Why Supply Chains Get Hacked

Without automated security checking in the DevOps cycle, it's easy for busy developers to accidentally introduce malicious code, amiss regulations/governance or security violations in their code. Here are five data points that show the impact of the problem and how to prevent it in your own software.

Unpatched Vulnerabilities

OVER 16,000

The number of new vulnerabilities reported to and analyzed by The National Institute of Standards and Technology NIST so far in 2023 (through July - https://nvd.nist.gov/general/nvd-dashboard).

2548

of them are rated Critical, most of them exploitable over the network, requiring no privileges or user interaction, and with low attack complexity.

The solution to this problem is persistent scanning of all source code and libraries by a system with access to a thorough database of documented vulnerabilities.

Exposed Secrets

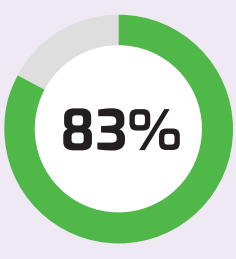
1 BILLION

The number of personal records exposed in the largest data breach in history, caused by an exposed secret. (https://securityboulevard.com/2022/07/the-worlds-largest-breach-root-cause-secrets-in-code/) The code, posted on a blog site on the Chinese Software Developer Networks (CSDN), "contained access credentials in plain text for all to see."

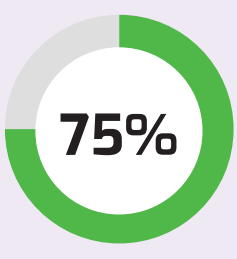
The records, for sale on the dark web, include "name, address, national id, mobile, police and medical records from one Asian country."

Secrets, carelessly left visible by developers, are a serious problem, leading to the compromise of confidential company data, of assets like source code, and even compromise of the build system itself.

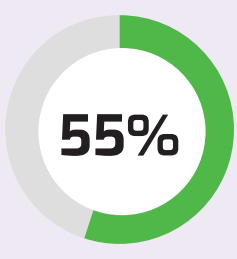
Disturbing statistics related to this problem are easy to find. Based on a study of data from Unit 42, their threat intelligence team, Palo Alto Networks found that:



of the organizations have hardcoded credentials in their source control management systems.



of organizations have VM instances with non-HTTP ports exposed to the public internet.



of organizations don't enforce AWS EBS volume encryption.

https://start.paloaltonetworks.com/unit-42-cloud-threat-report-volume-7

The methods by which developers mistakenly leave secrets exposed are known, and an intelligent scanning tool can analyze the source code for these problems, flagging them before they expose the company.

Productivity from DevSecOps

$6.7 MILLION

The value in increased operational efficiency from automated vulnerability and compliance workflows on open source software, according to a study by Forrester Consulting on behalf of JFrog, of "The Total Economic Impact of the JFrog Platform." (https://jfrog.com/tei/)

Forrester found that automation of these processes led to accelerated software delivery, increased operational efficiency, increased productivity, and reduced cloud engineering management costs.

In conventional DevOps systems, productivity is impeded by separate processes to perform checks for vulnerability and compliance issues. Automation of these checks increases the productivity of the whole development cycle by letting developers keep doing their jobs.

The return on investment was **393%**

The Cost of No DevSecOps

$4.45 MILLION

The average total cost of a data breach in 2023, according to IBM's Cost of a Data Breach 2023 Report (https://www.ibm.com/reports/data-breach).

They dug deeper, measuring 27 factors for their impact, positive and negative, on the cost of the data breach.

The largest savings

($249,278)

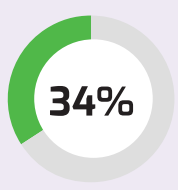
were from a DevSecOps approach.

Then they looked into the impact of those breach targets with the highest as opposed to lowest levels of DevSecOps adoption.

Savings for those with the highest degree of DevSecOps were

$1.68M

or



of those with the least.

Whenever you see a data breach it means someone has made a mistake, and software security errors are at the top of the list of those mistakes. Whatever you can do to increase the effectiveness of development security will make those data breaches less likely.

Curated Open Source

9,074,000

The number of versions created in the 12 months prior to July 3, 2023 in npm, the popular package manager for open-source JavaScript code, used by developers the world over. (https://blog.sandworm.dev/state-of-npm-2023-the-overview#heading-packages-created-per-month.

That's just the new ones. There are a total of 3.3 million packages available on npm, which doesn't provide a lot of help for developers to decide which are secure or even do what they claim properly.

The solution is to give developers a curated source of libraries they can use, containing code that has been vetted for security issues. Even so, the DevOps cycle should scan all code for known and unknown vulnerabilities. This not only saves time on research but also prevents a great deal of time and money being wasted on recovering from vulnerabilities imported into the code base. Also, make sure that a list of imports is current for inspection by the CSO or other responsible officer.

References

- **National Vulnerability Database Dashboard:** https://nvd.nist.gov/general/nvd-dashboard
- **The World's Largest Breach. Root cause: Secrets in Code (Security Boulevard):** https://securityboulevard.com/2022/07/the-worlds-largest-breach-root-cause-secrets-in-code/
- **Palo Alto Networks Unit 42 Cloud Threat Report, Volume 7:** https://start.paloaltonetworks.com/unit-42-cloud-threat-report-volume-7
- **Forrester Consulting Report - Total Economic ImpactTM of the JFrog Software Supply Chain Platform:** https://jfrog.com/tei/
- **IBM - Cost of a Data Breach Report 2023:** https://www.ibm.com/reports/data-breach
- **State of Npm 2023 - Packages Created Per Month (Sandworm blog):** https://blog.sandworm.dev/state-of-npm-2023-the-overview#heading-packages-created-per-month

DARKReading

Sponsored By JFrog