InformationWeek

TOP TIPS

DEFENDING YOUR SOFTWARE SUPPLY CHAIN

Experience has taught us all that every system everywhere is either under attack or at least could be. Software applications are one of the primary attack vectors for security breaches. Mitigating these risks requires software engineering teams to integrate security into the SDLC by adopting end-to-end developer-centric application security tools. Nearly all enterprises are working on improving their security posture and hardening it with automation to discover known and unknown problems.

But developers get impatient with security overhead. Anything that gets in the way of doing their jobs is a problem. You need real DevSecOps, which means automated security processes embedded into the DevOps pipeline at every step.

Software companies these days use a LOT of open-source software, as much as 90% in modern applications. This has made the software supply chain complex. Companies like SolarWinds that have had their own supply chains corrupted have ended up corrupting their customers and have faced severe costs financially, reputationally, and perhaps legally.

2021's most famous and widespread software supply chain vulnerability was Log4shell, a series of critical vulnerabilities in the extremely popular Apache Log4j logging service. But there have been many significant attacks since then:

- June 2023 – MOVEit Supply Chain Attack
- March 2023 – 3CX Supply Chain Attack
- February 2023 – Applied Materials Supply Chain Attack
- December 2022 – PyTorch Framework Supply Chain Attack
- December 2022 – Fantasy Wiper Supply Chain Attack

They keep on coming, and they are successful because it is hard to defend against them. But you can make yourself a lot safer if you follow certain best practices. Here are the 10 best measures you can take to protect your own development supply chain:

1. Curate Your Open-Source Packages Before They Enter Your Organization

The good news is that there are many quality and secure tools and libraries available for whatever programming task you need. The bad news is that there are many times that number of options that are low quality and/or insecure. Save your developers and customers a lot of grief by using a platform or service that creates curated repositories of trusted open-source programming options for you.

2. Implement a Shift-Left Strategy

The DevOps movement persuaded developers to test repeatedly during development, with test results feeding back into the development process. DevSecOps involves doing the same with security, scanning for problems as thoroughly as possible in each iteration of the process. Both find problems before they become severe and costly.

3. Don't Just Scan for Known Vulnerabilities

Many of the most common vulnerabilities are not going to show up in a vulnerability database but are programming errors in your own code. Many result from insecure practices, such as storage of credentials in code, insecure configurations of cloud services, and inadequate use of cryptography for network use. Intelligent services are available that can look for these sorts of errors and find them before there are consequences.

4. Don't Just Scan Your Own Code

It's a mistake to assume that even well-known and trusted libraries your programmers import are secure. Critical vulnerabilities are found in these libraries all the time. Log4J is a good example of a library that was trusted by many, and then suddenly, it was untrustworthy. If you always check, you will find out quickly.

5. Create a Good SBOM

The concern over software supply chain security has led to new standards for a Software Bill of Materials (SBOM). The SBOM allows a customer to scrutinize the components of a software product and see for themselves if any of them are problematic. President Biden ordered in 2021 that all projects in the government and in businesses that work with the government require an SBOM from suppliers. With a net that large, soon, everyone will need to create them.

6. Automate Security to Accelerate Development

Developers don't want to spend time scrutinizing libraries and tools, and they are generally not expert in it. Their job is to write code, and they want to get on with it. Nor do they want to wait for security people to do their jobs. The more security measures you can build into the automated pipeline, the more security will get done, and the more problems will be discovered at an early stage.

7. Balance Security with Compliance

Many of the security measures you should take as part of the development pipeline correspond to monitoring necessary for regulatory compliance and governance. Use the opportunity to create a record of your compliance.

8. Implement Policies Globally and By Project

Different teams may well have different security requirements, so you should be able to apply different policies to them. At the same time, you have your own standards that you should be able to apply to all projects. You should be able to make such changes without reprogramming the DevOps cycle.

9. Partner With Strong Application Security Vendors

Your security can't be agile if your vendors aren't. The vendors need to have up-to-the-minute intelligence on the threat landscape and bring that information to bear in your protection. They should provide you with all the information you need and be available if you need to consult.

10. Choose a Platform With a View Towards the Big and Little Pictures

One system for managing the security of your development cycle can give you visibility into every individual input as well as the big picture of the system. It can tell you details of every input to every version you have built. You may be able to put a pipeline like this together from parts, but it won't likely integrate as well, nor will you have that one throat to choke.

1. Log4Shell repository (National Cyber Security Center, Netherlands/NCSC-NL) - https://github.com/NCSC-NL/log4shell
2. Apache Log4j™ 2 - https://logging.apache.org/log4j/2.x/
3. Ransomware Reaches New Heights (Dark Reading) - https://www.darkreading.com/threat-intelligence/ransomware-reaches-new-heights
4. Alert - Supply Chain Attack Against 3CXDesktopApp (Cybersecurity and Infrastructure Security Agency) - https://www.cisa.gov/news-events/alerts/2023/03/30/supply-chain-attack-against-3cxdesktopapp
5. Chip company loses $250m after ransomware hits supply chain (Malwarebytes Labs blog) - https://www.malwarebytes.com/blog/news/2023/02/chip-company-loses-250m-after-ransomware-hits-supply-chain
6. Compromised PyTorch-nightly dependency chain between December 25th and December 30th, 2022 (PyTorch.org blog) - https://pytorch.org/blog/compromised-nightly-dependency/
7. Fantasy – a new Agrius wiper deployed through a supply-chain attack (welivesecurity bog/Eset) - https://www.welivesecurity.com/2022/12/07/fantasy-new-agrius-wiper-supply-chain-attack/