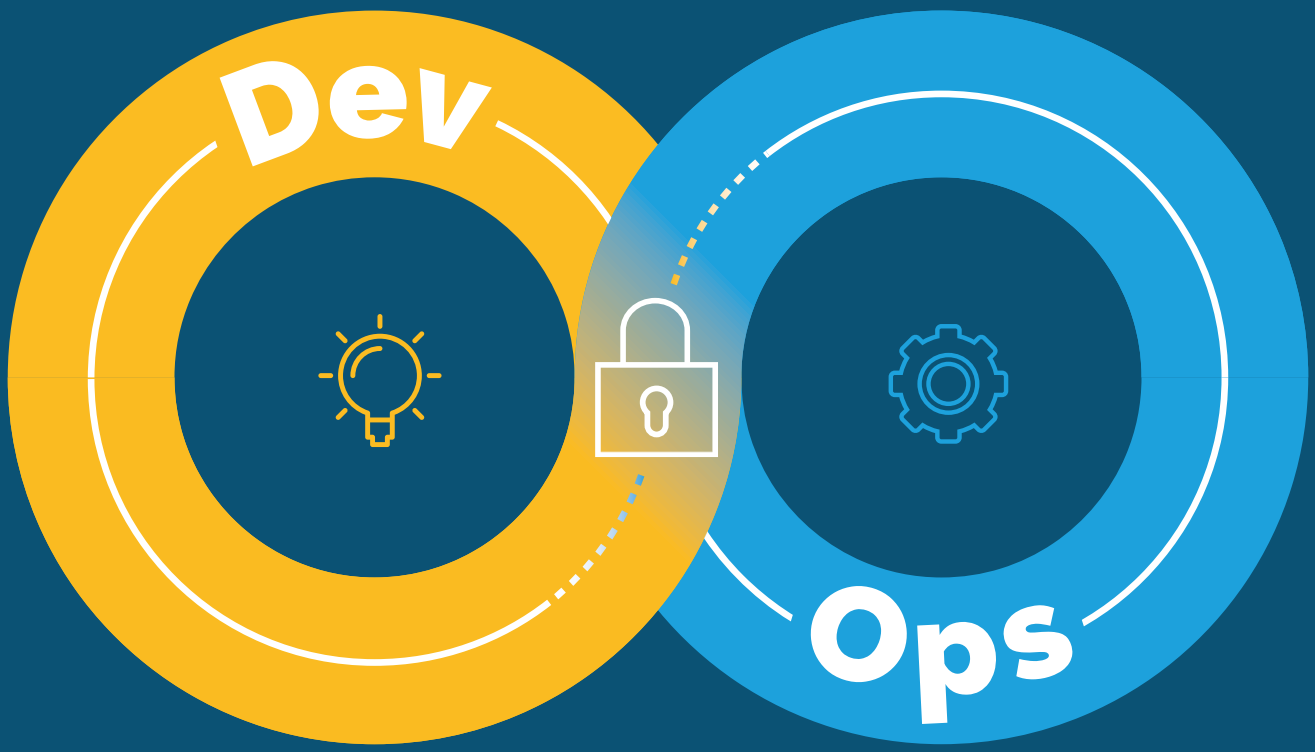


The DevOps Guide to Application Security

A new, collaborative approach for securing apps

Outpost24

Outpost24

This white paper is for

- IT security and risk managers responsible for app security
- Agile DevOps managers who need to understand the new, collaborative approach to Appsec

What you'll learn in this whitepaper

1. Why security is the #1 challenge for agile DevOps
2. A new, collaborative approach for securing apps … helps in-house and outsourced agile developers
3. How to help your DevOps team operationalize security testing or implement a robust, mostly automated Appsec program for agile development ("DevOpsSec")

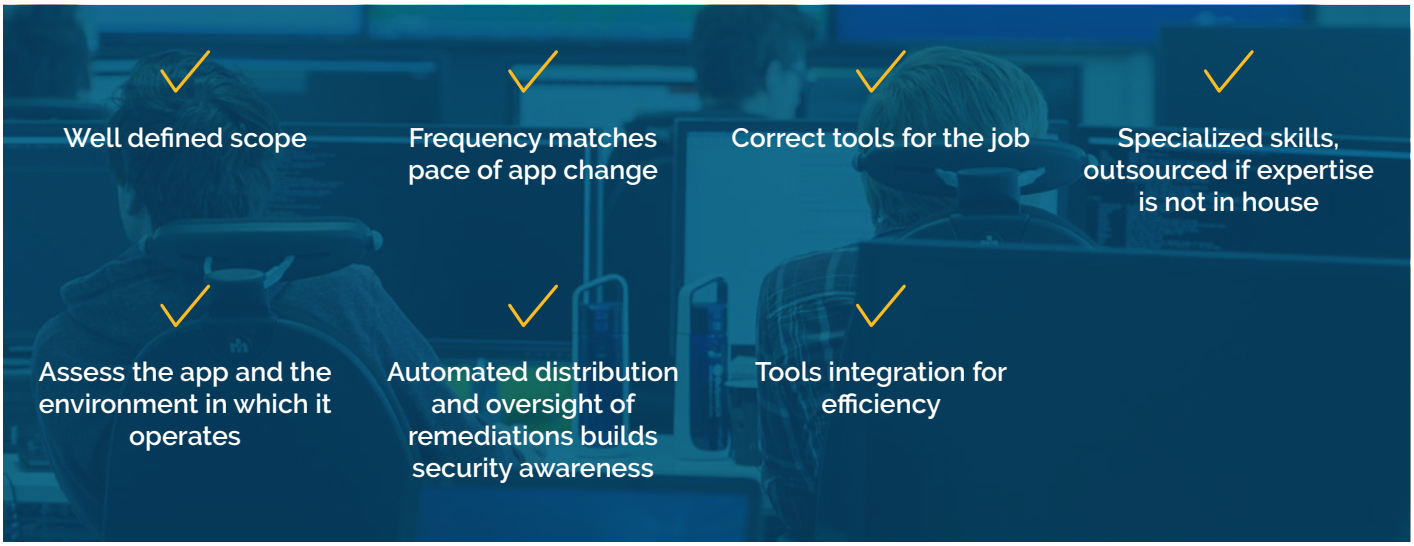
Executive Summary

From improving customer experiences to streamlining processes, applications created with an agile process can be a major asset to your company. They can also introduce new security risks – especially when their security does not keep up with speedy app changes. Getting visibility and control of Appsec for DevOps is vital, especially as virtual apps scale to touch all parts of your business processes.

The risk is a clear and present danger: vulnerable web apps are the #1 vector exploited in breaches reported by Verizon. The risk is amplified by the rushing workflow of app changes driven by agile development.

Fortunately, there is a new, collaborative model for Appsec that you can programmatically operate in mostly automated form. The collaborative approach will help your security team protect apps – whether developed in house or outsourced to external coders.

In this white paper, we'll describe the high points of who does what and how you can tune or implement an effective Appsec program for DevOps. Elements of a successful Appsec program include:



> "vulnerable web apps are the #1 vector exploited in breaches"
>
> Verizon report 2017

Outpost24

Why security is the #1 challenge for DevOps

The tsunami of change in application development has spawned a crisis in security. The change, triggered by digital transformation of businesses, has devalued the legacy "waterfall" process of slowly developing and evolving software into a speed-driven DevOps world where everything is an app. With the old way, changes occurred on a predictable monthly, quarterly or annual basis. With agile development, changes to an organization's hundreds or thousands of apps emerge on a weekly, daily, even hourly basis.

The vibrant use of apps brings many benefits. Agile app development aims to constantly improve the customer experience – especially on mobile devices. Apps help automate internal processes for faster results and more efficiency. Apps also can extend efficiencies directly to customers and the supply chain.

But agile efforts to achieve these benefits also increases the risks of creating new vulnerabilities or perpetuating old ones. The potential fallout of risky apps is a specter that organizations cannot afford to neglect! Recent research by Verizon shows that web app attacks were the #1 vector exploited in successful breaches – causing twice as many breaches as the next largest vector.[1]

Due to the major risk of insecure web apps, security and risk managers need to step up a programmatic response. An organization's typical budget for IT security and risk management averages just 5.6 percent of the overall IT budget.[2] By comparison, a recent SANS survey[3] notes 17.6 percent of organizations spend less than 1 percent on app security. Another 10.8 percent spend 1 percent, and 22.6% spend 2-5 percent. While buying solutions for domains of security such as network and data is important, the leading cause of breaches is risky web apps so rebalancing efforts may be helpful for reducing your organization's application risk profile.

Since the way application development has changed, it's important to clarify the new requirements of agile DevOps for app security, and what roles are associated with each phase from development to pre-production and then to production.

A new, collaborative approach for securing apps

An Appsec program for DevOps is a repeatable, mostly automated process with six objectives:

1. Better security
2. Focus for leveraging limited resources
3. Meet compliance policies
4. Build security awareness
5. Measure and shrink the attack surface
6. Maintain attack surface at the smallest level

The new world of DevOps is similar to some aspects of legacy software development but with a hyper-accelerated rate of production. Agile development still requires coding, committing, and eventually reaching a complete build ready for testing for pre-production and production. DevOps injects a new twist by using continuous integration and automated testing to achieve faster build rates. The new approach entails integration of automated security testing with the development lifecycle. It also requires a new level of collaboration with other stakeholders.

[1] Verizon 2017 Data Breach Investigations Report (10th ed.), p. 38.

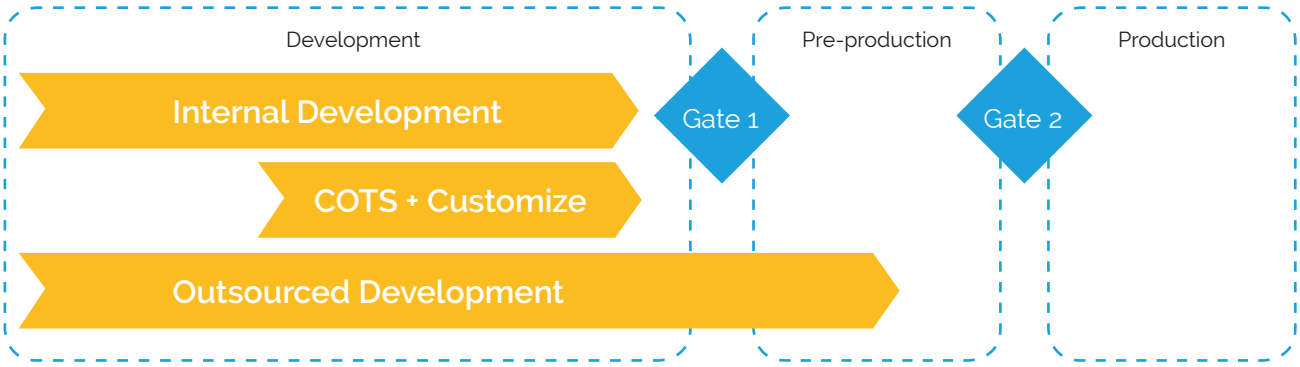
[2] Gartner, IT Key Metrics Data, https://www.gartner.com/newsroom/id/3539117

[3] SANS Institute, 2016 State of Application Security: Skills, Configurations and Components, p14, https://www.sans.org/reading-room/whitepapers/analyst/2016-state-application-security-skills-con%1Fgurations-components-36917

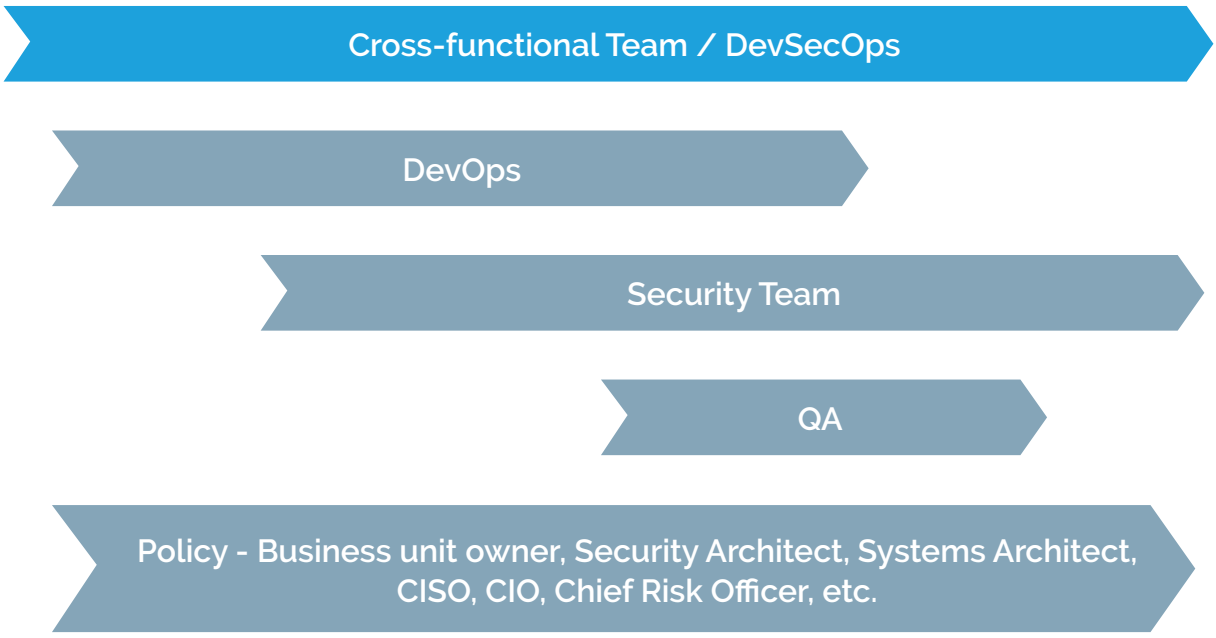
What's new for DevOps is that responsibility for app security does not solely rest in the hands of coders. There are many stakeholders ranging from the business unit owner or CIO to security or risk managers to technical implementers who use testing tools. Policy owners are especially concerned about Appsec affected by significant use of external contract coders, so the new Appsec program needs to address outsourcing. The illustration below shows the phases of agile development with associated roles for security.

Appsec Collaboration for Secure DevOps

Application life cycle



Roles



Enforcing policy with a cross-functional DevSecOps team

The notion of a cross-functional team, called DevSecOps, is a fundamental requirement of the collaborative approach. The DevSecOps team does not require big membership thanks to extensive automation of testing tools. But some coordination is required to promote consistent use of testing tools, effective use of testing data, open communications and transparency for all stakeholders. The cross-functional team should include developers with experience in secure coding best practices – including use of automated testing tools for Appsec. Some app scanning tools can be integrated with automatic IT vulnerability management solutions. The team should therefore include experts in other areas of IT security to help ensure that agile apps do not compromise the confidentiality, availability or integrity of critical business applications or data. Expertise in security and privacy compliance is also beneficial to fulfilling policy and regulatory requirements.

Sometimes your available staff may not possess all the experience required for a cross-functional team. In this case, an external outsourcing provider such as Outpost24 can fill in the skills gap and keep the Appsec program in motion.

The result of a collaborative approach to Appsec for DevOps is the ability to test apps continuously with minimal effects on your bottom line. Agile development requires speed, depth, or a combination of both. And the two biggest drivers to achieving this in application security testing are cost and time. In the next section, we describe how an Appsec program for DevOps can use the right tools to help keep costs low and achieve rapid production requirements of an agile process.

How to tune or implement your Appsec program

Whether designing a brand new Appsec program for DevOps from scratch, or by tuning an existing program with more efficient tools or processes, Industry best practices should guide your organization's approach. One example is the U.S. National Institute of Standards (NIST) Cyber Security Framework. Version 1.1 was released in April 2018. Its five-function framework of Identify, Protect, Detect, Respond, and Recover address asset management (apps, infrastructure and data), risk assessment (app security testing and vulnerability assessment), and risk management strategy including test results.

App security will not entail every aspect of cyber security! For example, while the NIST framework provides 22 categories within the five functions, just three apply to Appsec. These are asset management, risk assessment, and risk management strategy. The tools and processes you select will constitute the framework "controls" used to ensure that application security is automatically built into DevOps for all apps.

Structure your program to counter "The Hacker Pivot" and help your team and AppSec defenses stay nimble. The pivot represents a hacker's modus operandi: (1) establish objective; (2) attack multiple entry points, and (3) move laterally to objective. Addressing the pivot requires use of a variety of automated and human-based tools applied wherever apps are vulnerable.

Choosing the right tools

Depending on your requirements, there are six types of vulnerability detection tools that may be used in your Appsec program.

Static App Security Testing (SAST)

These tools automatically examine the code to identify vulnerabilities. They are focused on developers. Chief issues: SAST is "noisy" in that it provides too much information; data is not prioritized.

Bug Bounty App Security Testing

Manual testing of apps by independent security researchers who are paid for each finding. Issues: Bug Bounty programs are complex to initiate and difficult to administer on a systematic basis, which hinders utility for agile DevOps.

Penetration App Security Testing

The pen test is a manual process that provides a deep dive into potential app vulnerabilities. While humans control and execute the pen test, automated app testing tools may also be used by the team. Pen test is the most comprehensive assessment you can use. Issues: the manual process means pen test are slow and rare, typically done once a year and often in conjunction with a compliance audit. The results of a pen test are the state of app security in a snapshot of time; they do not address constant changes and potential new vulnerabilities in an agile process. Pen test are also more expensive than automated testing so they are reserved for critical apps.

Dynamic App Security Testing (DAST)

DAST is the workhorse of Appsec tools. They automatically test the app in an operational setting. Tests leverage knowledge about prevalent vulnerabilities, such as the OWASP Top 10 and Common Vulnerability Scoring System (see sidebar, "Major App Vulnerabilities"). Issues: DAST does not test for unknown vulnerabilities and it cannot address all use cases; it also requires some human intervention such as determining the most critical business logic for testing.

Interactive App Security Testing (IAST)

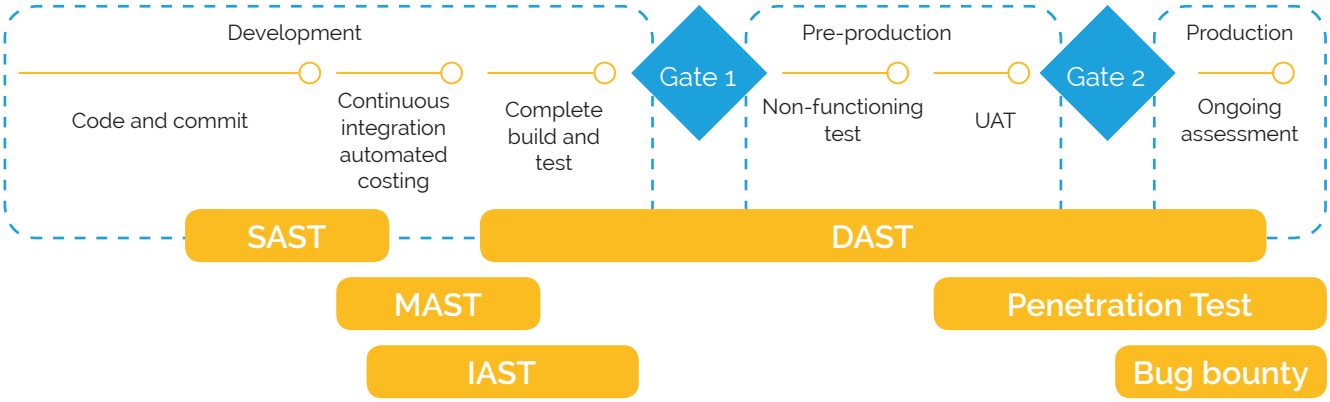
Employs an agent or code inside the app to enable real-time security data – including evidence of an attack. IAST is related to Runtime App Self Protection (RASP). Used for testing apps developed in-house or by contractors; not used for commercial apps. Very useful for capturing security data in virtual apps and hybrid environments – particularly app instances in Infrastructure-as-a-Service or Platform-as-a-Service and containers.

Mobile App Security Testing (MAST)

Partially automated MAST is focused on developers. It's used to discover vulnerabilities in non-standard configurations, such as jailbroken OSes and mobile devices. Issues: requires many variants of the test target and manual coordination of the testing process.

Use the right tools at the right time

There is an appropriate time to use each of the app testing tools described above. Timing is pegged to the app lifecycle, which we first correlated with roles of the people involved with each step. The diagram shows where each of the testing tools are applied for Appsec.



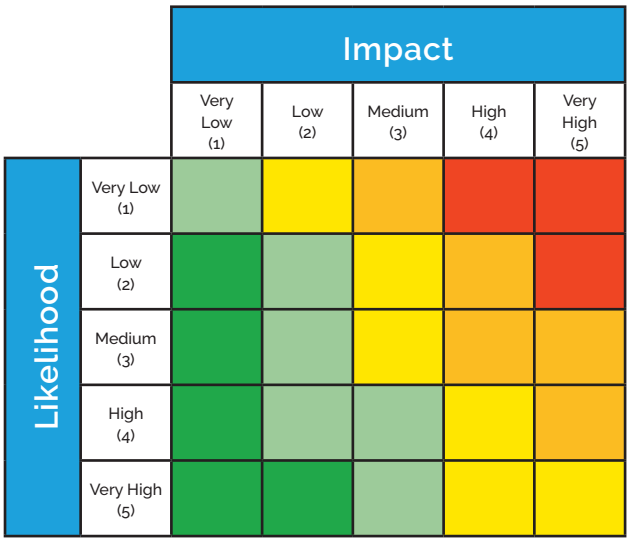
Test in the right places

App security does not exist in a vacuum. Apps can use the entire IT infrastructure so your Appsec program needs to address vulnerabilities in two domains: owned infrastructure and cloud infrastructure.

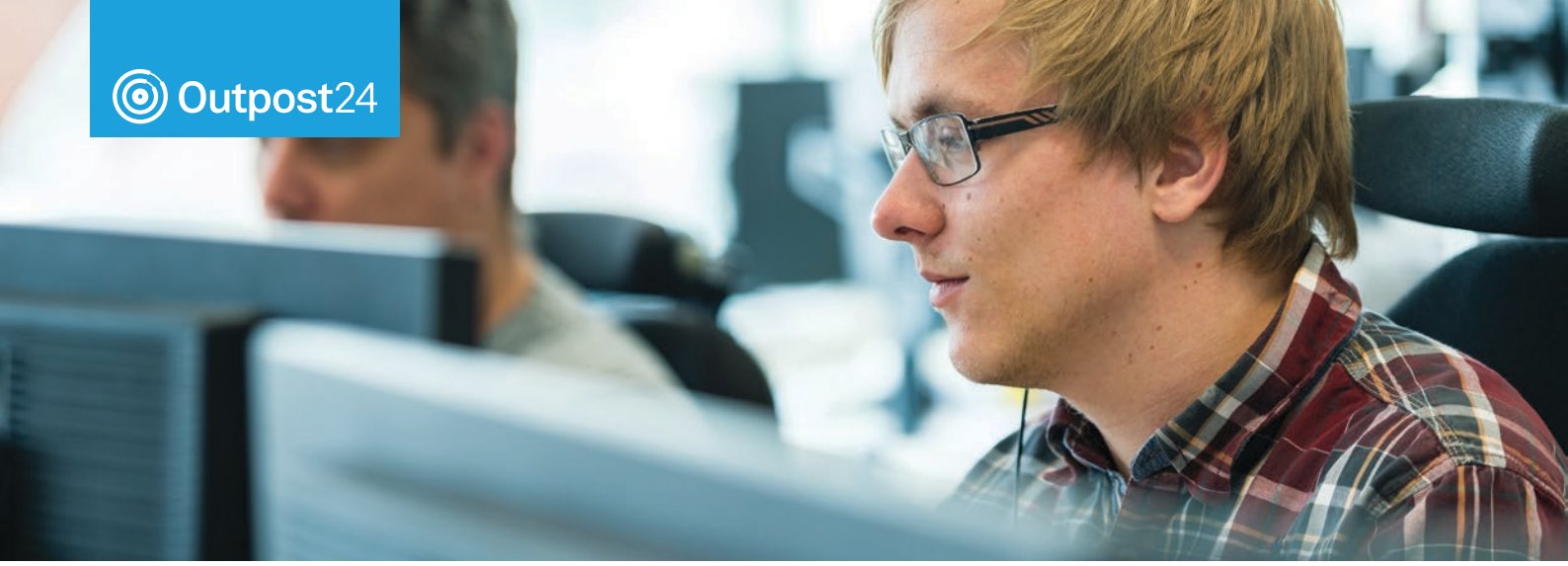
- Owned infrastructure – Testing can be automated and should include network scanning and vulnerability assessment of OS and virtualized OS; installed software; and network configuration.
- Cloud infrastructure – Testing can be automated but authorization for scanning usually must be granted by the cloud service provider. Scanning should include instances (Infrastructure-as-a-Service and Platform-as-a-Service) and containers; and vulnerability assessment of installed software and cloud configuration.

Use risk-based scoring

The end result of Appsec testing is the score, or usually many scores that represent vulnerability data and exposure to risk. A primary objective should be integrating the results into a clear comparative display of priority for remediation. Usually you are on your own to synthesize results from a variety of tools from different vendors or open source projects. Some providers have solutions that do this for you, so use what you can to simplify the process. The overall risk score is expressed as an equation: Simple risk = f (likelihood, impact). This value is not directly calculated from a test result. It factors those results by corresponding business risks to determine severity of a



The Appsec program should distribute scoring and data to stakeholders based on their role in securing apps. Ticketing remediation for specific vulnerabilities should go to respective operations specialists: system owners, data owners, DevOps, and service management. Business and policy owners will focus on building security awareness, understanding trends, shrinking risk, and avoiding institutional perpetuation of the same mistakes in code and configurations.



Major App Vulnerabilities

The leading model for understanding Appsec vulnerabilities is the OWASP Top 10, provided by the Open Web Application Security Project. It identifies the most critical Appsec vulnerabilities and provides risk-based information to assess the likelihood of threat agents and vulnerabilities, and potential impact from technical and business perspectives. The OWASP project is an open community helping organizations to develop, purchase and maintain apps and APIs that can be trusted.

The OWASP Top 10 – 2017 includes:

A1 – Injection (was also #1 in 2013)

A2 – Broken Authentication (was also #2 in 2013)

A3 – Sensitive Data Exposure

A4 – XML External Entities (XXE) – (new)

A5 – Broken Access Control

A6 – Security Misconfiguration

A7 – Cross-Site Scripting (XSS) – (was #3 in 2013)

A8 – Insecure Deserialization – (new)

A9 – Using Components with Known Vulnerabilities

A10 – Insufficient Logging & Monitoring – (new)

OWASP
Open Web Application Security Project

Another model is the Common Vulnerability Scoring System (CVSS) provided by FIRST, an international confederation of trusted computer incident response teams. The CVSS is used for comparing vulnerabilities in applications, infrastructure and data. It can be very useful for Appsec because implementation of malware frequently entails subversion of networking and data. Integration of all the vulnerability data helps give your organization a clear view of Appsec risks in agile environments.

Outpost24

Conclusion

Institutional risks of insecure apps are fueled by the constant pressure to innovate and publish updates with agile development. The demand for speed leads to inevitable lapses. The risks can grow significantly as organizations outsource more and more coding to independent shops. To control these risks, your team needs to get visibility of vulnerabilities in apps – both in owned infrastructure and in the cloud. Full visibility is the enabler of remediation, and the way to achieve this is with an Appsec program. If you have one now, perhaps it only needs tuning. If you don't have one, now is the time to establish your path to app security.

The white paper described a new, collaborative model for Appsec that allows you to programmatically detect app vulnerabilities mostly with automated tools. The collaborative approach will help you operationalize security testing and protect apps – even if development is outsourced to external service providers.

We urge you and your organization to consider the benefits of implementing an Appsec program for DevOps. Please contact Outpost24 to learn more by visiting outpost24.com

Get a Free Appsec Program Assessment

Contact us for a free assessment of your organization's Appsec program, which will help you strengthen app security with agile, continuous use of DevOpsSec best practices and automated tools. The assessment will focus on your organization's ability to test continuously with minimal effects on your bottom line. This includes assessment of programmatic ability to test apps in three ways:

- Automated testing for quickly acquired business-related results based on the last scanned-for vulnerabilities.
- Scheduled penetration testing for a thorough view of vulnerabilities – especially when making major changes to an app.
- Hybrid testing to combine the benefits of automation and human expertise.

Contact us at
outpost.com/getdemo
to improve your appsec program

75%

of organizations have experienced a

50%

or greater reduction in vulnerabilities since using Outpost24.

Outpost24 customer survey, 2017

About Outpost24

Outpost24 is an innovator in identifying and managing cyber security exposure by enabling its customers to achieve maximum value from their evolving technology investments. By delivering insights that reduce vulnerabilities and attack surface across the full stack, Outpost24 customers continuously improve their security posture with low effort. Over 2,000 customers in more than 40 countries around the world trust Outpost24 to inspect their devices, networks, and web applications, cloud infrastructure, and report compliance status to government, industry sector, or internal regulations. Founded in 2001, Outpost24 serves leading organizations across a wide range of segments including financial and insurance, government, healthcare, retail, telecommunications, technology, and manufacturing. For more information, visit outpost24.com

Outpost24 Headquarters
Skeppsbrokajen 8
SE-371 33 Karlskrona, Sweden
Phone: +46 455 612 300
info@outpost24.com

Outpost24 US
50 South Main Street, Suite 200
Naperville IL 60540
Phone: +1 (630) 352 2283

Outpost24