oxsecurity

5 Musts for Building a Software Supply Chain Security Strategy

Software Supply Chains Are Under Attack

Over recent years, the software supply chain has become a major target for cyberattacks. Notable hacks include Equifax and CCleaner in 2017, SolarWinds and Codecov in 2020, and CircleCI and Progress Software in 2023.

The software supply chain has become a very lucrative target for attackers because they only have to succeed with a single attack vector for a single company. Once they do, they can then exploit the system's assets to access the data and systems of the target's many customers.

"By 2025, 45% of organizations worldwide will have experienced attacks on their software supply chains."

- Gartner Research

The Solution is Clear, but Progress is Lacking

In response, many software development organizations are advising to put the following safeguards in place, each with increasing levels of protection:

LVL 6 - Handling new attack vectors, which we see being invented every week.

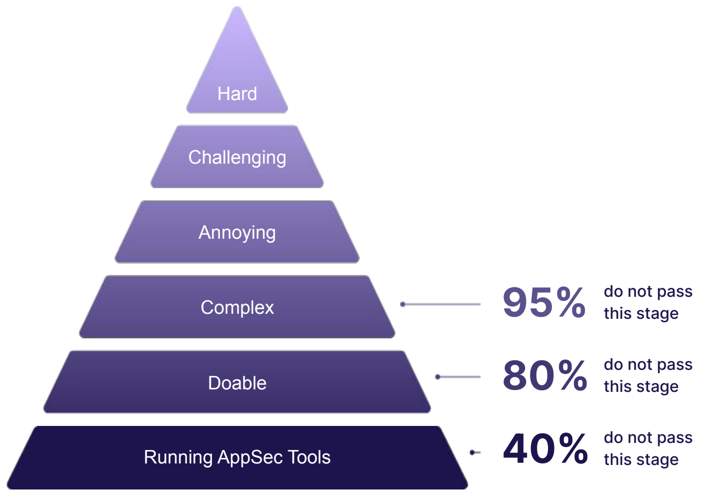
LVL 5 - Implementing a shift-left program, making R&D accountable for securing applications.

LVL 4 - Securing shadow development and pipelines that exist outside of R&D.

LVL 3 - Triaging the signal-to-noise ratio by prioritizing the thousands of risks that arise during development.

LVL 2 - Integrating AppSec tools with the CI/CD pipelines to make security a repetitive motion.

LVL 1 - Running application security tools.



The path to protection is clear, but most companies (80%) fail to progress beyond level 2. As a result, organizations accrue massive technical security debt, and continue to release vulnerabilities into production

This ebook provides you with 5 proven strategies for achieving software supply chain security completeness.

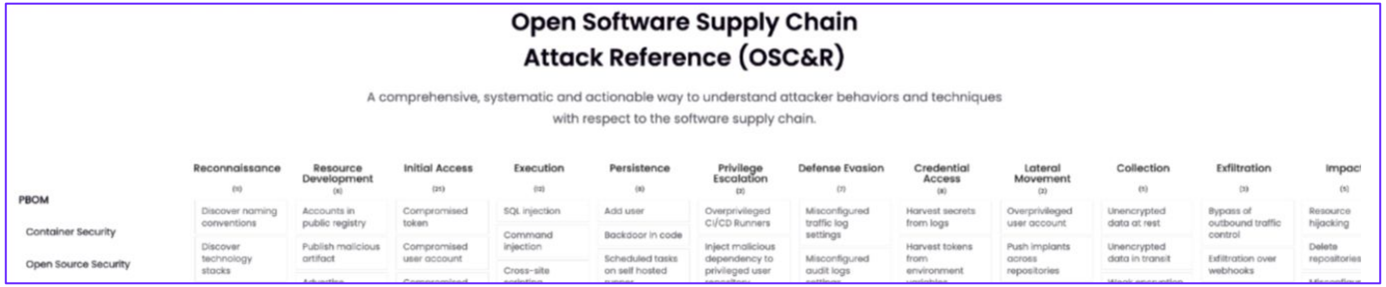
oxsecurity

Strategy 1

Use OSC&R to Understand the Underlying Risks

The first step in securing your software supply chain is to understand the underlying risks. You need to know how attackers find and exploit vulnerabilities that lead them to your or your customers' data. These paths, from recon to exploited asset are called attack kill chains.

Attack kill chains on endpoints are famously documented in the Mitre ATT&CK matrix. There is a similar framework, the Open Software Supply Chain Attack Reference (OSC&R), which documents the vulnerabilities that attackers exploit at every stage of the software development lifecycle.



Open Software Supply Chain Attack Reference (OSC&R)

A comprehensive, systematic and actionable way to understand attacker behaviors and techniques with respect to the software supply chain.

PBOM

Container Security

Open Source Security

Reconnaissance (11)	Resource Development (8)	Initial Access (21)	Execution (12)	Persistence (8)	Privilege Escalation (2)	Defense Evasion (7)	Credential Access (8)	Lateral Movement (2)	Collection (5)	Exfiltration (3)	Impac (5)
Discover naming conventions	Accounts in public registry	Compromised token	SQL injection	Add user	Overprivileged CI/CD Runners	Misconfigured traffic log settings	Harvest secrets from logs	Overprivileged user account	Unencrypted data at rest	Bypass of outbound traffic control	Resource hijacking
Discover technology stacks	Publish malicious artifact	Compromised user account	Command injection	Backdoor in code	Inject malicious dependency to privileged user	Misconfigured audit logs	Harvest tokens from environment	Push implants across repositories	Unencrypted data in transit	Exfiltration over webhooks	Delete repositories
			Cross-site	Scheduled tasks on self hosted							

Led by OX Security, the OSC&R framework was created by cybersecurity leaders from across the industry, and it clarifies the methods attackers use to reach and exploit your systems and cause damage.

"OSC&R helps security teams build their security strategy with confidence by giving them a single point of reference to proactively assess their own strategies for securing their software supply chains and to compare solutions."

- Hiroki Suezawa, Gitlab

Strategy 1

Using OSC&R to understand the risks to your software supply chain helps you do three important things:

Assess the tool coverage of your software supply chain.

Do you have tools and processes in place to detect and remove these threats?

Define which threats need to be prioritized.

See the difference between active kill chains that must be broken ASAP versus lower-priority hygiene problems.

Track current attack trends.

Focus on protecting against the most prevalent threats.

Join the OSC&R community to stay current on software supply chain threats:

Visit https://pbom.dev to learn more about the OSC&R model.

Join the OSC&R community Slack to stay current on OSC&R, new threats, and attacks.

oxsecurity

Strategy 2

Adopt a Continuous Approach

Securing your dev pipeline is as important as securing the software you deliver. A continuous approach is end-to-end, from planning through deployment, and it ensures no one can intervene during the process and do something malicious to your system.

Here are three components to a continuous approach:

1

Adopt security tools at each phase of the pipeline.

Everything that you're doing needs to be secure. From the planning, coding, staging, and verification, through release, configuration and operations. This ensures your pipeline is “secure by default,” and your security tools will indicate new risks as they arise throughout the pipeline so you can resolve them.

2

Reduce development and production attack surface.

Protect the development and production environments from attackers by implementing security tools and processes that reduce the attack surface These include vulnerability scans, governing and auditing access to development and production, and continuous risk assessments.

3

Know origins and provenance of everything in production.

This is true for internal code as well as external code like modules and open source. Knowing the origin and provenance of everything running in production helps ensure that no one has intervened to insert malicious code.

oxsecurity

Strategy 3

Reduce Developer Friction

To succeed with a continuous approach to AppSec throughout every stage of software delivery, it must be done in a way that does not increase complexity or friction for developers. They are focused on completing their engineering tasks, and it's important to avoid security becoming a distraction or worse, ignored.

Here are some recommendations to help you avoid developer friction:

1

Create a shared security experience for engineering and security teams.

You have lots of security tools generating issues. It's important to build trust with developers by integrating issues into a single, shared experience for engineering and security. Avoid silos, and instead enable everyone to see the same issues and issue details, regardless of the system that generated them.

2

Work within developers' existing workflows, don't create new ones.

Security tools, access controls, governance and other software supply chain protections should be integrated in a way that developers find easy and seamless. Flooding their inboxes with email alerts, raising popups in their IDEs is rarely a good idea. Find ways to work within the existing workflows like ticketing and CI/CD systems.

3

Get issues to developers in a rationalized, timely way.

Get findings to the right person in a streamlined way and assign issues to developers with details such as why the issue matters and how to fix it. And assign issues as they arise; developers can fix the issues quickly while they're still focused on the problematic code, versus weeks afterwards when it will take them longer because they have to relearn the code.

OX Security was named a Gartner® Cool Vendor for its innovative ways to:

- *Orchestrate security workflows*
- *Enforce consistent security policies as part of the SDLC*
- *Gain visibility and traceability into software delivery pipelines*
- *Benchmark applications using security risk scores*

Gartner
COOL VENDOR 2023™

Read the Cool Vendor report

oxsecurity

Strategy 4

Automate, Automate, Automate

According to Gartner, more than half of software engineering leaders are "shifting left," and taking direct responsibility for application security. This means that engineers are not only tasked with creating clean code and fast code, but now also secure code.

AppSec has an opportunity to make shift-left successful by giving engineers the right tools and the right information at the right time to deliver secure code. You have to flag all critical vulnerabilities for them to fix, but not waste their time or delay releases fixing non-critical issues. With thousands of issues being raised during a software development cycle, it's very challenging to get this balance correct. It is essential that you put automation in place to help shift-left programs succeed.

1

Automate integration

As mentioned, it is important to unify security risk data from your various tools into a single, coherent dashboard for engineering and security to share. Commercial DevSecOps dashboard solutions are a better choice than developing a custom dashboard in house. Harmonizing risk data from across all your tools is tedious and error prone and will take months to develop and troubleshoot. By contrast, commercial solutions solve the need immediately.

2

Automate prioritization

Avoid the low signal-to-noise trap. It's not uncommon to see thousands of risks raised by an organization's software supply chain security tools. That's a lot of alert noise; so much in fact, that developers often tune them all out. At this scale, you require automation, based on a risk framework like OSC&R, to deduplicate risks, separate critical issues from hygiene issues and help engineering focus on breaking live attack kill chains before addressing other issues.

3

Automate remediation

Finally, you should define standard policies for remediating issues, and then automate the execution of those policies. Opening tickets, creating PRs, blocking code merges and other actions can be automated to help ensure that identified risks do not end up in production systems, and that they are addressed in a timely manner that engineering and security agree on.

oxsecurity

Strategy 5

Continuously Control DevSecOps Security Debt

DevSecOps is often so focused on securing the development happening today in R&D, that they do not have the resources available to address the security debt that exists in legacy systems or in shadow development projects occurring outside of R&D...and there's usually a lot of it!

The good news is that the end-to-end tooling, OSC&R threat framework, and automation discussed in strategies one through four can help. They make securing legacy and shadow development pipelines possible without having to hire many more resources.

With a manageable, automation-driven approach as a template, it's worth taking the time to secure all of your software supply chains–new, legacy, and shadow. Risks can be raised and remediated quickly and thoroughly following the same approach and with the same tooling.

And remember that today's new development will be tomorrow's legacy system. Applying these strategies to all systems ensures that new threats can be identified across the enterprise, and not just in active development projects.

oxsecurity

Next Steps - Software Supply Chain Security Checklist

Now that you have the strategies, it's time to execute. Here's a summary checklist of next steps to secure your software supply chain.

1. Understand the Risks

- ☐ Visit pbom.dev to learn about the OSC&R framework
- ☐ Join the OSC&R community on Slack

2. Adopt a Continuous Approach

- ☐ Map your existing tooling coverage to the threats and SDLC stages in the OSC&R framework
- ☐ Acquire new tools to cover any gaps found in your OSC&R mapping
- ☐ Reduce your attack surface area

3. Remove Developer Friction

- ☐ Start a free OX Security trial to integrate your AppSec tooling into a unified risk dashboard
- ☐ Connect OX Security to your CI/CD infrastructure to integrate with developer processes
- ☐ Review risks with engineering to show threat intelligence OX provides to help with remediation

4. Automate, Automate, Automate

- ☐ Review OX prioritization of your software supply chain security risks with engineering
- ☐ Create an remediation workflow in OX to automate a recurring remediation task

5. Control DevSecOps Security Debt

- ☐ Create an inventory of your legacy and shadow development pipelines
- ☐ Connect them to OX Security to reveal and remediate risks

If you would like to discuss any of these strategies, feel free to contact us at contact@ox.security.

oxsecurity

Prevent risks across your software supply chain

Visit ox.security to try OX Security for free

Get Started >>

OX gives security and DevSecOps teams full visibility over the software supply chain attack surface, source code, pipeline, artifacts, container images, runtime assets, and applications. OX connects to an organization's code repository and performs a scan of the environment from code to cloud and then provides remediation strategies based on a list of prioritized risks and recommendations, taking into consideration the context and business objectives. OX further reduces exposure during the building stages, minimizing the attack surface without impeding developer agility.