

HADRIAN

Why DevSecOps is the best bet for secure development

And why some companies
still fail to use it right



Why DevSecOps is the best bet for secure development

And why some companies still fail to use it right



Executive Summary

In today's rapidly evolving development environment, traditional security measures often fall short, struggling to keep pace with the ever-expanding attack surfaces of modern organizations. Security should be a collective responsibility that extends beyond dedicated security teams. Embracing DevSecOps means integrating security throughout our development processes.

Hadrian's approach emphasizes that security isn't just a one-time task; it's an ongoing commitment. Prioritizing continuous monitoring that adapts to the evolving threat landscape over point-in-time testing.

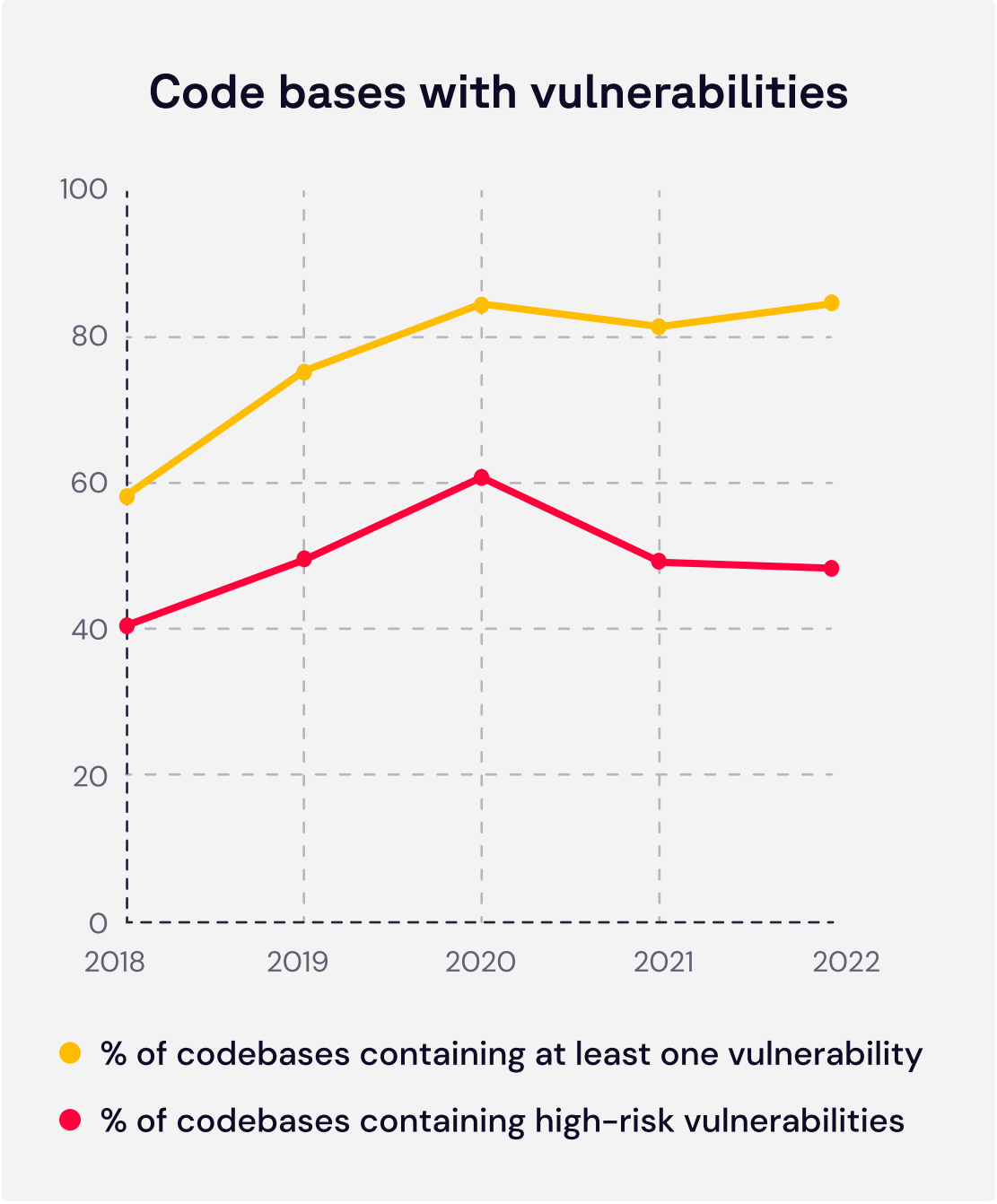
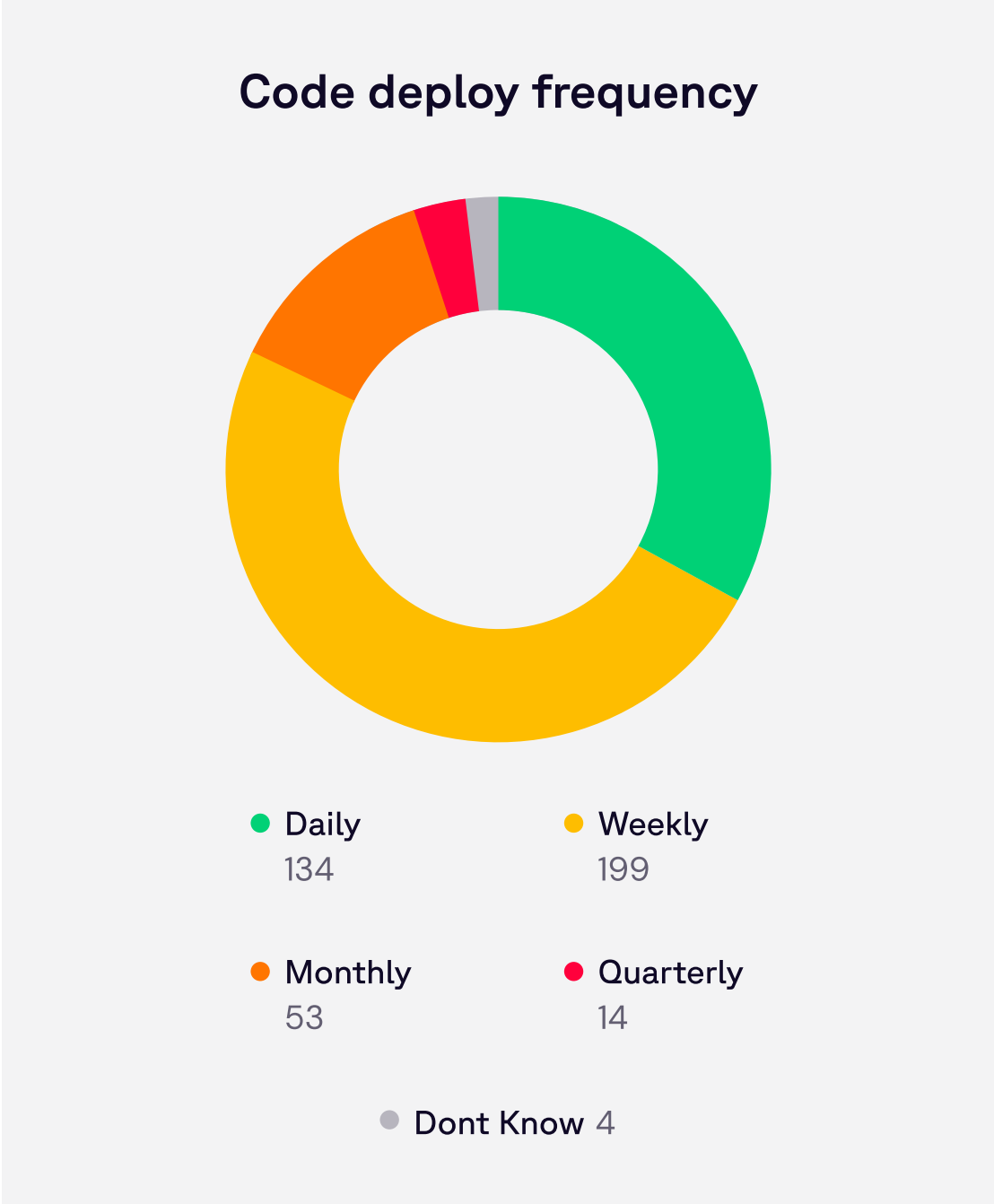


Prash Somaiya
CTO, Hadrian

Table of Contents

2. Executive Summary	8. Conventional software development vs. DevSecOps
3. It all started with DevOps	9. Why traditional penetration tests won't work with today's software development life cycle
4. Speed isn't enough, The benefits of DevSecOps	12. How to create an efficient and effective DevSecOps, Pillars for Success in DevSecOps
5. DevSecOps takes commitment, DevSecOps employs both shift-left and shift-right strategies	13. How Hadrian can help you fulfill the 6 Pillars of Success
6. The gaps: Why some companies still don't have a mature DevSecOps	16. In Summary, Bibliography

Technology in today’s world is all about innovation, but with it comes growing pains. Nowhere is that more evident than with software development. The world is calling for more applications and feature updates daily. This poses a number of challenges, with cybersecurity at the top of the list.



It all started with DevOps

Before there was DevSecOps (Development, Security and Operations), there was DevOps. DevOps--the combination of software development (Dev) and IT operations (Ops)--has done a lot to speed up software delivery to meet the growing customer demand.

Putting these two normally distinct teams together, Dev and Ops, accelerates product delivery by embracing continuous integration and continuous delivery (CI/CD) practices.

“DevOps is all about meeting software users’ ever-increasing demand for frequent, innovative new features and uninterrupted performance and availability,” **IBM** says.¹

Sixty percent of developers using DevOps say they are releasing code twice as quickly. “But increased speed creates a tradeoff: Nearly half of organizations consciously deploy vulnerable code because of time pressure,” according to a recent GitLab survey.²

Speed isn't enough

When it comes to development, speed isn't enough. Software also needs to be secure--now, more than ever. According to Cybersecurity Ventures, the global annual cost of cybercrime is predicted to reach \$8 trillion in 2023, with the rising cost of damages resulting from cybercrime expected to reach \$10.5 trillion by 2025.³

The need for more security efforts in DevOps produced the next wave of development, DevSecOps. Arriving relevantly recently on the scene, it embeds security at every phase of the software development lifecycle, without slowing down development.

The DevSecOps motto is "software, safer, sooner." And developers quickly put their hopes in such a strategy. **Gartner** describes DevSecOps as a good way to seamlessly and transparently integrate security into DevOps, without reducing the agility or speed of developers or requiring them to leave their development toolchain environment.⁴

The benefits of DevSecOps

Some say that DevSecOps is what DevOps was intended to be from the start. The benefits of DevSecOps are quite clear. It gives development and IT teams the security and efficiency they are looking for, along with a more efficient DevOps practice, easier automation, cost and time savings, and better collaboration.⁵

DevSecOps is gaining traction. Thirty-six percent of respondents in GitLab's recent developer survey say they already develop software using DevSecOps, compared to 27% in 2020.⁶

As security breaches continue to rise, organizations will seek ways to make the security phase of development faster and easier. The combination of AI, cloud technologies, and automation makes DevSecOps more beneficial than ever.⁷

Perhaps that's why researchers estimate the DevSecOps market to reach 41.6 billion by 2030, up from 3.7 billion in 2021.⁸

\$10.5 trillion

The cost that cybercrime is expected to reach in 2025.

36%

Of developers surveyed said they are using DevSecOps

DevSecOps takes commitment

A recent **IBM** data breach report shows that the most significant factor to reducing the cost of a breach is having a mature DevSecOps approach.⁹

During 2020’s COVID pandemic, DevSecOps became more popular, giving organizations high hopes of achieving the speed and security they were looking for. But now disillusionment is setting in.

It’s not easy to prove the value of DevSecOps as an emerging strategy. “Because, in truth, DevSecOps entails a significant cultural and process shift,” and a great deal of commitment and investment to succeed.¹⁰

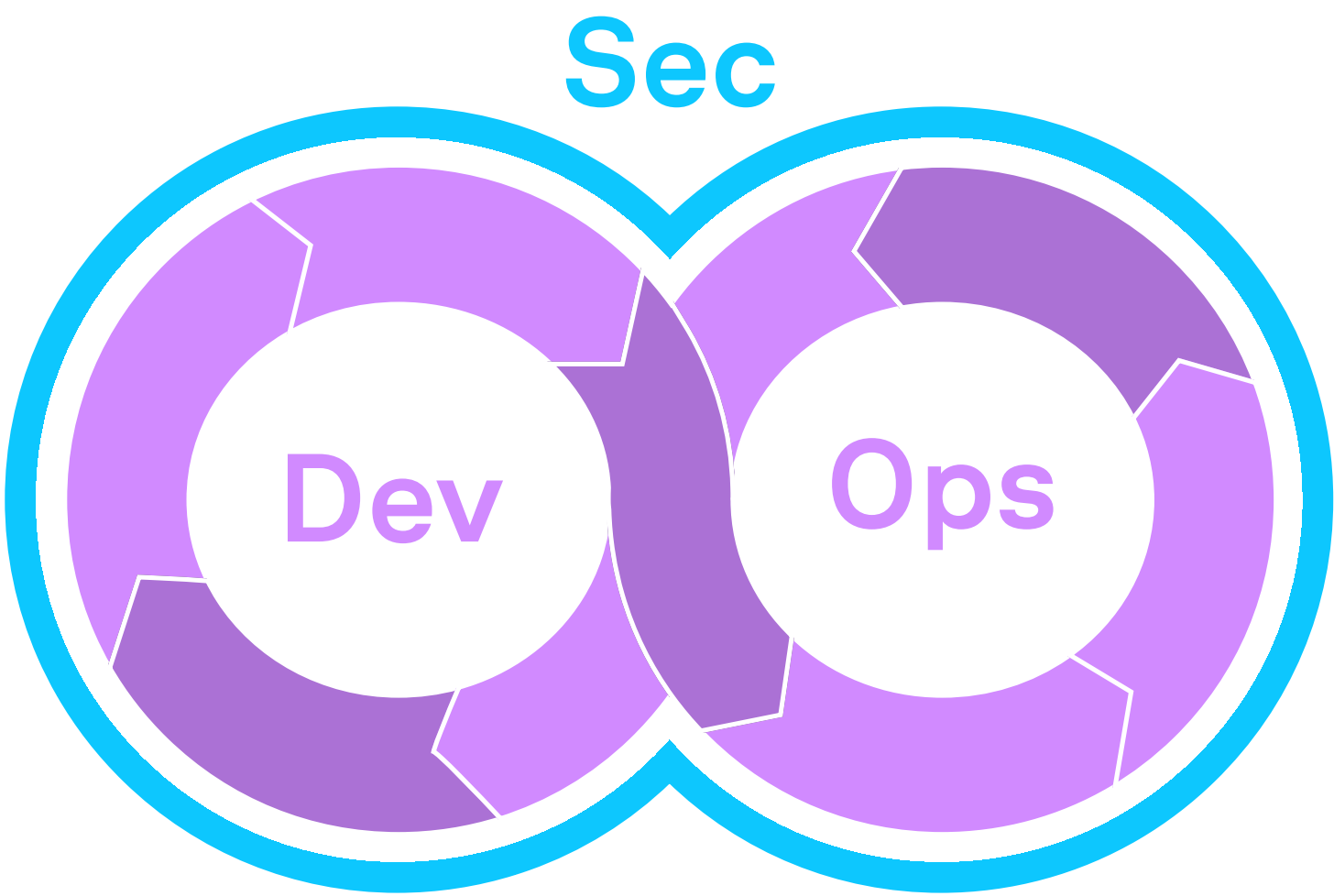
Establishing a mature DevSecOps practice can realize massive speed and security benefits for businesses but integrating security into the workflow is challenging. Automating the detection of vulnerabilities and weaknesses simplifies DevSecOps workflows, embedding security into the process and making the software more secure.

DevSecOps employs both shift-left and shift-right strategies

If development is looked at on a continuum from start to finish, DevOps is a shift-right strategy. DevOps looks at security after (to the right of) an application’s development and deployment. This is a problem because vulnerabilities might escape detection before customers use the new software.

DevSecOps is better because it uses both shift-right and shift-left strategies. It uses shift left to protect software as it is being developed--to the left of the product being released. It shifts right post-deployment, with manual penetration testing.

Even in development environments, some risks aren't readily identified. Post deployment testing is significant for finding risks, because that is when the software is live. Penetration testing is the last element of manual testing--and that is when it matters most.



The gaps: Why some companies still don't have a mature DevSecOps

DevSecOps brings together development, security, and operations in hopes of creating a seamless and secure software delivery pipeline, but it hasn't been a given for many companies. It's not as easy as some developers might have hoped at first.

To be effective, DevSecOps requires effective automated penetration testing. And many still struggle to meet that objective. Here are more reasons why some companies struggle to succeed with DevSecOps:

Failure to have good continuous integration and continuous delivery (CI/CD)

Automation is critical for continuous deployment of code. CI/CD automates workflows, streamlining tasks for DevSecOps, freeing them up to do what they do best—coding. Continuous integration (CI) automatically builds, tests and integrates code changes, and continuous delivery (CD) automatically delivers those. Then continuous deployment (CD) automatically deploys code changes to customers. CI/CD contributes to performance and speed.

It only works if vulnerabilities are detected

In conventional software development, security testing is a separate process from the software development life cycle (SDLC). The security team discovers security flaws only after they build the software. The DevSecOps framework improves the SDLC by detecting vulnerabilities throughout the software development and delivery process. But this only works if those vulnerabilities are detected.

DevSecOps requires something faster than traditional pentesting.

"Waterfall" engineering, where code pushes are larger and less frequent, can work well with traditional pentests. In such an environment, where products see a maximum of 2–3 updates per year, waiting a few weeks for test results before each patch release doesn't cause problems. But with DevSecOps, that doesn't work. Pentesting requires 4–6 weeks to scope out and start the pentest, then an additional 2–3 weeks to run the test, collect the findings, and deliver results in a report. This is too time consuming to support a rapid development process like DevSecOps.

Gaps in automation make efficiency impossible

DevSecOps teams build a more efficient process, while breaking down silos between development, IT operations and security, and by increasing collaboration and the use of security tools and procedures. All of this needs to be supported by automation. Without it, DevSecOps is dead in the water.

Culture lacks collaboration

Convincing teams to embrace a collaborative mindset is crucial. Developers, security experts, and operations personnel must work together, breaking down silos for DevSecOps to succeed.

Failure to Integrate security practices into the DevOps workflow

This is a complex process. It requires security checks at every stage—from code development to deployment. And it requires automated security testing as part of the CI/CD pipeline.

Improper balance between rapid delivery and security checks

It's hard to balance speed and security when it comes to software development. Mature DevSecOps teams have found a way to work with automated security testing to ensure it doesn't hinder development.

Still using legacy applications

Some organizations still have legacy applications with outdated security practices. This doesn't allow for the automatic security updates needed for DevSecOps.

Need automated compliance checks

Compliance and auditing can slow down the speed and agility of development. That's why compliance checks need to be automated.

The right tools aren't seamlessly integrated

A mature team doesn't get overwhelmed by the vast number of vendor choices and their products. They carefully choose the right tools for their specific DevSecOps team and ensure those tools work well together. Then, they seamlessly integrate them into their process.

Need a balance of shift-left and shift-right mentalities

Companies that fail to encourage security practices early in the development lifecycle will struggle. Using the shift-left approach of integrating security into requirements, design, and coding phases will ensure success of DevSecOps. On the same token, shift right is equally important for testing once the software goes live. Some risks aren't detectable until then.



"Cybersecurity is not a separate stage in the SDLC, nor does it only show up in a single stage. Cybersecurity is a constant thread that weaves through each stage of the SDLC. A 'security by design' approach fundamentally shapes how the SDLC carries out its processes, significantly enhancing the security and robustness of the final product."¹¹

CrowdStrike

Conventional software development vs. DevSecOps

In conventional software development methods, security testing is a separate process from the software development life cycle. The security team discovers security flaws through pentesting, months after the software goes live.

This is a problem because:

- Engineers lose interest in fixing the problems if too much time goes by
- Fixing them might require undoing lots of code

Conventional software development and DevSecOps are two different approaches to software development. Conventional software development is a traditional approach that focuses on developing software in a linear fashion, with each stage of development being completed before moving on to the next. This approach is often criticized for being slow and inflexible, as it can be difficult to make changes once a stage of development has been completed.

While conventional software development can be effective for certain types of projects, DevSecOps is becoming increasingly popular due to its emphasis on security and collaboration. By integrating security into the development process, DevSecOps helps to ensure that software is secure and compliant from the moment it is launched, rather than leaving it vulnerable to attack. Additionally, by emphasizing collaboration between teams, DevSecOps helps to ensure that software is developed efficiently and effectively.

On the other hand, DevSecOps is a more modern approach that emphasizes collaboration between development, security, and operations teams throughout the entire software development life cycle. DevSecOps aims to integrate security into the development process from the very beginning, rather than treating it as an afterthought. This approach helps to ensure that security is built into the software from the ground up, rather than being bolted on at the end.



“DevOps practices have led to 60% of developers releasing code twice as quickly. Nearly 25% of respondents reported having full test automation, a 13% increase from 2021.”¹³

Global Market Insights



“Ninety-six percent of respondents in a recent study believe their organization would benefit from automating security and compliance processes, a core principle of DevSecOps.”¹²

StrongDM



By integrating security at all stages from strategy to operations, you can improve quality by moving tasks to the left early in the life cycle. This helps bring the security mindset to the “codefront.”¹⁴

IBM

In summary, while conventional software development and DevSecOps are two different approaches to software development, DevSecOps is becoming increasingly popular due to its emphasis on security and collaboration throughout the entire software development life cycle.

Efficient software development is paramount to the success of many businesses, particularly in light of the surging adoption of Software as a Service (SaaS). Across various industries, businesses heavily depend on software and applications to meet their business objectives and to deliver products to customers. That’s why many organizations now embrace DevSecOps to ensure streamlined and secure code creation and maintenance.

Why traditional penetration tests won't work with today's software development life cycle

Many application flaws get past security checks. To succeed, organizations need resources to ensure the security of their software development life cycle (SDLC), and especially for DevSecOps teams with their shift-left mentality.

Many application flaws get past security checks. To succeed, organizations need resources to ensure the security of their software development life cycle (SDLC), and especially for DevSecOps teams with their shift-left mentality.

The SDLC works by following a plan that removes the typical pitfalls of software development projects. It starts by evaluating existing systems for deficiencies. Next, it defines the requirements of the new system and creates the software through six stages. These are: analysis, planning, design, development, testing, and deployment.¹⁵

DevSecOps is involved in all six of the software stages, and especially on testing through attack surface management (ASM). Vendors usually conduct passive or active vulnerability assessments to confirm a risk's presence and its exploitability. Autonomous penetration testing enhances ASM with deeper risk analysis and emulates human adversaries, providing a more comprehensive grasp of potential threats.

The problem with penetration testing is that it takes a month or so to scope, then it needs to run tests for two or three weeks. The report is usually 100 pages long. And that doesn't work when you're doing things on a daily cycle. It doesn't include regression testing, to make sure that things are fixed. You need to identify issues sooner and improve the supply chain's logical organization. Costs can be exorbitant if run regularly or at a large scale.

Many organizations struggle to conduct rigorous penetration testing regularly and instead rely on annual assessments. This gap creates an opportunity for cybercriminals to find and exploit vulnerabilities.

Regular penetration testing can be intrusive and disruptive for an organization's daily operations. Testing often has to be conducted outside of business hours to prevent any impact on business operations.

Confirming that security risks have been successfully remediated using manual penetration testing methods is time-consuming. Organizations can spend hours just validating their work.



Existing tools such as security rating services and vulnerability scanners (even ASM) lack fidelity of risks to help with security testing. Some comprehensive tools do exist, but they can be difficult to configure and use for fast-paced DevSecOps workflows, which have non-security personnel.

Pen testing is often used for smaller scale testing. These tests focus on a subset of an organization's technologies, such as identifying security weaknesses in a network, machine, or piece of software.

But pen testing doesn't factor in the wider enterprise because it is designed to test the integrity of an individual system. It fails to consider how systems and assets work together to create vulnerabilities. This short-sighted view can easily lead to a breach.

In addition, it's not suited for larger organizations. Pen testing is less scalable, too resource-intensive and more expensive for large organizations with complex networks.

In the rush to develop new applications, security considerations are often overlooked or put aside for the sake of speed. But where development is taking place—in cloud computing, IoT devices, and mobile applications--has led to an expanded attack surface for threat actors to exploit.



"More often than not, security teams tend to provide purely negative feedback (e.g., 'Your product has these 12 vulnerabilities') and toss that information over the proverbial fence, back to development, which usually creates not only an adversarial relationship but also creates delays that don't always need to exist.

Working as a partnership, on the other hand, means working with the developer to come up with a suitable solution in many cases that addresses the underlying security issue rather than the outward symptoms of the issue.

For example, take a simple case of a service vulnerable to Indirect Object Reference issues. A quick answer for the developer is to implement a system where each object has an identifier based on the hash of the actual ID.

This "fix" will get the security people off the developer's back but doesn't actually fix the issue since an attacker might very well notice that the ID values are all hashes and make them curious to poke and see if it's just an obfuscation on top of an easily enumerated set of ID values."

Jeremy Banker, Staff Security Software Engineer - VMWare



“There is the perception that organizations focus their security efforts around their traditional “known” environments. This creates an incentive for threat actors as they are more likely to find an exposed asset on the internet that they can exploit.”

Olivier Beg, Head of Hacking at Hadrian

Penetration Testing as a Service (PTaaS) is another choice for testing, and it can be a game-changer. PTaaS combines automation for basic tasks performed by penetration testers with human assessment. It allows for more frequent monitoring and testing, starting with an initial assessment and adapting scope based on changes and new vulnerabilities in services.

PTaaS offers immediate actionable insights, unlike traditional reports which may not provide ongoing visibility. However, this testing can be expensive due to its dependency on humans.

Red teaming simulates real-world attackers but even if red teaming is put to good use, it's still antiquated. It is largely manual, hard to scale and prohibitively expensive for frequent use. For some organizations, it is too expensive to do at all. Today's growing threat landscape calls for a more proactive strategy.

Continuous Automated Red Teaming (CART) is vastly better than an annual pentest. It is also focused on a goal. That goal is to hack your company — giving it a much broader scope than penetration testing.

Unlike traditional red teaming, CART never quits in its mission. And it uses artificial intelligence (AI) and machine learning to grow even smarter at hacking you. Just like a real hacker.

CART attacks any vulnerability it can find, making it much more like a real world attack. It exposes a company's security weaknesses the same way a hacker would — with no warning and no predictability.

CART also:

- Operates with little human intervention
- Is less likely to err than humans
- Is consistently updated to test for the latest threats
- Focuses on an organization's entire network, including systems and applications
- Bridges the gap between annual penetration tests and red/blue team testing
- Considers how assets link together and the multi-stage attacks a malicious hacker might perform

Pen-testing can't do any of this with the speed and accuracy needed for DevSecOps. Automated testing backed up by AI is what it's going to take. Edge users and developers are the first line of defense.

How to create an efficient and effective DevSecOps

With the recent digital explosion in development, it’s no wonder that software is a prime target for cyberattacks and software is at the top of business risks today, according to the **Cloud Security Alliance**.¹⁶

The rapid pace of development makes it hard to keep up with security. This is where DevSecOps comes in and why it’s so important.¹⁷

DevSecOps reduces the complexity of development and publication of software and ensures that only known and trusted components and services are used. It empowers software development teams with security resources and uses development environments that are well secured and monitored. The goal is to deliver the end-product with only the features it was designed to have.¹⁸

6 Pillars for Success in DevSecOps

By focusing on these pillars, you can more readily address weaknesses in the development of your software and give your DevSecOps team the best chance of success, says the Cloud Security Alliance.

Collective responsibility

Get your whole development team and your entire organization on the same page with security. Everyone has a critical part to play. It cannot be an afterthought or siloed. And security needs to be measured.

Collaboration and integration

Skill and talent are at a shortage in DevSecOps today. Success can’t be achieved under these conditions without a collaborative culture. Members of all functional teams need to pull together.

Pragmatic implementation

Compliance requirements help to identify appropriate controls. For a successful DevSecOps, translate these controls into software measures that can be automated and measured.

Bridging compliance and development

By following the process of shifting left, software teams can prevent undetected security issues when they build their applications.

Automation

To make multiple revisions in a day, security scanning tools need to be incorporated to prevent security evaluations from slowing down development.

Measure, monitor, report and action

It’s important to monitor metrics. Some of the most critical ones for DevSecOps are deployment frequency, vulnerability patch time, percentage code automatically tested, and automated tests per application.

How Hadrian can help you fulfill the 6 Pillars of Success



Collective responsibility

Use these solutions to build your team effort.



Hadrian can help with:

Transparent risk scoring: With Hadrian’s transparent risk scoring, you can engage teams dissociated from security with transparent scoring that makes the impact of an issue apparent and how it affects the overall posture of your organization.

Asset tagging. Our centralized asset management allows you to assign assets to users and helps teams take ownership of their assets. Hadrian makes it easy to allocate specific risks to an individual or team. Our asset tagging feature lets you add detailed information about the asset's owner. By tagging assets with the owner's information beforehand, you streamline the process of assigning remediation tasks, saving valuable time. Hadrian’s asset structure is scalable and flexible, enabling us to add new asset types easily.



Collaboration and integration

We have a number of ways to get your DevSecOps team on the same page.



Hadrian can help with:

Autonomous validation: All threats found in Hadrian’s Verified Risks have been autonomously validated and teams can double check this themselves with step-by-step reproduction instructions. Hadrian’s proprietary severity scoring automatically assigns a level using a combination of metrics including our built-in threat intelligence. When security teams receive an alert, they are provided all the information they need to assign it for remediation.

Secure sharing: Hadrian’s innovation, Secure Share, simplifies the mobilization process enabling faster remediation with less effort. The new feature provides a seamless workflow for security teams to share all of the risk details with the relevant development team for remediation and track resolution progress. At a high level, Secure Share enables non-Hadrian users to access specific risk detail pages. The feature provides teams with the information they need, eliminating the need for back-and-forth communication, and speeding up remediation.

Role-based access: Gives users visibility of the assets they are responsible for and features necessary to resolve security issues.



Pragmatic implementation

We can help you meet compliance requirements.



Hadrian can help with:

Asset inventory: Our asset inventory helps you understand everything in your environment and how they are connected. Within the inventory is information about your technologies, software versions, certificate information, and other important configuration information

Continuous security: At Hadrian, our continuous security uses an event-driven architecture, which limits intrusion. Our contextualization of a potential vulnerability allows for only relevant tests to be run, and only when there is a change in the system. We don't perform unnecessary tests that create "noise." There is a need to be thoughtful about how tests are run. Too much testing at high speeds can overwhelm and degrade company IT infrastructure.

Instant testing when changes occur: Hadrian's platform performs continuous attack surface discovery, whenever a change is detected (such as new code being pushed) new penetration tests are run. Assets are correlated by machine learning algorithms that review text, images, scripts, DNS records and numerous other characteristics.



Bridging compliance and development

We can help you with undetected security issues.



Hadrian can help with:

Speed: Hadrian's Orchestrator AI autonomously identifies risks on a continuous basis. We conduct tests the moment something happens, providing instant feedback. Hadrian's in-house hacker team continuously updates and adds to the "hacker modules" used by Orchestrator AI so that our platform can detect new exploits within 24 hours of discovery.

Regression testing: We uncover new defects in existing functionality after changes have been made to software to ensure that the new code has not affected other parts of the software. We make sure the previous functionality of the application is still working and bug-free after code changes.



Automation

Our scanning tools prevent security evaluations from slowing down development.



Hadrian can help with:

Automated modular testing: At Hadrian we believe we have struck the balance between continuity and lack of intrusion. Through modular testing and speed in event scheduling we collect continuous threat intelligence without being a burden on your system.

Continuous security validation: Our platform consistently checks to make sure your security protocols work as intended. Benefits for you include more effective security budgeting; optimization of your detection and response tool stack; visibility to your IT infrastructure and more effective protection of sensitive client information.

AI-powered automation: Hadrian leverages event-based AI to automate the penetration testing process. Testing is automatically triggered whenever the event-based probes detect changes to the attack surface. This automated approach mimics a hacker's process, providing continuous and cost-effective testing.

Scalability: Our platform finds and tests everything external. Everything is in scope, so nothing is missed.



Measure, monitor, report and action

We will help you measure and report critical DevSecOps metrics.



Hadrian can help with:

Automated reporting: We help monitor your organization's overall health/security posture to provide consistent and accurate insights. These are provided through on-demand reports that are easily exported and shared with stakeholders, and highly consumable. Reporting automation streamlines the reporting process, eliminating manual data analysis. Hadrian's reporting enables security and compliance teams to focus on implementing changes instead of building reports. Data is contextualized and correlated by Hadrian's AI to create accurate and detailed insights about the assets and risks present within the organization.

Generative AI assistant: Hadrian's new Generative AI assistant, Vibia AI, expands your security team by serving as a dedicated offensive security expert. It gives your team 24x7 access to a penetration tester to validate risks, probe threats in your attack surface, and facilitate remediation. Built into the Hadrian interface, users can chat with Vibia AI as if they were messaging a colleague through an instant messaging platform.

Security posture measurements: Hadrian provides an overview of the organization's security posture, helping your team track its progress and continuously improve security.

In summary

By establishing a mature DevSecOps practice, you can realize massive speed and security benefits for your business. But integrating security into the workflow is challenging. Automating the detection of vulnerabilities and weaknesses simplifies DevSecOps, by embedding security into the process and making the software more secure.

At Hadrian, we enhance cybersecurity with automated penetration testing, emulating hackers for comprehensive security validation using event-based AI.

Hadrian’s platform continuously carries out risk assessments, validating their results and alerting you when remediation is required. The only area where Hadrian cannot help DevSecOps is with scanning code or testing applications in pre-production environments. We test live applications.

Many organizations rely on annual security assessments, but this is too slow for DevSecOps. Cybercriminals will find and exploit vulnerabilities during the development process. Hadrian detects vulnerabilities and enables DevSecOps teams to proactively remediate risks as they are discovered.

Bibliography

[1] <https://www.ibm.com/topics/devops>

[2] <https://www.globenewswire.com/news-release/2021/05/04/2222642/0/en/GitLab-s-Fifth-Annual-Global-DevSecOps-Survey-Reveals-2020-Was-Catalyst-for-DevOps-Tool-Adoption.html>

[3] <https://www.esentire.com/resources/library/2022-official-cybercrime-report>

[4] <https://www.gartner.com/en/information-technology/glossary/devsecops>

[5] <https://about.gitlab.com/developer-survey/>

[6] <https://about.gitlab.com/developer-survey/>

[7] <https://www.strongdm.com/blog/devsecops-statistics>

[8] <https://www.verifiedmarketresearch.com/product/devsecops-market/>

[9] <https://thehackernews.com/2023/12/cost-of-data-breach-report-2023.html>

[10] <https://www.forbes.com/sites/forbestechcouncil/2021/08/31/the-pros-and-cons-of-building-a-devsecops-culture/?sh=743417df2359>

[11]<https://www.crowdstrike.com/cybersecurity-101/secops/software-development-lifecycle-sdlc/>

[12] <https://www.strongdm.com/blog/devsecops-statistics>

[13] <https://www.gminsights.com/industry-analysis/devsecops-market>

[14] <https://developer.ibm.com/articles/devsecops-what-and-why/>

[15] <https://stackify.com/what-is-sdlc/>

[16]<https://cloudsecurityalliance.org/research/topics/devsecops/>

[17] <https://cloudsecurityalliance.org/research/topics/devsecops/>

[18] <https://cloudsecurityalliance.org/research/topics/devsecops/>

HADRIAN

Hackers are upping their game — shouldn't you?

Find out more about Hadrian’s 24/7 cybersecurity platform that uses a proactive, automated, and adaptive approach to identify and address security vulnerabilities.

Get a demo

Learn more