DevSecCon

DevSecOps Whitepaper

The business benefits and best practices of DevSecOps implementation

DevSecCon.com

Index

About this Whitepaper

The DevSecOps whitepaper is a collection of ideas and it's free for everyone to use, augment and add to. It has been written to create a basis for anyone who wants to learn more about DevSecOps and for companies to measure their security programs against. Feedback and ideas are always welcome, you can contact us at: info@devseccon.com.

Author

Francois Raynaud

Francois is the founder of DevSecCon – a conference that centres on DevSecOps and continuously secure delivery. As an independent IT Security Advisor he is actively involved in implementing DevSecOps projects and has 17 years of industry experience working with FTSE 100 and Fortune 500 companies

Reviewers

A big thanks to the people who have helped to improve, expand and launch this whitepaper:

Akash Mahajan
Felipe Zipitria
Madhu Akula
Phil Parker
Robert Davis
Robert Hurlbut

Why DevSecOps?

Two seemingly contradictory imperatives are bearing down on the modern global enterprise organisation.

On the one hand, security is a huge challenge that can have dire consequences if improperly handled. A known vulnerability led to TalkTalk being hacked in 2015, for example, resulting in a record breaking fine for the company and a massive brain drain as embarrassed IT professionals sought to distance themselves from the brand.

On the other hand, as software continues to 'eat the world', high velocity IT becomes the foundation of competitiveness in the modern marketplace. Every business should become an agile and innovative software delivery machine in order to survive. Which leads us to the enterprise IT paradox: Go faster and innovate. But always stay secure.

The enterprise IT paradox: Go faster and innovate – but always stay secure

Security Threats are multiplying exponentially

Cybercrime in its various forms is expected to cost the world more than US$6 trillion per year by 2021. The global cybersecurity skills shortage is expected to grow in inverse proportion to this, with an estimate of over 1.5 million security jobs unfilled by 2019. These statistics together indicate an oversight and lack of commitment on the part of both governments and business in the past to take the necessary steps to fight cybercrime, until, of course, they suffer a data breach themselves. [1]

[1] https://www.wilsoncenter.org/sites/default/files/cybersecurity_in_mexico_an_overview.pdf

While business and government shift their attention and resources to the need for investment in security, the list of data breaches grows longer. Breaches in the US since 2013 include Target (data from up to 40 million credit and debit cards stolen in 2013), Anthem (80 million patient and employee records hacked in 2015), Ashley Madison (more than 30 million user accounts hacked and released to the public in 2015).[2]

UK incidents include Sports Direct (employee data compromised in early 2017); Three Mobile (over 76,000 accounts hacked in 2016), Tesco Bank, and the telecoms company TalkTalk which suffered a data breach in 2015 that cost the company a total of £80 million in fines. Australia led the APAC (Asia-Pacific) region in reported data breaches in 2016.

The breaches referred to here, and the recent global 'success' of WannaCry ransomware (May 2017), which utilised a well-known, but unpatched, system flaw, demonstrate the consequences of ignoring security as a business priority.

When organisations suffer a data breach, companies do not only incur the cost of data damage and destruction, stolen money, IP theft, business disruption and reputational harm. Other costs, such as legal and PR fees, drops in share price, interruptions to e-commerce, loss of customers and competitive advantage can also impact organisations affected by cybercrime.

Too often security is an afterthought, the 'poor relation' in the Software Development Cycle

A more positive consequence is that the entity affected by a data breach focuses on improving security, and recognises software security as a business priority. Too often, until a breach occurs, security is an afterthought, the 'poor relation' in the Software Development Cycle. A central tenet of DevSecOps is that security is an integral and essential element of DevOps.

In 2015, breaches like those described above, (and several breaches of US federal agencies, including the IRS and the Office of Personnel Management), prompted President Obama to declare a national emergency to deal with cybercrime.

[2] http://fortune.com/2015/10/02/heres-whos-been-hacked-in-the-past-two-years/

The US government has since announced massive investment (over US$19 billion) in cybersecurity. US corporations have also increased cybersecurity budgets in the war against cybercriminals, including JPMorgan Chase (but only after a major breach in 2014); Bank of America, who announced an unlimited budget for combating cybercrime, and Microsoft, who will invest over US$1 billion annually on cybersecurity research and development in the coming years.[3]

What is DevSecOps?

Increasing governmental measures to combat cybercriminals, and punish organizations that don't protect their customers' data, mean that security, and security risk management gain ever greater currency.

DevSecOps is the answer to integrating these various challenges into a coherent and effective approach to software delivery. It is a new method that helps identify security issues early in the development process rather than after a product is released.

DevSecOps can reduce the costs associated with fixing security flaws, by building security into every stage of the development process, from the requirement stage onwards.

Privacy and security principles should be integral to any company's culture via DevSecOps best practices, and they should be endorsed at board level. Security must be part of the application development process. **DevSecOps makes everyone responsible for security.**

[3] https://www.unsw.adfa.edu.au/australian-centre-for-cyber-security/sites/accs/files/uploads/ADVANCED%20TECHNOLOGY%20THREATS%20AND%20AUSTRALIA%2030%20May%202106mediaversion.pdf

Benefits of DevSecOps

Companies that embrace DevSecOps benefit from numerous advantages.

The following list is a summary of what can be achieved by implementing DevSecOps best practices:

- **Cost reduction** is achieved by detecting and fixing security issues during the development phases which also increases the speed of delivery.
- **Speed of recovery** is enhanced in the case of a security incident by utilising templates and pet/cattle methodology.
- **Threat hunting** can avoid bad publicity, and therefore can potentially increase sales. It is obviously easier to sell a secure product.
- **Immutable infrastructure** allows companies to tear down infrastructure while managing an attack vector identified by scanning. If a node is compromised, it won't remain compromised for long, as it will be torn down and rebuilt with new credentials. Zero defects in the code is the ideal to aim for, although zero variations are the minimum requirement.
- Immutable infrastructure improves overall security by **reducing vulnerabilities, reduces insecure defaults, and increasing code coverage and automation**. It also encourages companies to move to the cloud instead of using depreciating and increasingly vulnerable hardware.
- **Security auditing, monitoring, and notification systems** are managed and deployed so that they can be continuously enhanced, to keep in step with the frantic innovation intrinsic to cybercrime.
- **DevSecOps ensures the 'secure by design' principle** by using automated security review of code, automated application security testing, educating, and empowering developers to use secure design patterns.
- **Creates targeted customer value** through secure iterative innovation at speed and scale.[4]
- **Everyone** is responsible for security.
- **DevSecOps fosters a culture of openness and transparency**, and does so from the earliest stages of development.
- **The ability to measure** different things which can be seen by everyone. This also enables a culture of constant iterative improvements.

[4] 2015 ISACA Ireland: Embracing DevSecOps to support Rugged Innovation at Speed and Scale, http://www.devsecops.org/presentations

Best Practices

Successful security programmes involve three intersecting parts: people, processes, and technologies.

DevSecOps is no different, but DevSecOps recognises that security is the responsibility of everyone in an organisation, and everyone has a role to play in security.

This section will explore key DevSecOps practices across its three key pillars: people, processes and technology.

People

People are the starting point of the DevSecOps implementation. Through ensuring proper training and restructuring of teams security will become a frame of mind rather than a hindrance.

Processes

DevSecOps aims to align and implement processes common to an enterprise to facilitate cooperation and achieve more secure development processes as a whole.

Technology

Technologies enable people to execute DevSecOps processes, which aim to reduce the enterprise attack surface and enable effective management of the technical security debt.

People

No matter how many technologies you implement, the weakest link will always be the human factor. This is the starting point for any DevSecOps implementation.

One of the most important, but equally most difficult aspects of DevSecOps is challenging the way traditional security teams integrate with the wider business. Most people approach risk from a place of denial rather than acceptance and preparation. Changing habits and raising awareness across all levels of a company are not easy tasks and require a top-down approach if attitudes are to change.

Security needs to shift from being exclusive to being inclusive to facilitate this culture change. By integrating security teams within development teams, as DevSecOps does, companies will get earlier feedback on the quality, from a security point of view, of the code, software or application, thus reducing the costs associated with implementing these fixes.

The human factor is the starting point for any DevSecOps implementation

Hiring security specialists, giving them a voice in project delivery, and allowing them to integrate their processes in the agile development world will deliver the necessary results. Agile development helps to speed up product release dates, but often at the cost of neglecting security. Appointing security champions and providing good training will also ensure security is a priority in your organisation.

Breaking Down Silos and Integrating Security Personnel

For security to be effective, we need to include security personnel as early as possible in the software delivery lifecycle. One way of doing this is by training security champions in the development team.

Security Champions are members of a team that help to make decisions about when to engage the Security Team. Security champions act as the "voice" of security for a given product or team, and they assist in the triage of security bugs for their team or area.[5]

[5] http://blog.diniscruz.com/2015/10/what-are-security-champions-and-what-do.html

The Microsoft Agile SDL states that the Security Champion does not have sole responsibility for ensuring that a software release has addressed all security issues, but is responsible for coordinating and tracking security issues for the project. This role is also responsible for reporting status to the security advisor and to other relevant parties (for example, development and test leads) on the project team.

Security Champions are a key element of the DevSecOps methodology, since they are the first step to creating a cross-functional team focused on Application Security and Security Operations.

Cross-functional teams are created from Subject Matter Experts, influencers and diverse members to foster serendipitous conversation and tackle issues outside of the boundaries of rigid meetings.

We need to include security personnel as early as possible in the software delivery lifecycle

Some of the most important duties of the Security Champion include the following:

- **Ensure that security is not a blocker** on active development or reviews
- **Be empowered** to make decisions
- **Work with AppSec team** on mitigations strategies
- **Help with QA and Testing**
- **Write Tests** (from Unit Tests to Integration tests)
- **Help with development of CI** (Continuous Integration) environments
- **Keep track of and stay up to date** on modern security attacks and defences
- **Introduce body of knowledge** from organisations such as OWASP (Top 10, Application Security Verification Standard, Testing Guide etc.)

Training

Any successful programme will invest in good training and professional development for its staff. To foster and develop good security staff, organisations must provide new hires with the appropriate training and tools they need to do their jobs well, and to contribute to the successful release of secure software. Engaging specialist security and DevOps training organisation(s) to raise staff skills and awareness are essential for maintaining consumer trust.

Training can be computer-based, instructor-led, or a combination of both. Good training ensures that standards are implemented correctly. Training must be rooted in company goals, policies, and standards for software security, and learning media must be flexible and tailored.

Although software developers are typically not meant to become professional pentesters, it is still valuable to teach them about the attacker's perspective, and about practical hacking exercises and vulnerable applications.

It is valuable to teach developers about the attacker's perspective, practical hacking exercises and vulnerable applications

People – Conclusion

The correct DevSecOps processes and technologies will not be able to achieve anything if the company culture - embedded in people across all areas of the business - does not enable them to be properly utilised.

The security team has traditionally been a drag on release performance, the 'naysayers' who come along at the end of a development cycle and add poke holes in the product and force parts to be fundamentally rethought far too late on in the process. As a result, the security team is marginalised over time, creating a self-reinforcing downward spiral of division between teams.

DevSecOps aims to break down these barriers and stop security being its own echo chamber without taking into consideration the wider business when implementing policies or tooling. Proper training, a restructuring of teams and the appointment of security champions means that 'security' becomes less the function of a department and more a frame of mind that permeates the company. This sets the foundation for the successful implementation of security processes and technologies, making for enhanced security much earlier on in any project and quicker, easier and cheaper software delivery cycles.

Processes

People implement the processes in any organisation. Although processes have been siloed per team and are often not interconnected with other teams, thus not productive within an enterprise. DevSecOps aims to align and implement processes common to an enterprise to facilitate cooperation and achieve more secure development processes as a whole.

The following sections describe the essential processes of DevSecOps.

Version control, metadata, and orchestration

Within an automated world, the only constant is change, and change needs to be both consistent and traceable. To track all changes, we must ensure that adequate and immutable versioning is in place. Every action needs a version in the same way that code is managed to allow quick recovery. Once turned into metadata, operational teams can efficiently track a change and measure it.

Orchestration software doesn't only provide a repeatable way to deploy infrastructure, it also provides a huge amount of metadata regarding any task. The metadata can in turn be used not only by the orchestration software itself, but as an authoritative source for integrated tooling. Once coupled with versioning, orchestration software becomes a powerful source of information for all operational teams.

Within an automated world, the only constant is change, and change needs to be both consistent and traceable

Integration of processes

Integrating information security into agile development enables organisations to have a fully secure workstream through every single stage of the project development cycle. In the agile world, the integration of security must start at the earliest possible stage which, in most cases is the requirement definition stage. This methodology has been called "shift security to the left" and it strives to reduce the cost of implementing security.

Codifying security requirements and checklists (which are required for integration) allow for a built-in type of development rather than the bolt-on approach which is typically followed.

Security tooling in CI/CD

Wouldn't it make more sense to let the operations teams run the security tooling as part of their pipeline? Security has fought against shadow IT for a while, although it created its own shadow IT by having separate tooling for security. If you take Vulnerability Management and hook it to your pipeline via APIs, you can then let the orchestration call them for every build.

Security sets the requirements and DevOps manages the frequency of scan occurrences according to the development practices.

Compliance

Implementing compliance doesn't have to be a paper-based exercise. We can create metadata representing the compliance requirement and integrate it in our assets. This can also be used by security policy automation by tagging assets that can in turn implement the desired security architecture, for example, zoning.

By coding your compliance requirements, you can meet the GDPR standard

Imagine the ability to respond to a breach under the new GDPR rules in 72 hours. By coding your compliance requirements, the task would be much simpler.

Security Architecture

Security architecture is supported by a set of principles that are specific to each company. These principles depend on the type of data being processed, although a high-level set of principles can be used to guide software delivery towards more secure practices.

Coding these principles are part of your DevSecOps methodology and having them built in the requirements stages allow product managers to seamlessly integrate security as part of their plan and supporting architecture. If there is any deviation due to business process and/or lack of resource, the deviation is captured and the risk associated with it taken into account.

Incident Management

Responding to security incidents should not be an improvised or non-scripted activity. It is key that workflows, action-plans, playbooks and runbooks are created in advance. This is to ensure that the response to an incident is consistent, repeatable, and measurable. Incident management should make use of the metadata to help simplify this process, thus changing the metrics to highlight the time taken to redeploy a compromised asset.

In turn, once the playbooks have been codified, they can be integrated in your CI/CD to automate them. In a DevSecOps world, proactive and pre-emptive threat hunting, and continuous detection and response to threats and vulnerabilities mean that there are fewer major incidents and more mitigations. The use of red teams and bug bounties also mitigate against breaches. While continuous detection is a great thing, never stop watching out for standard notification and alerting fatigue.

Red Teams and Bug Bounties

All companies should deploy a red team to hunt for threats as part of the DevSecOps methodology. Red teams are built from security team personnel and usually virtual to facilitate its ad hoc nature.

Instead of discussing what is wrong with an application, the red team demonstrates what is wrong and provides the solution. This allows a positive feedback loop between security and the developers, demonstrated by clear recommendations to improve the software quality. Create playbooks from the red team's kill chain, and convert them as negative tests to simulate attacks.

Create playbooks from the red team's kill chain, and convert them as negative tests to simulate attacks

All companies should also, occasionally, implement bug bounty programs. Bug bounties are rewards given to researchers, usually company external, for finding and reporting a bug in a software product.

Threat Intelligence

Threat intelligence should follow similar procedures to those of the red team. Compare the threat intelligence data collected from 3rd party providers, monitoring devices to your existing automated playbooks and update them with current data. Then replay these playbooks against your templates and artefacts.

Processes – Conclusion

Processes are key to the success of DevSecOps. Their aim is to create agreed and repeatable ways of working which are clearly documented and public to the company to ensure transparency of the security towards the rest of the business. If these agreed ways of doing security (sets of principles, playbooks, as defined above) are implemented, problems (faults, bugs, threats etc.) can be automatically identified much sooner and responded to in an agile fashion.

Where, prior to implementing proper DevSecOps processes, organisations would respond too late and too slow, DevSecOps makes short, feedback-driven security loops possible that quickly identify problems and react swiftly.

DevSecOps enables short, feedback-driven security loops that quickly identify problems and react swiftly

Technologies

Technologies are what enable your people to properly execute DevSecOps processes. This section outlines the required technologies to implement a successful DevSecOps methodology within your enterprise.

Automation and Configuration Management

Leveraging the automation and using orchestration to implement DevSecOps is key to success. Orchestration and automation make auditing easier: the use of metadata makes decisions easier as they are based on data points and repeatable processes. The use of templates within configuration management helps to implement traceability of each code/configuration change, thus making it easier to identify the root cause of an issue and any deviation from immutable artefacts.

Secure coding practices/Security as Code

All coding standards must be constantly checked against new security recommendations. All changes to the code need to be verified and tested against these recommendations: no change is too small to avoid in this process. This is not a trivial exercise, and the benefits associated with such practices should not be underestimated; they are not limited to the amount of changes occurring in the development lifecycle.

The OWASP Top 10 is a great place to start this review by converting the code changes into your QA testing, taking advantage of the automated testing facility to provide just-in-time feedback to the development teams. Additionally, the OWASP ASVS with its 19 verification domains lends exceedingly well to the craft of building secure software.

With the ever-increasing pace of new software development techniques and frameworks, Attack Driven Development lays out a process through which developers can learn about the tools, techniques, and procedures for software development and application security in parallel.

Host Hardening

The practice of host hardening is not new, but if it were used more often, fewer services and applications would be unnecessarily exposed to the Internet. Countless examples of security incidents can be directly related to leaving a generic attack surface that allows automated attack tooling to succeed in the most basic attacks. The hardening checklist and methodologies are mature enough to be easily included in the creation of templates to reduce the attack surface and reinforce a trust model. The latter can be codified as metadata for further processing by the CI pipeline, and then used for other processes such as patching.

CI/CD for Patching

Once your metadata has been associated with each asset, we can use this data to implement patching at the CI/CD level. Feeds from Threat intelligence and Vulnerability Management are compared to the deployed software stack to identify matches in the templates in turn queued for deployment. Patching live systems becomes a thing of the past, thus limiting the impact of downtime. This will also provide the ability to have a risk exposure in near real time.

Application-level Auditing and Scanning

Auditing and scanning are a crucial aspect of DevSecOps that allows business to fully understand their risk posture. Each of the following solutions represents a higher degree of security assurance of the code, as reflected in the organisation's risk appetite:

Source Code Scanning

Source code scanning should be covered by implementing Static Application Security Testing (SAST). SAST is used for scanning the source code repository, usually the master branch, identifying vulnerabilities and performing software composition analysis. It can be integrated into existing CI/CD processes. Having a SAST tool integration in place enables remediation of vulnerabilities earlier in the software development lifecycle, and it reduces application risk and exposure.

Dynamic Application Scanning Tool (DAST)

Dynamic Application Scanning Tools are designed to scan the staging and production website in running state, analyse input fields, forms, and numerous aspects of the web application against vulnerabilities.

IDE Integration

IDE integration and static code analysis plugin allows the developer to have an enhanced view of the problems in the code within the integrated development environment. This provides an effective way to optimize and mitigate vulnerabilities straight away without needing to leave the development environment.

Binary Scanning

All binaries must be scanned for security issues derived from the coding checklist, and then the binaries must be digitally signed. The digital signature is treated in the same fashion as the metadata. For example, within the CI, only signed binaries can be used and implemented, thus ensuring the correct level of security sign-off without having to wait for free cycles from the security team.

Pre-Deployment Auditing

Using a pre-defined template for building assets is essential to ensure the desired security level, although this should be supplemented by host-based scans. Most security scanners now provide a compliance module that allows you to import your template.

Post-Deployment Auditing

These predefined templates once instantiated can be checked for any delta against the pre-deployment scans to identify any changes which may introduce security threats. This should be achieved by using API integration for obvious automation purposes.

Auditing and scanning allows business to fully understand their risk posture

Automated Vulnerability Management Scanning

All vulnerability management software in use should be capable of being integrated via API for infrastructure and web application scanning.

The real-time correlation of active threats against identified vulnerabilities helps to identify the following:

1. What **assets** are subject to known exploits
2. Any new **threats** that may pose an immediate risk to the business

The vulnerability management processes should be fully integrated with the developer bug-tracking system.

Vulnerability management software should be capable of being integrated via API for infrastructure and web application scanning

Automated Compliance Scan

Compliance can be achieved using automated security configuration assessments to reduce risks and maintain continuous compliance. This helps to cut compliance costs by reducing the effort and time required to assess the systems, and it allows the sharing of compliance data with the business GRC tool and help-desk applications to provide visibility of the compliance status.

Managing Secrets

'Secrets' in an Information Security environment include all the private information a team should know, for example a database, or a third-party API. To establish a trusted connection, credentials, or a certificate, or an API token are necessary, but even with these precautions, handling secrets can be challenging, and can often become a source of error or even a security breach.

Techniques that make the task of handling secrets easier include having a constant in the source code, or storing secrets in a configuration file that is not checked into version control. These techniques solve some problems, but they generate their own challenges, particularly for key rotation.

The ideal is a synchronised, encrypted, shared password store, that can be decrypted by all team members individually, but without the use of a shared password. Two tools are available to achieve this: GPG (the Gnu Privacy Guard) and 'Pass'. GPG allows the implementation of a public key infrastructure, and is often used in email encryption. GPG can be complex to use however, so 'Pass', whose developers call it the 'standard Unix password manager', gives users a convenient wrap around GPG. Pass allows you to encrypt secret information with one or more private keys, and all the encrypted information is stored as flat files in one directory that can be shared using version control. These tools facilitate an encrypted, shareable pool of information that is still secure.

Aim for a password store, that can be decrypted by all team members individually, but without the use of a shared password

Effectively managing secrets using tools like GPG and Pass are an essential element of DevSecOps, as they work from request to creation and distribution, ensuring security right along the chain.

Technologies – Conclusion

Technologies are key and the successful implementation of the ones listed above will greatly reduce the enterprise attack surface as well as the ability to effectively manage their technical security debt.

As well as automating scanning and monitoring, implementing the above technology practices means that security, regulatory and compliance requirements can be embedded as code into the software delivery pipeline to ensure that any code deployed is secure and compliant. Any deviation from this can be spotted early and fixed quickly.

Ownership of these technologies does not need to reside within the security team and properly distributing these to the relevant operational team will help the security team to concentrate on hunting the threat rather than operating the said technologies.

Conclusion

DevSecOps addresses the need for pro-active, customer-focused security that anticipates rather than reacts to data breaches or other cyberattacks.

DevSecOps shifts security from reactive to proactive, supported by different techniques such as Test Driven Development and Attack Driven Defence. It champions the importance of security at all levels of an organisation, and empowers security staff to make decisions that have a positive influence their business. As such, DevSecOps, as both a concept and a practice, is growing all the time, with an increasing number of organisations implementing DevSecOps as a solution to their security issues.

The benefits DevSecOps brings to companies that embrace it are numerous, including cost reduction, speed of delivery, speed of recovery, compliance at scale, and threat hunting. The cumulative effect of these benefits is an enhanced business reputation and a smoother business model. A DevSecOps enterprise will have successfully removed the barriers between DevOps and Security helping them to work as one towards the enterprise business goals without friction.

The DevSecOps enterprise will have the ability to detect and fix security issues earlier in the development process thus reducing greatly the cost associated with identifying and fixing them. Shifting security to the left through the use of people, processes and technology will help to achieve this goal.