your teams put into integrating and customizing the solution will have been wasted
- If the vendor has no proven track record, they are likely to be making empty promises –and that means a high risk of project failure

How to evaluate

- ☐ **Look for the history of the product and eliminate products that have appeared on the market during the last few years. A long-term track record guarantees that the product is technologically mature and matches the needs of the industry.**
- ☐ **Look at the release history of the product and eliminate products that do not release new versions at least every month. Some legacy products on the market can't keep up with the fast development of web technologies and won't offer you protection against new threats.**
- ☐ **Inquire about the company that is behind the product, and see whether it is growing and expanding or has remained stagnant. Eliminate products from companies that are not growing, as these are likely to be dying products.**
- ☐ **Check renowned sources such as G2 Crowd to provide insight into customer satisfaction with each vendor's solution.**

Seek best-of-breed software vendors

Why you need it

While for many businesses, especially large ones, it's easier to work with a company that provides a wide range of products that meet many different needs, the quality of those products will never be the same as the quality you can get from a specialized company. Many companies providing web application and API security products come from the network security space and have merely added web application and API security to tick another box in their portfolio, often through a recent acquisition. However, vendors need to take a completely different approach to research and development to provide a practical web application and API security solution. All-in-one products cannot hope to keep up with the innovations introduced by specialized companies. If the DAST tool is not a priority product in the portfolio, there's a risk that it will fall behind in development or eventually be abandoned when the company decides to focus on its core business.

What to look for

When considering potential vendors, focus not just on their current market position but also **their history**. Have they ever focused on web application or API security? Or have they always primarily sold network security solutions and only acquired a web application security company recently? Is the web application and API security solution their primary product or just a minor part of the portfolio? Can the web application and API security solution effectively work standalone, or is it marketed as part of an all-in-one package to get you to buy other products that you might not need at all?

How to evaluate

☐ **Check information about the company, both in the official materials and by actively searching for it on the Web. See how the company began – did they start with web application and API security or, for example, network security? Are they perceived as a web application and API security company or a network security company?**

IMPORTANCE

HIGH

In the DAST field, generic vendors cannot provide products that match specialized ones

RISKS IF REQUIREMENTS ARE NOT MET

- You end up with a product that is far back on the vendor's list of priorities and is not actively developed to keep up with the latest web technologies and trends
- You end up with a product that cannot effectively work standalone and requires you to purchase other products from the same maker
- In the future, the vendor may decide to focus on their primary field of interest and abandon your product, leaving you without an up-to-date DAST solution

☐ **Check information about the product history. How long ago was the product introduced? Was it their first product or one of the most recent ones? Was it acquired from a specialized vendor and made part of a generic portfolio?**

☐ **Ask vendors to provide a strategic product roadmap. See how strongly they prioritize web application and API security. Do they have ideas on what to introduce next? Do they follow current trends? Do they have evidence to support their plans?**

Web application security buyer's checklist

REQUIREMENT & IMPORTANCE	YES	NO
Discovering web applications MEDIUM		
Can the tool discover existing web assets?		
Can the tool discover new web assets as soon as they appear?		
Can the tool automatically treat discovered assets as scan targets?		
Discovering inputs (crawling) CRITICAL		
Can the tool parse SPAs with complex HTML5/JavaScript?		
Can the tool recognize/support frameworks such as React, Angular, and Vue?		
Is the tool crawler based on Chromium or a similarly renowned engine?		
Discovering unlinked assets (IAST) MEDIUM		
Can the tool discover additional files and directories from the server side?		
Can server-side discovery provide information to the crawler?		
Authenticated scanning HIGH		
Can the tool automatically recognize typical authentication schemes?		
Can you record a macro to enter areas that require custom authentication?		
Business logic support HIGH		
Can you record a macro to follow business logic (e.g. multi-level forms)?		
Scanning web services and APIs CRITICAL		
Can the tool import WADL, WSDL, Swagger, OpenApi, and GraphQL definitions?		
Can the tool import data from a proxy such as Fiddler or Paros?		
Can the tool import data from API development platforms like Postman?		

REQUIREMENT & IMPORTANCE	YES	NO
Scanning for web vulnerabilities and more CRITICAL		
Can the tool detect and prove out-of-band/blind vulnerabilities?		
Can the tool detect web server misconfigurations?		
Can the tool detect vulnerable client-side libraries (dynamic SCA)?		
Can the tool detect other types of issues related to web application security?		
Reducing or eliminating false positives CRITICAL		
Can the tool provide direct evidence for vulnerabilities such as SQLi or RCE?		
Can the tool provide proof of exploit for vulnerabilities such as XSS?		
Does the tool provide a confidence rating for issues that can't be proven?		
Confirming and finding vulnerabilities from the server side (IAST) MEDIUM		
Can the tool find vulnerabilities by scanning from the server side?		
Can the tool provide server-side information about vulnerabilities?		
Is the IAST module processing overhead no more than 5%?		
Developer support CRITICAL		
Does the tool provide a dedicated developer report that includes all details?		
Does the tool provide easy-to-read HTTP request and response content?		
Does the tool highlight the payload in the request/response to explain the attack?		
Does the tool provide the file and line of code or a stack trace for the issue (IAST)?		
Does the tool explain the vulnerability?		
Does the tool suggest best practices for fixing vulnerabilities?		
Does the tool provide external links to additional information?		
Do your developers like the reports provided by the tool?		
User interface and reporting MEDIUM		
Do future users of the UI like the look-and-feel and presented information?		
Do future report consumers like the look-and-feel and presented information?		

REQUIREMENT & IMPORTANCE	YES	NO
Flexibility in the SDLC HIGH		
Does the tool have a proven track record of shift-left deployments?		
Does the tool have a proven track record of being used even in production?		
Out-of-the-box integration capabilities HIGH		
Can you integrate out-of-the-box with issue trackers like Jira?		
Can you integrate out-of-the-box with CI/CD platforms like Jenkins?		
Can you integrate out-of-the-box with communication tools like Slack?		
Custom integration capabilities HIGH		
Does the API use an industry standard like OpenAPI?		
Does the API provide access to as many functions as the UI (or more)?		
Is the API documentation easy to read?		
Does the vendor have a track record of custom integrations?		
Can the vendor design a custom integration specifically for your organization?		
WAF automation (temporary protection) CRITICAL		
Can the tool create WAF-compatible rules for discovered vulnerabilities?		
Can you manually export WAF rules to import them into your WAF?		
Can you automate WAF export using the API?		
Compliance support HIGH		
Can the tool generate compliance reports in required formats?		
Does the tool have both generic and specific compliance reports?		
Does the vendor work together with auditors to support compliance?		
Does the vendor have a compliance-related track record?		
Product renown CRITICAL		
Has the product been on the market for at least 10 years?		
Is the product still developed with the help of the original team?		

REQUIREMENT & IMPORTANCE	YES	NO
Product renown (continued) CRITICAL		
Are new functions introduced into the product at least every month?		
Is the solution positively rated in G2 Crowd?		
Best-of-breed specialization HIGH		
Does the vendor specialize exclusively in web application and API security?		
Does the vendor prioritize enhancements to its web application and API security solutions?		
Did the vendor first enter the market with a web application and API security solution?		
Is there a dynamic growth roadmap for the web application and API security solution?		

About Invicti Security

Invicti Security is transforming the way web applications are secured. An AppSec leader for more than 15 years, Invicti enables organizations in every industry to continuously scan and secure all of their web applications and APIs at the speed of innovation. Through industry-leading Asset Discovery, Dynamic Application Security Testing (DAST), Interactive Application Security Testing (IAST), and Software Composition Analysis (SCA), Invicti provides a comprehensive view of an organization's entire web application portfolio and scales to cover thousands, or tens of thousands of applications. Invicti's proprietary Proof-Based Scanning technology is the first to deliver automatic verification of vulnerabilities and proof of exploit with 99.98% accuracy, returning time to development teams for critical projects and innovation. Invicti is headquartered in Austin, Texas, and serves more than 3,500 organizations all over the world.

invicti

FIND US

linkedin.com/company/invicti-security
twitter.com/invictisecurity
facebook.com/invicti-security
invicti.com
invicti.com/contact/