2ND EDITION

The DevSecOps Security Checklist

sqreen

INTRODUCTION

Damn, but security is hard.

It's not always obvious what needs doing, and the payoffs of good security show up as the absence of bad events rather than the addition of good events. Who is surprised when it falls off our priority lists?

DevSecOps is a practice that better aligns security, engineering, and operations and infuses security throughout the DevOps lifecycle. We'd like to offer a little help on the journey to DevSecOps. And by « help » we don't mean « pitch you our product »—we genuinely mean it.

Sqreen's mission is to empower engineers to build secure web applications. We've put our security knowledge to work in compiling an actionable list of best practices to help you get a grip on your DevSecOps priorities. It's all on the following pages.

We hope you find it useful. If you do, share it with your network. And if you don't, please take to Twitter to complain loudly—it's the best way to get our attention.

The Sqreen Team
@SqreenIO
howdy@sqreen.com

WANT THIS HANDBOOK AS A PDF? GO TO:
https://www.sqreen.com/checklists/devsecops-security-checklist

DEVELOPMENT

☑ Make security part of the entire development process

Bring security in early in development and throughout the whole cycle. Give its requirements the same weight as the functional requirements. This involves adding security controls and processes, as well as automating the core security tasks in the workflow. This enables developers to address known vulnerabilities up front, hence provide secure and resilient software.

https://www.opcito.com/wp-content/uploads/2017/08/DevSecOps%E2%80%93Building-Security-in-to-Your-DevOps-Processes.pdf

https://www.agileconnection.com/article/devsecops-incorporate-security-devops-reduce-software-risk

http://mattboegner.com/secure_cicd_pipeline_2/

https://www.contino.io/files/Introduction-to-DevSecOps-Best-Practices-for-Adoption.pdf

---

☑ Test your security throughout the development cycle

Make security testing a continuous process and an integral part of the entire app development cycle. Perform tests on applications, APIs, containers, data, processes, and microservices. The earlier you catch flaws, the easier they are to fix, but being able to identify flaws everywhere from development through to production will ensure that you're able to stay on top of vulnerabilities no matter where they surface.

https://securityintelligence.com/three-effective-ways-to-make-application-se-

curity-testing-a-successful-part-of-your-devops-program/

https://owaspsamm.org/

https://www.sans.org/reading-room/whitepapers/analyst/devsecops-approach-securing-code-cloud-37597

☑ Automate as many processes as you can

Automating security, configuration management, testing, and other tasks reduces the workload for your teams while providing a faster way of doing things. Automate functionality and non-functional security tests; application, infrastructure and configuration security tests, as well as application logic security tests.

https://devops.com/automated-security-testing-continuous-delivery-pipeline/

https://jaxenter.com/turn-devops-into-devsecops-156910.html

☑ Monitor your processes, infrastructure, and apps

Gathering real-time intelligence enables you to make better decisions, and accurate enforcement. Collect and analyze relevant metrics, event logs, and machine data to gain real-time insights across the application lifecycle. Monitor your applications in production to ensure that you detect new vulnerabilities and security events. Pursue the opportunity to fix issues, earlier, faster and at little cost.

https://www.sumologic.com

https://www.sqreen.com

☑ Generate actionable alerts when there are issues

Deploy a tool that notifies the team when there is an issue across all of your key focus areas -- security included. This should have the ability to send actionable alerts to the relevant people.

https://www.sumologic.com/blog/log-management-analysis/2-key-principles-creating-meaningful-alerts/

https//www.sqreen.com

CULTURE

☑ Develop a strong security culture

A strong security culture among developers, operations, and security is essential. Develop openness, clear communication pathways, as well as strong feedback loops. Additionally, shift the responsibility for security to all of these teams as opposed to the traditional approach where it was solely the work of the security department. If you make security come from a place of "yes, let's figure out how to do this securely" rather than "no, you can't do that for security reasons," security will move from a blocker to an enabler.

https://dzone.com/articles/shifting-security-left-3-devsecops-challenges-amp

https://tech.gsa.gov/guides/building_devsecops_culture/

https://www.zdnet.com/article/devsecops-is-the-new-devops/

☑ Develop a Security-as-Code culture

Introduce a security-first mindset without affecting the agile practices the developers rely on to produce apps. Encourage your developers to add security to the code as they build the applications by making the secure actions the easiest actions wherever possible.

https://www.networkcomputing.com/data-centers/devsecops-3-things-infrastructure-pros-should-know/316636710

http://www.devsecops.org/

http://aspetraining.com/resources/blog/devsecops-101

☑ Provide training and tools to developers

Ensure that the developers have the required training, support, and tools to perform their tasks efficiently. You should also promote knowledge sharing and create a decision-making process among the different departments to promote team autonomy.

https://enterprisersproject.com/article/2018/6/how-build-strong-devsecops-culture-5-tips

https://thenewstack.io/devops-security-overcome-cultural-challenges-transform-true-devsecops/

https://techbeacon.com/how-build-devsecops-grow-your-security-culture-ground

https://www.alldaydevops.com/blog/devsecops-overcoming-culture-of-no

ENVIRONMENT

☑ Secure and monitor your entire physical and virtual environment

Security needs to be integrated across your environment. Take steps to secure your entire infrastructure, including on-premise and cloud environments, networks, CI/CD pipeline, code, data, operating systems, applications, and software. Use sustainable processes and tools to identify and block internal and external attacks, and malicious traffic and files.

- https://www.sqreen.com
- https://www.threatstack.com
- https://www.cloudflare.com

☑ Gather metrics to gauge success

Security is a journey that never ends, so focus on making progress. Collect and act on security and compliance information from on-premise and cloud environments. Use both the high-value and supporting metrics to get insights and determine the effectiveness of your security processes. Iterate whenever things change.

https://tech.gsa.gov/guides/dev_sec_ops_guide/

https://thenewstack.io/security-metrics-that-actually-matter-in-a-devops-world/

☑ Secure and harden your containers

Follow best container security best practices. Secure authentication and authorization. Inspect, scan, and provide file, image and container security. Use private registries such as GCR or quay. Also, build from trusted and verified container images.

https://blog.sqreen.com/docker-security/

https://blog.sqreen.com/kubernetes-security-best-practices

https://dzone.com/articles/integrating-docker-solutions-into-your-cicd-pipeli

---

☑ Isolate Dockers and Kubernetes

Secure and isolate your containers early, often, and continuously. Isolate and segment containers using tools such as Apparmor, Seccomp, SELinux. Create isolation layers between different applications as well as between applications and hosts. This reduces the host's surface area, hence restricting access and protecting it as well as the co-located container.

https://docs.docker.com/engine/security/userns-remap/#prerequisites

https://www.slideshare.net/ssuser9ebf46/docker-container-isolation-and-security

---

☑ Perform threat modeling exercises

A threat modeling exercise identifies the design flaws and components that are most at risk, and should provide the security team with the opportunity to prioritize and address flaws according to their impact. In particular, threat modeling helps teams to understand the type of assets they are protecting, the sensitivity levels, potential threats, and their impact.

https://medium.com/starting-up-security/how-to-measure-risk-with-a-better-okr-c259bccf359e

https://speakerdeck.com/zeroxten/threat-modeling-the-ultimate-devsecops

---

☑ Automate infrastructure configuration and management

Automate and simplify the configuration and management of servers, infrastructure, compliance, and applications. Use tools such as Puppet, Chef, and Azure Automation Desired State Configuration and other DSCs. These tools can help you further DevSecOps through functionality like automatically provisioning an environment, applying security settings, and deploying apps.

https://devops.com/simplify-expedite-server-management/

https://searchitoperations.techtarget.com/tip/What-is-the-Puppet-configuration-management-tool-and-how-does-it-work

https://docs.microsoft.com/en-us/azure/automation/automation-dsc-getting-started

---

☑ Harden your cloud deployments

Cloud environments can provide a secure infrastructure if implemented properly. Review the teams and individual roles and permissions you've allocated. Grant access to only what each individual or team needs to perform their jobs. Enforce two-factor authentication for everyone. Check the security groups, standard AMIs, IAM roles, MFA tokens, etc.

https://www.sqreen.com/resources/aws-security
https://docs.microsoft.com/en-us/azure/security/security-best-practices-and-patterns

CODE

☑ Code security into your apps

Create secure code from the start of development all the way to the deployed application. Ensure that security is integrated into the code instead of adding it as an afterthought. This requires involving the security teams throughout the development process. Keeping the code and implementations as simple as possible avoids complexities that may compromise security. Implement processes and best practices that make it easy and straightforward for developers to make secure decisions.

https://devops.com/building-security-code-culture/

https://medium.com/nmc-techblog/when-devops-met-security-devsecops-in-a-nutshell-b228386c30f6

☑ Continuously review code at every stage

Review the code and standards at each stage to ensure that they comply with security best practices. Use SAST and DAST to analyze code, and other automatic tools to track dependencies and scan all third party and open source codes. Perform pre-commit, commit-time, build-time, test-time, and deploy-time checks in your CICD pipeline.

https://dzone.com/articles/lets-talk-about-code-reviews

https://www.codacy.com/

https://techbeacon.com/best-open-source-devops-security-tools-how-use-them

☑ Introduce chaos in the comfort zone

Use chaos engineering to test how prepared your systems are to respond to security threats under unfamiliar operational environments. Run scripts to randomly shut down server instances, take down containers in a random manner, disrupt some services, or create unexpected outages in the applications and infrastructure. This helps the teams to provide a moving target defense that protects your systems in a wide range of conditions and ensures that something unexpected won't bring everything tumbling down.

https://blog.sqreen.com/what-is-chaos-engineering-and-why-does-it-matter/

https://solutionsreview.com/devops/2018/03/28/chaos-engineering-interview-gremlins-tammy-butow/

https://en.wikipedia.org/wiki/Chaos_Monkey

https://www.ibm.com/developerworks/library/a-devops4/index.html

---

☑ Maintain an inventory of your applications and components

Create and maintain an up-to-date inventory of your application assets. Get visibility into what's being deployed and what your organization has out there. Maintaining an up-to-date inventory will help you uncover new security insights and prevent you from being caught off guard when something unsecure gets deployed or when security vulnerabilities are uncovered in something old and forgotten.

https://blog.sqreen.com/security-in-depth-introducing-in-app-waf-and-app-inventory/

https://aws.amazon.com/blogs/mt/preventing-blacklisted-applications-with-aws-systems-manager-and-aws-config/

☑ Scan and secure your open source and third-party components

Stay on top of your open source and third-party dependencies. Make sure that they are always up-to-date, and that you regularly verify that they aren't vulnerable.

https://www.slideshare.net/blackducksoftware/devsecops-the-open-source-way-82238900

https://www.sqreen.com

https://github.com/dependabot

☑ Start a security analytics program on your code

Use threat modeling, penetration tests, and vulnerability testing to confirm that your code is secure. Determine the number of severe vulnerabilities, and how long they persist before your team resolves them. Analyze the frequency and scope of automated tests as well as the number and type of attacks on your applications.

https://medium.com/starting-up-security/communicating-risk-across-complex-teams-c088793d4070

https://www.csoonline.com/article/3412064/security-analytics-its-all-about-the-data.html

APPLICATION PROGRAMMING INTERFACES (APIs)

☑ Secure your APIs

APIs enable interaction and sharing of data between applications and therefore are more exposed and prone to security risks. Secure all the APIs the company consumes as well as those it exposes to the public. Use encryption to protect request information in transit while limiting the amount of information in the API error messages.

https://blog.cloud-elements.com/devsecops-for-your-apis

https://www.redhat.com/en/topics/devops/what-is-devsecops

☑ Authenticate and authorize API users

Use API IDs and API keys to identify and authenticate users, devices, or applications. Use an access control framework such as the OAuth to control the APIs that authenticated users or specific API keys can access.

https://cheatsheetseries.owasp.org/cheatsheets/REST_Security_Cheat_Sheet.html

https://oauth.net/

☑ Apply security policies to APIs

Approach API security from both the consumption and exposure perspectives. Manage identity, security keys, tokens, certificate policies, authentication, and authorization policies. Do not forget to log and audit keys, policies, and logs stores.

https://www.cl.cam.ac.uk/~rja14/Papers/SEv2-c18.pdf

https://www.soapui.org/learn/security/state-of-api-security.html

---

☑ Secure all your transmission paths

Secure your transmission paths to prevent data loss and security breaches. Encrypt all connections to prevent Man-in-the-Middle attacks. Enforce SSL/TLS.

https://www.ssllabs.com/ssltest/

https://observatory.mozilla.org/

☑ Validate input data, content types, and responses

Validate all data to prevent application layer attacks. Ensure safe input data from users, database systems, external sources, as well as infrastructure. In addition, perform integrity checks as data crosses the boundary between a trusted and less trusted environment. This ensures that compromised data does not enter into your systems indirectly.

https://www.owasp.org/index.php/Data_Validation

https://www.owasp.org/index.php/Testing_for_Input_Validation

☑ Use RBAC to manage access to resources and operations

Role-Based Access Control (RBAC) is a flexible process that simplifies the tasks of assigning users and developers the access rights to resources. Instead of assigning each individual user specific rights, the administrator creates roles which can then be given to a group of users. This is useful in organizations with many users to manage and a present need to manage and control API usage.

https://en.wikipedia.org/wiki/Role-based_access_control

https://cloudify.co/2016/04/15/simple-secure-role-based-access-control-rest-api-rbac-server-devops-cloud-orchestration.html

---

☑ Prevent API parameter tampering, attacks, and hijacks

Tampering enables the reverse engineering of the API, such that it exposes data or becomes vulnerable to DDoS attacks. Protecting them ensures that your web, cloud, and mobile applications are secure and safe. Monitor the APIs, infrastructure, and external services to detect and prevent DDoS attacks.

https://www.programmableweb.com/news/understanding-api-connectivity-to-resolve-app-ddos-attacks/analysis/2017/10/13

https://www.sqreen.com/

PROTECTION

☑ Use security best practices and tools

Observe the standard security best practices. Reduce your attack surface (harden the infrastructure and services), encrypt your data and communications channels, and filter and block bad traffic and malware. Don't forget to perform regular security audits, logging and analyzing your events and assets.

https://www.securitymagazine.com/articles/89283-ways-to-reduce-your-attack-surface

https://www.secureworldexpo.com/industry-news/best-practices-encrypting-data

www.sqreen.com/resources/pentest-checklist

☑ Detect and block unusual behavior

Monitor your application in production to detect and block unusual behavior, including account takeovers, and suspicious actors. This helps to prevent attacks from your user base.

https://www.sqreen.com/

https://www.scmagazineuk.com/4-reasons-why-behaviour-based-indicators-of-compromise-enhance-security/article/702283/

☑ Automate security testing and protection

Perform automatic security scanning for vulnerabilities in the code, infrastructure, and applications in staging. Use a security solution that can detect and block attacks in production in real time, such as SQL injections, NoSQL injections, and XSS. Ensure that your solution limits false positives and doesn't block benign traffic.

https://www.sqreen.com/

https://phoenixnap.com/blog/devsecops-best-practices-automated-security-testing

---

☑ Automate data policy management

Use an automated policy enforcement to manage the data lifecycle and flow. Create audit logs before and after any security issue. Address all the audit and compliance issues you uncover.

https://www.enterpriseready.io/features/audit-log/

https://thenewstack.io/why-every-company-needs-a-data-policy/

☑ Automate security tasks and practices

Use existing DevOps tools to automate some security functions. For example;

* Chef – to automate security testing
* Puppet – test compliance and enforce security policies
* Ansible – to define and automate security best practices such as applying custom policies, configuring firewall rules, locking out certain users, etc.
* SaltStack – to automate security practices

Combine common tools with a continuous security monitoring platform.

https://techcrunch.com/2018/02/20/chef-inspec-2-0-wants-to-help-companies-automate-security-compliance-in-cloud-apps/

https://www.slideshare.net/petems/compliance-and-auditing-with-puppet

---

☑ Complement automatic testing with creative manual tests

Automatic testing scripts may fail to recognize or identify visual issues that a human eye can pick up. In addition, a human tester will interact with the software and discover if there are usability or interface issues. Another challenge is when the automated tests scripts contain errors or bugs that give false negatives or positives.

https://testlio.com/blog/manual-qa-testing-best-practices/

https://blog.testlodge.com/why-automated-testing-will-never-replace-manual-testing/

☑ Follow post-production protection best practices

Automate scanning and collect application-level metrics upon deployment. You can use a tool such as Chef to automate the configuration management as well as the provisioning of the runtime environment. Use runtime protection solutions to monitor and protect your applications in production.

https://searchsecurity.techtarget.com/tip/Getting-runtime-application-self-protection-launched

https://www.sqreen.com/

☑ Limit your attack surface

Integrate protection and detection measures in the architecture to limit your attack surface, and reduce exposure and your internal and external threats. Focus on high-risk areas, such as web forms, internet-facing code, access control, session management codes, data from external sources, and other entry points that interface with external networks.

https://www.owasp.org/index.php/Attack_Surface_Analysis_Cheat_Sheet

https://www.securitymagazine.com/articles/89283-ways-to-reduce-your-attack-surface

---

☑ Use security tools that continue to evolve

The security solutions you put in place must keep pace with changing application environments and infrastructure, as well as with your own growth. These should have the ability to protect your system in real time and automatically send alerts when security issues arise.

https://www.sqreen.com/

EMPLOYEE BEHAVIOR

☑ Encourage secure employee behavior

Implement data protection program that combines security best practices and user education. Create awareness for employees towards improving personal security and preventing attacks like spear-phishing incidences. Always update and patch operating systems and application software, preferably automatically.

https://sudo.pagerduty.com/

☑ Check employee security behavior

Simulate a malicious attack in a controlled way to identify and fix real-world vulnerabilities. Use on-premise attacks to test desktop security and visitor controls. Use red teaming or pentests to identify vulnerabilities and their impact on businesses and employees.

https://www.computerweekly.com/news/2240241483/How-to-use-red-teaming-to-find-real-world-vulnerabilities

https://www.sqreen.com/checklists/pentest-checklist

☑ Do a spear-phishing campaign

Perform a spear-phishing campaign to test employees' behaviors and responses. You can also try hacking your employees in a controlled manner to assess and address internal risky behavior and preparedness.

https://searchsecurity.techtarget.com/definition/spear-phishing

https://searchsecurity.techtarget.com/tip/How-to-prevent-phishing-attacks-User-awareness-and-training

NOTES

sqreen

Trusted by security teams, loved by developers.

Monitoring and protection platform made to be incredibly powerful, yet very easy to use.

Unmatched security insights
Get access to more detailed security analytics than ever before, including app-level incidents you can act on immediately.

Instant protection
Out-of-the-box modules protect apps against a broad array of threats, with multiple layers of protection. Setup takes minutes, no config required.

Remediate as a team
Developers, DevOps, and Security can see for themselves what's gone wrong, and prioritize together to get it right.

Start your free trial at www.sqreen.com

Want this handbook as a PDF?
Scan the QR-code, or go to:
https://www.sqreen.com/checklists/devsecops-security-checklist