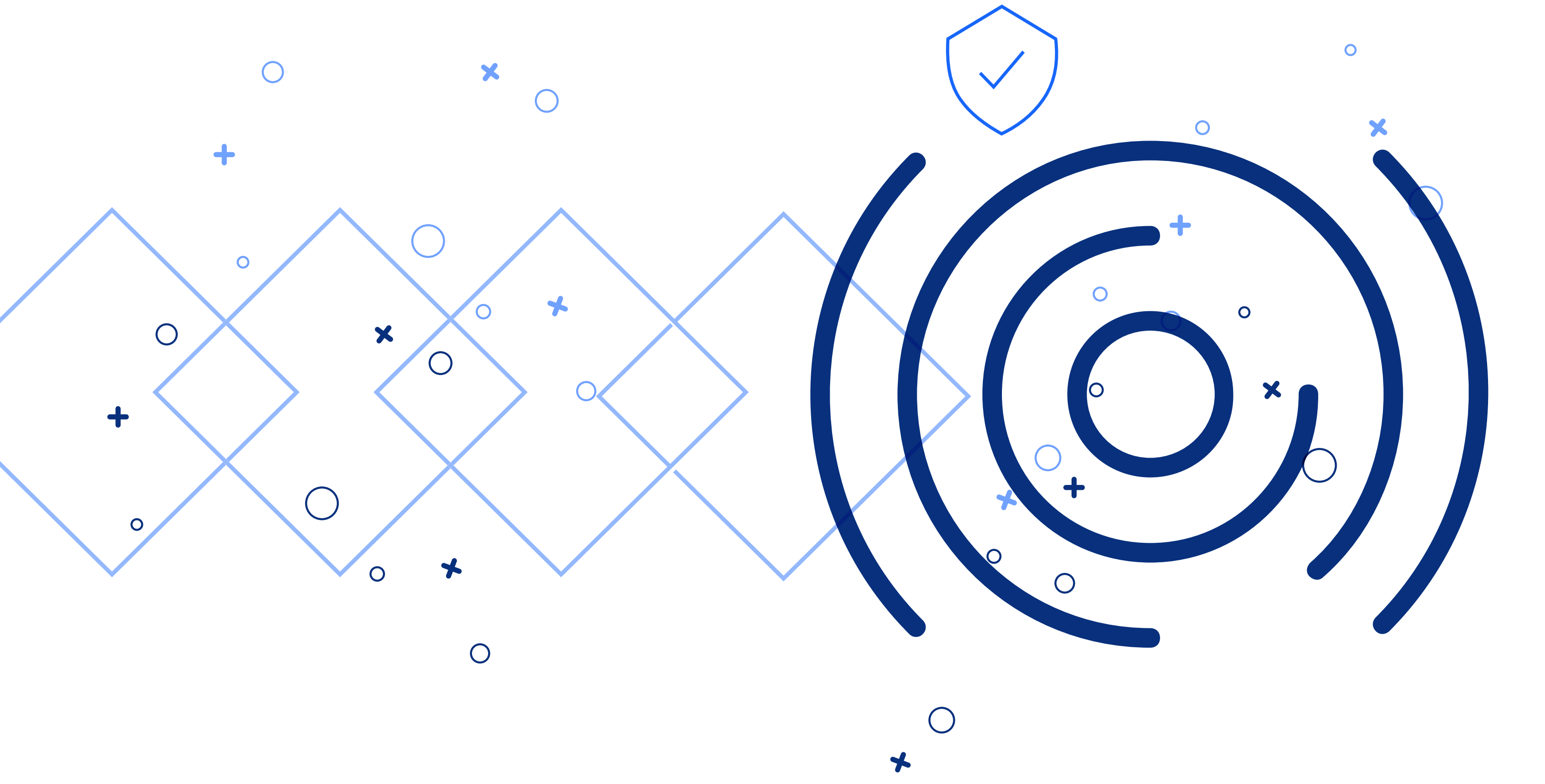
Invicti

WHITE PAPER

How to build a successful AppSec program

Enterprise Web Application Security Best Practices



Executive summary

An effective web application security program needs to cover every corner of your complex and fast-changing application environment and deliver reliable intelligence on your current security posture. At the same time, it has to mesh seamlessly with your development workflows so your organization can maintain security without hampering innovation. And it all needs to work today, tomorrow, and every day in the future – for all your applications.

Until recently, doing all this across a variety of web technologies and application architectures has been extremely challenging technically. Organizations have also struggled to deploy workable solutions in a reasonable time and see measurable improvements to their real-life security posture – but as the security industry matures, things are changing at long last.

This white paper presents the four pillars of a best-practice web application security program and outlines Invicti's tried-and-tested approach to holistic AppSec, including clear and practical steps to:

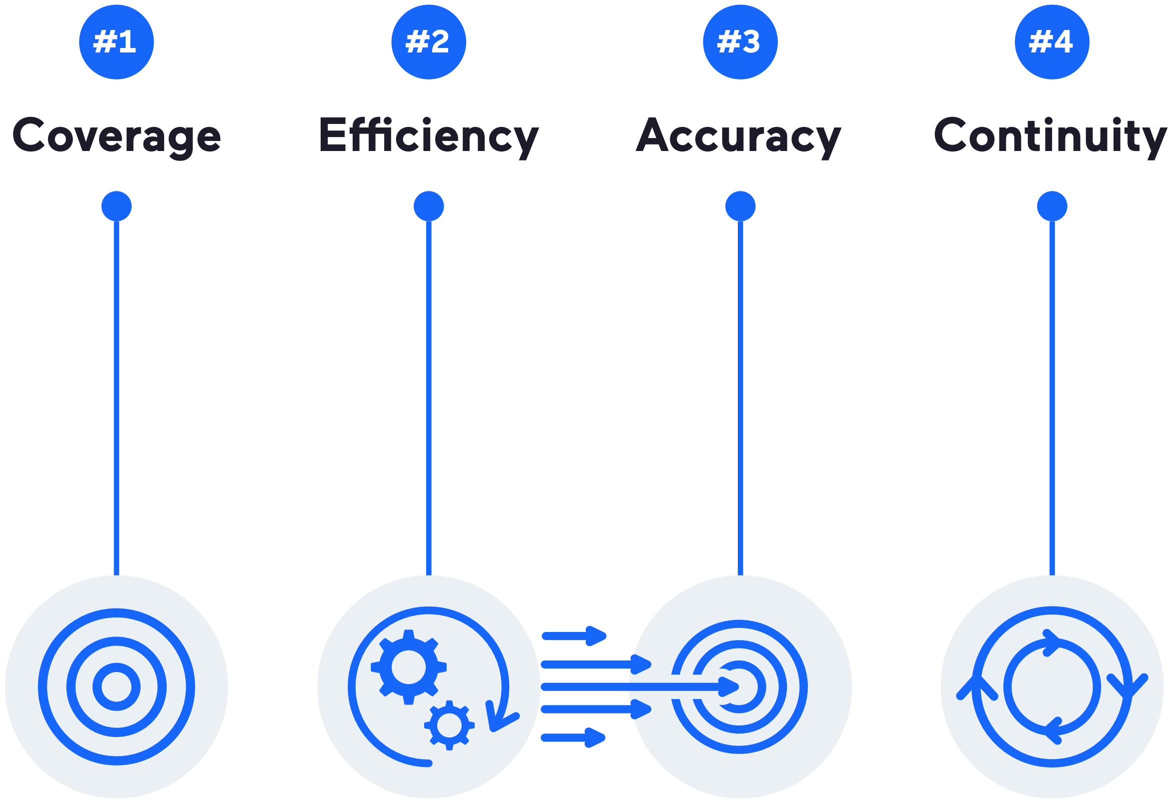
+ Keep track of your true web attack surface
+ Integrate security testing into web application development
+ Detect and permanently remediate web security defects
+ Improve your application security posture in the long run – starting today

Keeping a modern web application environment secure in the face of escalating threats and under relentless pressure to innovate needs a systematic and future-proof approach. Learn how to build an AppSec program that works from day one.

The four pillars of best-practice AppSec

As entire companies move to cloud-based infrastructures and systems while also building their own business-critical software in-house, web application security is no longer about securing a handful of web-facing assets – it's about protecting the entire organization. This puts your web application security program right up there with business continuity plans as a top priority, but with so many moving targets to cover and such high stakes, creating an effective program from scratch is not for the faint of heart.

This chapter outlines the four essential qualities of a best-practice AppSec program and shows how they fit into the reality of modern application environments.



#1 Coverage

#2 Efficiency

#3 Accuracy

#4 Continuity

AppSec pillar #1: Coverage

FIND EVERYTHING, TEST EVERYTHING

It’s a well-worn truth that defenders need to secure everything while attackers only have to find one weakness. Ideally, your application security program should ensure that you know your entire attack surface and can be confident that you’ve left no known gaps for the bad guys. While this might seem a straightforward (if ambitious) goal, you need to be very clear about defining your attack surface and what being secure means for your specific application environment.

In the pre-cloud days, cybersecurity was mostly about network security and building a secure perimeter to wall off internal systems and applications from the dangers of external network traffic. This was security understood first and foremost as blocking unauthorized access (and thus attacks), and it worked well – provided you could tightly control all possible routes of Internet access. Today, when nearly everything resides in the cloud, and most applications are themselves made up of dozens or hundreds of web services, there is no perimeter that you could realistically secure.

When you assume that every application and component could be accessed from anywhere in the world by malicious actors, the weight of security shifts very definitely from building fences to making sure all your doors and windows are locked – and that means testing everything.

To test every web asset in your organization, you need to know about it. Ideally, all your websites and applications should be listed in a central inventory for easy commissioning, maintenance, and decommissioning. While this is definitely a best practice, most organizations still have only a vague idea of their true web attack surface. This makes ongoing asset discovery and management a vital part of any AppSec program.

As for the testing itself, there is a vast array of approaches to application security testing out there, from manual penetration tests to various types and levels of automated scanning. Each method has its benefits and tradeoffs. To give just a handful of examples, manual penetration testing is the most accurate but slowest, static analysis is easy to add but cannot cover all application components or vulnerability types, and vulnerability scanning provides the broadest coverage but without the ability to pinpoint issues in the code. An effective AppSec program should thus incorporate a carefully considered mix of testing methods to provide maximum coverage across all assets and components – but without causing a performance bottleneck.

AppSec pillar #2: Efficiency

TEST AND REMEDIATE AT THE SPEED OF DEVELOPMENT

The days of having only a handful of websites that you could update and secure at your leisure are long gone. Any sizable organization is now a software company that develops some or all of its business applications in-house, so the web development process is deeply woven into the fabric of business operations and growth. At the same time, modern web frameworks and methodologies have made it possible for smaller teams to build complex applications and deploy new functionality faster than ever. Whenever this rapid development hits a speed bump, the entire organization is affected – so waiting for security is no longer an option.

Development teams are under intense pressure to innovate and deliver on time, often working in short, agile sprints. When you have two weeks to design, build, and test a new feature, waiting for security testing to catch up is not an option.

To be efficient, your AppSec program has to be deeply ingrained into the software development lifecycle (SDLC) and provide your teams with the tools and processes they need to release secure applications on schedule. Application security needs to be a routine aspect of everyday development and testing work. For developers, this means deep integration with their existing issue trackers so they can view and resolve application security issues like any other software bug. For security engineers, it means being able to focus on finding and investigating vulnerabilities while minimizing the overhead of avoidable manual tasks.

Automation is the cornerstone of agile development, so efficient AppSec has to be automated as well – but simply automating existing tasks is no guarantee of efficiency. In fact, automating the wrong things can merely result in moving an existing bottleneck further downstream, where someone else will have to sift through it manually. More than anything, efficient AppSec automation needs accuracy.

AppSec pillar #3: Accuracy

HAVE CONFIDENCE IN YOUR DATA

When you're running frequent security tests on hundreds of web assets, testing is only the beginning – you still need to act on the results. Getting accurate data to the right people and systems at the right time is another fundamental requirement for any effective and scalable AppSec program. After all, even if you are maximizing coverage and automating testing as much as possible, security defects don't fix themselves. To go from detection to remediation without wasting time and effort along the way, you need to plan for work and data flows that mesh seamlessly with existing dev processes while also cutting out noise.

Your AppSec program needs to scale rapidly to always keep pace with application development. This requires efficient automation combined with unconditional accuracy – because you don't have the time or resources to automate false results.

Not that long ago, web security testing could be either fast or accurate, but to keep up with the pace of innovation, you now need both. Combined with the requirements for SDLC integration and automation, this necessarily makes accurate automated application security testing the foundation of your AppSec program. While manual testing will always have its place, keeping up with automated dev toolchains and CI/CD (continuous integration/continuous deployment) pipelines requires integrated tools that can run automatically and quickly deliver feedback to developers. Crucially for workflow efficiency, the data they provide must be reliable and actionable so developers can remediate security issues without breaking their fine-tuned routines – and without the distraction of false alarms.

So a successful AppSec program needs to ensure accuracy on two levels: first in generating your security testing results and then in getting them to the right people for remediation. Once you can do this reliably and efficiently, the final piece of the puzzle is making the whole process continuous.

AppSec pillar #4: Continuity

KEEP GOING NO MATTER WHAT

Even if you are confident that you have full security testing coverage of your entire web application environment (and that's already a big if), each test run only gives you a snapshot of your current security posture. With applications being created and modified on a daily basis and then launching straight into a highly dynamic global threat environment, today's test results may well be out of date tomorrow, never mind next month. To maintain coverage across relentless and unpredictable changes, your AppSec program needs to ensure that testing and remediation can keep up no matter what.

Cybercriminals won't wait until you've completed your periodic penetration test. To minimize the risk of exploitable issues making it into production, you need to efficiently test and retest application security at every stage of development and operations.

A lot of enterprise web development is done in a continuous model, with new and updated functionality being deployed in small chunks and frequently – up to several times a day in some companies. While security is always a process rather than a one-time effort, continuous application deployments require equally continuous application security. After all, with modern web frameworks, a whole new application could be set up and deployed in a matter of hours, increasing your attack surface overnight. Unless it is quickly and completely covered by your AppSec umbrella, your organization could be vulnerable to attack until the next testing window – so whatever changes in your environment, you need the tools and processes in your program to catch it and test it.

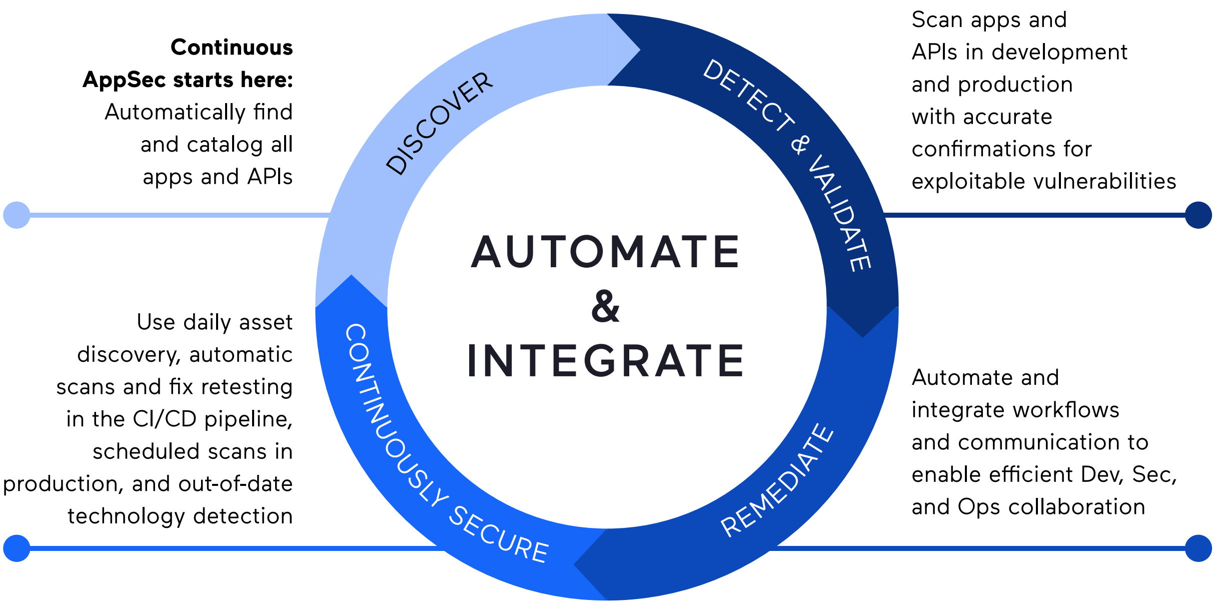
As cybercriminals add new tricks to their attack playbooks and the security community works day and night to uncover, report, and patch software vulnerabilities, a continuous application security program must also evolve continuously to at least keep pace – or, better still, stay one step ahead. In the next chapter, you will learn how to do this with Invicti by building an AppSec process that sits firmly on all four pillars to combine coverage, efficiency, accuracy, and continuity – and works in the real world from day one.

Building an enterprise web application security program with Invicti

To go from wishful thinking to everyday execution, you need to flesh out your AppSec program with the right tools and workflows for your organization. At first glance, there is no shortage of security solutions and certainly no shortage of acronyms, yet companies are struggling to find the product mix that works for them and brings measurable security improvements. The Invicti approach to web AppSec provides one solution to this puzzle – not the only possible solution, but definitely one that has already helped thousands of organizations cover all four pillars of best-practice security in their real-world environments.

This chapter shows how to rapidly implement a comprehensive web application security program based on provably accurate dynamic testing with Invicti Enterprise.

Invicti's Continuous Application Security Solution



Continuous AppSec starts here: Automatically find and catalog all apps and APIs
Use daily asset discovery, automatic scans and fix retesting in the CI/CD pipeline, scheduled scans in production, and out-of-date technology detection
DISCOVER
DETECT & VALIDATE
REMEDIATE
CONTINUOUSLY SECURE
AUTOMATE & INTEGRATE
Scan apps and APIs in development and production with accurate confirmations for exploitable vulnerabilities
Automate and integrate workflows and communication to enable efficient Dev, Sec, and Ops collaboration

DAST-driven AppSec that works

The Invicti approach is unique because it advocates fully automated dynamic application security testing (DAST) as the foundation of your entire AppSec program – and with good reason. First and foremost, dynamic testing is the only way to quickly extend security coverage to any web application at any stage of development and operations, regardless of the underlying technologies and architectures. It also provides a realistic picture of your security posture by identifying and reporting directly exploitable vulnerabilities and is fast enough to scan and rescan as often as you need without delaying releases. And, crucially, you get actionable results from day one, so you can immediately get to improving your application security – and that starts with mapping out your real-world attack surface.

STEP 1

DISCOVER WHAT YOU NEED TO SECURE

Building your security coverage starts from the moment you first log in to Invicti Enterprise using your company email address. Based on the company domain from your email and any additional domains you specify manually, Invicti's proprietary asset discovery service queries a dedicated database of Internet assets and returns likely matches for your domains in a matter of seconds. This database is highly optimized and continuously updated, which means you'll receive quick notifications whenever a new site appears in your organization – and this all starts before even running your first vulnerability scan.

Once you've gone through the discovery results and selected the sites and applications you want to test, you can organize and group them in a way that makes sense for your specific organization. For example, you may want to define website groups that correspond to specific geographies, business units, or development teams. You can also create custom tags for websites and website groups for an additional dimension of visibility, perhaps to tag your business-critical applications for quick access and easy filtering. Based on grouping and tagging, you can then run tests and generate reports for specific subsets of your web asset inventory.

Know your APIs, test your APIs

Apart from user-facing websites and applications, every modern enterprise operates a vast set of web services and APIs. To make sure this crucial attack surface doesn't slip under the radar, Invicti Enterprise lets you import API definition data in multiple industry-standard formats and automatically adds any API definition files found during crawling. API endpoints are then tested for vulnerabilities using security checks that cover REST, SOAP, and GraphQL APIs. Get the Invicti white paper to learn more: Web API Security: Defending Your Hidden Attack Surface

At a more granular level, discovery is also about knowing all the potential attack points for every site and application in your environment. When you run a scan, Invicti Enterprise goes through a multi-stage process of crawling and probing your applications to create a list of all the URLs that can be tested for vulnerabilities, including all externally accessible files and links. When the interactive application security testing (IAST) agent is enabled, it provides additional server-side intelligence about files that would normally be inaccessible to the external scanner. The crawling phase also includes dynamic technology detection to identify runtimes, frameworks, databases, libraries, and web servers to optimize testing and immediately flag outdated versions.

At this stage, you already know what sites and applications you need to test, and the scanner has detailed information about their attack surface – time to run the security checks.

STEP 2

CHECK FOR VULNERABILITIES

Invicti Enterprise comes with several thousand high-quality security checks – automated tests that safely probe identified attack points for vulnerabilities. To control which checks you want to run, you use scan profiles. You can start with the built-in profiles and then define any number of custom scan profiles to customize the subset of security checks you need, optimize performance, and tweak scan settings to precisely match your technical requirements. Among other things, scan settings are where you define authentication behavior to ensure that restricted site areas are also scanned. It is also where you can enable IAST and software composition analysis (SCA) functionality by installing a server-side agent that communicates with the core scanning engine during testing.

Unlike some basic vulnerability scanners, Invicti uses a full embedded browser engine to load test targets and observe the effects of its security checks. To ensure accuracy, proprietary Proof-Based Scanning technology is used to automatically confirm the vast majority of directly exploitable vulnerabilities, with a false positive rate of less than 0.02%. With the IAST agent enabled on your server, you get additional insights into how your application responds to security checks, which translates into more vulnerabilities uncovered and more detailed information about them – including SCA checks to find vulnerable components. Even as the scan runs, Invicti continues to expand your test coverage with advanced features such as heuristic URL rewriting, which is also used to infer and test additional API endpoints based on known URLs.

Proof-Based Scanning works by automatically and safely exploiting direct-impact vulnerabilities and delivering evidence that an attack is possible. Real-world usage data has shown that these automatic confirmations are accurate for *99.98%* of cases.

Turning raw scan data into actionable intelligence is where reporting comes in. For each scan in Invicti Enterprise, you get multiple report levels, from a quick status overview to a complete technical report with full scan details. You can also generate a variety of compliance reports and track vulnerability trends across your entire organization or only specific websites or website groups. And to quickly and reliably get vulnerability reports to your developers, you can use out-of-the-box issue tracker integrations to automatically create tickets for confirmed vulnerabilities.

Having demonstrably accurate security testing results means you can eliminate manual verification and go from detection to remediation in a matter of seconds.

STEP 3 REMEDIATE WITH SDLC INTEGRATION

Before developers can start fixing security defects, you need to be confident that you are making the best use of their time. With manual testing and most vulnerability scanners, this used to mean having your security team sifting through test results   to weed out false positives, assign severities, and select issues for remediation. With Invicti's Proof-Based Scanning and accurate automatic severity assignment, you can confidently send your developers real and exploitable issues with little or no manual intervention. That way, your security professionals can focus on tasks that truly need their expertise and intuition, like prioritizing issues by business risk or investigating business logic vulnerabilities.

To automate the ticketing and remediation process, you can use built-in integrations with industry-standard issue trackers, collaboration platforms, and CI/CD tools, such as Jira, Slack, and Jenkins, to name just a few. If you need to tweak out-of-the-box behavior or integrate with a custom solution, Invicti Enterprise also comes with an extensive API that exposes all the necessary functionality for automation. When setting up integration, you have the flexibility to create any number of users and user groups within Invicti, assign fine-grained role-based permissions, and map the website and user structures to your internal teams and staff in a way that makes sense for your organization. This makes it possible, for example, to assign vulnerabilities found in a specific site directly to the developer responsible for that site, or to give team leaders visibility into vulnerability trends only across the specific website groups they are responsible for.

Invicti's IAST component runs on the server and attaches to the application runtime to monitor reactions to DAST security checks and provide an extra layer of insight as well as an SCA capability.

To fix a security defect, developers need to know what vulnerability was found, where it was found, and how to fix it. Vulnerability reports that developers get from Invicti Enterprise provide reliable and detailed information about each security weakness and its potential impact, including the location of the identified vulnerability – and if the IAST component is used, that can mean down to the specific line of code. Invicti also provides remediation guidance in each vulnerability report, helping developers to understand the issue and fully address its root cause. This is crucial to prevent superficial fixes that are only implemented to pass testing. To double-check each vulnerability that is marked as fixed, Invicti will automatically run an incremental scan to test if the fix is effective.

By automating the whole cycle of testing, remediation, and retesting, you can move towards making your AppSec process truly continuous.

STEP 4 MAKE SECURITY CONTINUOUS

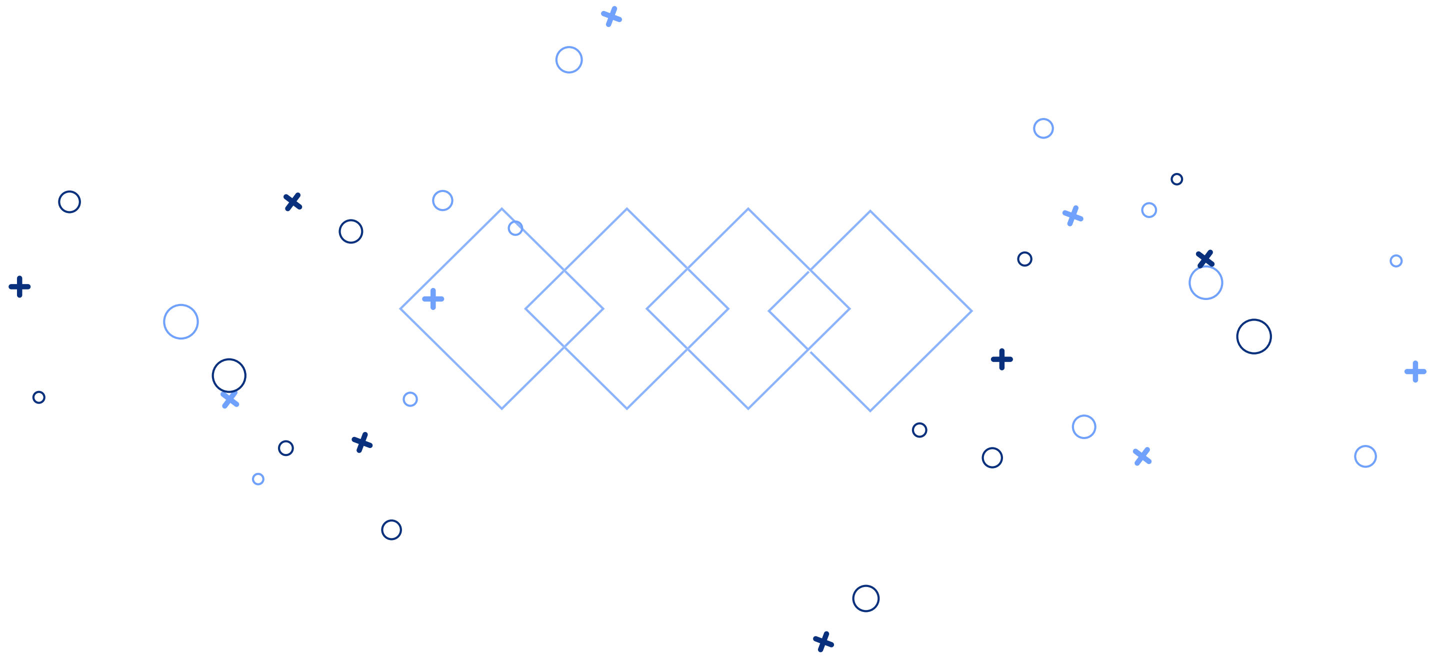
DAST-first application security testing with Invicti has another crucial advantage: you can run it as often as you need at different stages of the development and operations cycle. By hooking into the SDLC with two-way integrations, you can trigger incremental scans on commit, assign identified security issues back to the same developer, and retest the submitted fix – all without involving the security team. You even have the option of failing builds that contain vulnerabilities above a certain criticality threshold to ensure that security issues are eliminated as early as possible in the development process.

Vulnerability scanning has always been the main type of security testing performed during QA in staging environments (and sometimes the only place DAST was used), so naturally, Invicti also covers this phase of the SDLC. Running scans on complete pre-production environments allows you to catch runtime vulnerabilities and misconfigurations that might not show up earlier and remediate them before they can make it into the live application.

Depending on your process, you can schedule scans, launch them manually, or trigger them automatically at specific places in the workflow.

Once changes are committed to production, you need to make doubly sure that the live app you are exposing to the entire world has no known security holes. Invicti makes it possible to scan your production applications as often as you need – which could even be daily if that's what your continuous deployment process requires. Scanning a live production environment always carries some risk, especially with less mature scanners, but Invicti brings over a decade of experience to help you scan in production safely. It is recommended practice to clone your production environment and scan the latest clone rather than the live app. If you need to scan a live environment, you can gradually customize and optimize a baseline scan profile to ensure you are maximizing coverage without interfering with the application. Either way, keep scanning at every stage of the application lifecycle with regularly updated security checks to minimize your risk in a dynamic threat landscape.

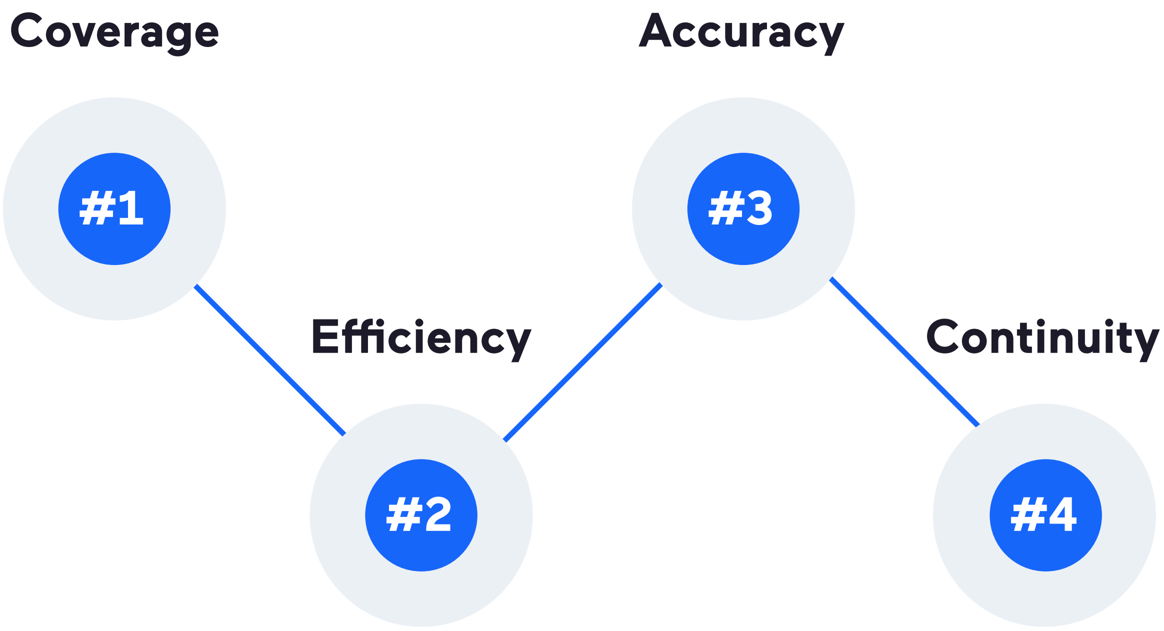
When application development runs in a continuous cycle, continuous security testing is the only way to stay safe while keeping pace with innovation.



Best-practice AppSec for the real world

Taking the four pillars of AppSec from a wishlist to a process that works in the real world for your specific organization can be extremely complex – or surprisingly simple. With countless moving pieces in your application environment and ever new security threats outside it, the only realistic way to cover your entire attack surface is with accurate and up-to-date dynamic security testing. Precisely because good DAST is a must-have, Invicti has made it the foundation of a holistic AppSec process that infuses reliable vulnerability scanning into your entire SDLC.

There are many ways to build out an application security program that covers all four pillars, but most of them need a lot of time to deploy and fine-tune, and during that time, you are not getting value from them. Beyond ensuring coverage, efficiency, accuracy, and continuity, the Invicti approach also has the key advantage of quick deployment to bring rapid value in the form of measurable security improvements. At Invicti, we are firmly convinced that this is where AppSec is going – and when it gets there, we'll be waiting.



Invicti

Invicti Security is changing the way web applications are secured. An AppSec leader for 15 years, Invicti delivers DAST, IAST, and SCA technologies that empower organizations in every industry to continuously scan and secure all of their web applications and APIs with a highly integrated, automated approach spanning the entire software development lifecycle. Invicti is headquartered in Austin, Texas, and serves more than 3,000 organizations of all sizes all over the world.

FIND US

in linkedin.com/company/invicti-security

twitter.com/invictisecurity

f facebook.com/invicti-security