Cheatography

Gartner Guide to Successful DevSecOps Cheat Sheet

by David Pollack (Davidpol) via cheatography.com/2754/cs/14417/

Introduction

The analysts have taken lessons learned from the organization and its clients, and released 10 steps they believes will set businesses on a successful DevSecOps path.

Source: https://sdtimes.com/developers/gartners-guide-to-successful-devsecops/

1. Adapt your security testing tools

"Adapt your security testing tools and processes to the developers, not the other way around:"

According to the analysts, the Sec in DevSecOps should be silent. That means the security team needs to change their processes and tools to be integrated into DevOps, instead of trying to enforce their old processes be adopted.

2. Quit trying to eliminate all vulnerabilities

"Quit trying to eliminate all vulnerabilities during development."

"Perfect security is impossible. Zero risk is impossible. We must bring continuous risk- and trust-based assessment and prioritization of application vulnerabilities to DevSecOps," Head and MacDonald wrote in their report. DevSecOps should be thought of as a continuous improvement process, meaning security can go beyond development and can be searching and protecting against vulnerabilities even after services are deployed into production..

3. Focus first on identifying and removing

"Focus first on identifying and removing the known critical vulnerabilities."

Instead of wasting time trying to break a system, find focus on known security issues from pre built components, libraries, containers and frameworks; and protect against those before they are put into production.

4. Don't expect to use traditional DAST/SAST

"Don't expect to use traditional DAST/SAST without changes."

Scan custom code for unknown vulnerabilities by integrating testing into the IDE, providing autonomous scans that don't require a security expert, reducing false positives, and delivering results into a bug tracking system or development dashboard

5. Train all developers on the basics

"Train all developers on the basics of secure coding, but don't expect them to become security experts."

5. Train all developers on the basics (cont)

Training all developers on the basis of security issues will help prevent them from creating harmful scenarios. Developers should be expected to know simple threat modeling scenarios, how to think like a hacker, and know not to put secrets like cryptographic keys and passwords into the code, according to Head.

6. Adopt a security champion model

"Adopt a security champion model and implement a simple security requirements gathering tool."

A security champion is someone who can effectively lead the security community of practice, stay up to date with maturity issues, and evangelize, communicate and market what to do with security and how to adapt.

7. Eliminate using known vulnerable components

"Eliminate the use of known vulnerable components at the source."

"As previously stated, most risk in modern application assembly comes from the use of known vulnerable components, libraries and frameworks. Rather than wait until an application is assembled to scan and identify these known vulnerabilities, why not address this issue at its source by warning developers not to download and use these known vulnerable components," Head and MacDonald wrote.

8. Secure and apply operational discipline to

"Secure and apply operational discipline to automation scripts."

"Treat automation code, scripts, recipes, formation scripts and other such infrastructure and platform artifacts as valuable source code with specific additional risk. Therefore, use source-code-type controls including audit, protection, digital signatures, change control and version control to protect all such infrastructure and platform artifacts," according to the report.

9. Implement strong version control

"Implement strong version control on all code and components."

Be able to capture every change from what was changed, when the change happened and who made the change

10. Adopt an immutable infrastructure

"Adopt an immutable infrastructure mindset." **

Teams should work towards a place where all the infrastructure is only updated by the tools. This is a sign that the team is maturing, and it provides a more secure way to maintain applications, according to Head.

By **David Pollack** (Davidpol)
cheatography.com/davidpol/

Published 8th February, 2018.
Last updated 8th February, 2018.