carahsoft

Software Supply Chain Security

Best Practices For Cloud-Native Application Development

Thank you for downloading this white paper. Carahsoft represents proven DevSecOps solutions, delivering agencies the innovative solutions needed for every phase of the DevOps and DevSecOps lifecycles and with security built-in every step of the way. These solutions provide support for collaborative planning, rapid code builds, iterative testing, rapid release, optimized deployment and ongoing monitoring that continuously feeds into the next wave of planning.

Carahsoft combines extensive knowledge of the technologies we provide with a thorough understanding of the government procurement process to analyze needs, provide configuration support, simplify the ordering process, and offer special government pricing. Speak to a Carahsoft representative today about achieving your DevSecOps objectives.

For more information, contact Carahsoft or our reseller partners:
carah.io/solve/devsecops |devsecops@carahsoft.com | (703) 871-8570

anchore

SOFTWARE SUPPLY CHAIN SECURITY

BEST PRACTICES FOR CLOUD-NATIVE APPLICATION DEVELOPMENT

CONTENTS

Executive Summary

The compromise of the SolarWinds software supply chain has sent shock waves through business and government. No organization can afford to sit still as they increasingly depend on software supply chains to support the development of cloud-native applications that serve their internal and external customers.

Software supply chain attacks will remain a mounting concern for business and government agencies for the foreseeable future. However, there are actions you can take as a software supply chain participant to improve your security infrastructure, collaboration, communication, and processes and reduce risks for your organization and its customers. The time to start on these actions is now.

Introduction

Software supply chain attacks strike at the most vulnerable segment of software, the build process. It's where software vendors, system integrators, or internal developers bring together contributions and integrations from multiple sources. A typical list of software supply chain providers includes open source software (OSS) projects, subcontractors, and third-party software vendor partners.

The shift to cloud-native applications has only exacerbated this issue as use of containers has exploded. By using containers, developers can leverage software components from outside their organization more easily, speeding development but increasing the level of risk and the due diligence required to ensure secure and compliant software. Organizations must enhance their development processes to deliver the highest levels of security for their cloud-native applications.

Developers Play a Large Role in Security

"Today 28% of developer respondents said that they are completely responsible for security in their organizations while 41% said they were '*responsible, but part of a bigger team*'."

— Mapping the DevSecOps Landscape 2020 Survey, GitLab

The advice in this white paper applies to all participants in the software supply chain who need to improve their toolchains, collaboration, and security practices to better protect from attack. Participants in the software supply chain include:

- Software supply chain owners: organizations who are responsible for building, and delivering a software component or application for either internal or external customers
- Software supply chain partners: organizations that supply software components for downstream organizations to use.
- Software vendors may act in both roles. They are an "owner" of their own supply chain to ensure they produce secure products while also acting as a "partner" in the supply chain of their customers.

Software Supply Chain Security Risks

The technical and operational complexity of a software supply chain takes security risks to a new level. The supply chain owner's DevSecOps teams need to work with their business and technology stakeholders to improve internal processes while also extending collaboration, communications, and security best practices out to their supply chain partners.

First, let's look at a common software supply chain that includes both external systems that are run by partners and other third parties such as commercial vendors and OSS projects from outside the enterprise that integrates open source, third-party, and internally developed software. The historically open, collaborative nature of software development has helped improve development efficiency. Unfortunately, this has led to one of the most pervasive operating principles: assume your suppliers are doing the right thing. The software supply chain security model makes it challenging to "trust but verify" so as a supply chain owner it's even more important to ask for more information about the software while improving collaboration and communications up and down the software supply chain.

There are two significant aspects to consider when securing your software supply chain.

- **Securing the software workloads**

  Cloud-native applications are built using both internally developed and externally sourced software components and dependencies. Security must be ensured for all of these components.

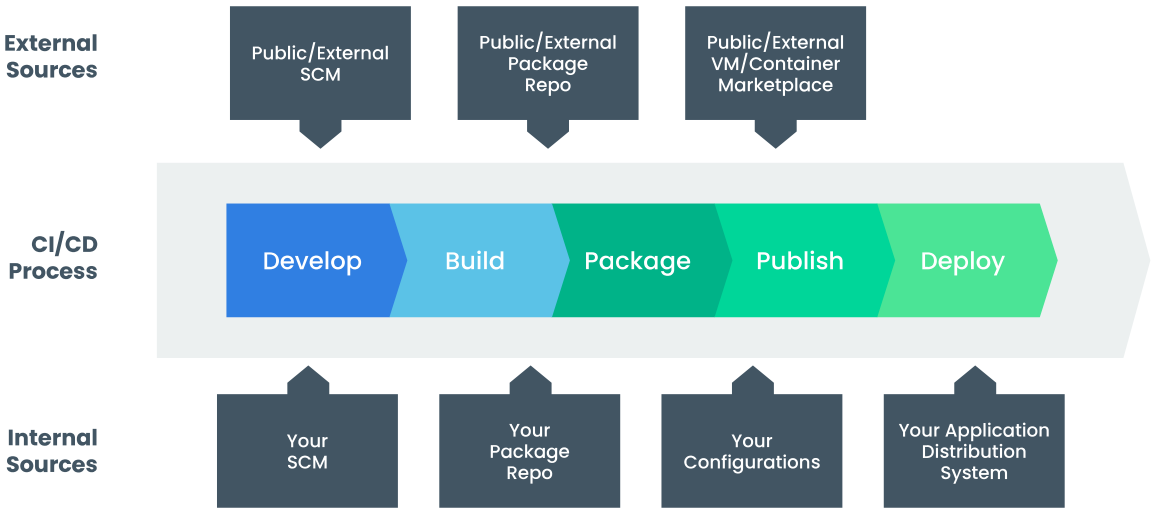
- **Securing development toolchains**

  Cloud-native applications are constructed using a number of software development tools, from source code management (SCM) to continuous integration and delivery (CI/CD) tools to container repositories. These tools may be SaaS applications or run internally, but in either case it's critical that these toolchains are secured in order to prevent internal or external actors from attacking the development toolchain and ultimately compromising the software that is being produced.

There are a myriad of attacks against both of these aspects of the software supply chain, making it critical that organizations consider and address all of the relevant threat vectors.

Breaking Down the Software Supply Chain

The following graphic shows the various external systems that impact your supply chain such as the public software configuration management (SCM) solution containing third party OSS code; public/external package repository such as a third party library/binary; and the public virtual machine (VM)/container marketplace. These external systems may span multiple partners across multiple time zones.



A continuous integration/continuous development CI/CD toolchain includes the following artifacts and components that vulnerable to attack:

- **App Source Code**
- **App Binary**
- **Executable**
- **Shipped Application**

The following actions in the CI/CD toolchain may also be subject to attack:

- **Sourcing of public packages**
- **Build**
- **Package**
- **Publish**

Internal and external systems and tools that interact with your software supply chain include:

- **Software Code Management (SCM) platform** that houses the App Source Code
- **Public package repositories** used to source components
- **Private package repositories** housing the app library/binary for the project
- **Your Configuration Repositories** housing your app or infrastructure configuration
- **Application distribution system** where your shipped application and updates reside

Your internal systems such as your CI/CD toolchains also face compromise by insider threats that can run the range from an employee or contractor making a mistake such as a misconfiguration that opens up your supply chain to attack to a disaffected employee or contractor can also attack the supply chain using their accounts and credentials to bypass external security measures. Such attacks can affect your organization's managed infrastructure, your SaaS applications, and even your cloud services provider (CSP).

There are different types of consumers for applications produced by software supply chains:

- **An independent software vendor (ISV)** such as SolarWinds, Microsoft, or Cisco, counts their customers as their consumers. Regardless of whether an ISV is producing a shippable or SaaS application, one attack on their software supply chain has the potential to affect all their customers downstream.

- **A software supply chain in a large enterprise** such as a Fortune 500 company may only serve internal customers such as their business units. In other cases, such as with large financial institutions, some applications may serve external customers or partners.

- **Government agencies** own software supply chains that may serve customers inside the government, such as in a shared services environment where government agencies share applications or cloud services. In other cases, the applications could serve outside customers such as constituents applying for government grant funding online.

Software Supply Chain Breaches: A New Attack Vector

We've become too accustomed to the news of data breaches in our news alerts, headlines, and evening news. People outside of the technology industry are even dulling to the "breach of the week" news. One thing all these past incidents have had in common was the target was in one industry. For example, Capital One — a bank noted for being cloud-first — had a data breach in 2019 that compromised the data of over 100 million of their customers.

The recent SolarWinds compromise represents a new level of attack with criticality well above what we unfortunately now seem to tolerate as just part of the daily news. Recent reports say the SolarWinds Orion IT monitoring platform has over 30,000 customers, including such industry luminaries as Microsoft. The Orion platform customer base also includes the United States Department of Defense (DoD), Federal Bureau of Investigation (FBI), and most major federal government agencies.

The recent HAFNIUM breach targeting Microsoft Exchange Servers is also reported to have impacted at least 60,000 customers in the US.

Current State of Software Supply Chain Security

The SolarWinds and HAFNIUM breaches show we're entering a new era of cyber attacks. While industry and government cybersecurity teams face new onslaughts of attacks every day, a software supply chain attack takes emerging threats to the next level. Conventional cybersecurity strategies can't counter an attack against an organization's software supply chain.

Here are two examples of why the current state of software supply chain security is coming up short:

Software Updates Can Introduce Risk

Service desks have been advised on best practices for software updates since the dawn of IT. One piece of advice dictates that IT teams should update software to the latest version and only install signed versions of updates, patches, and software that can counter such an attack.

However, in the case of SolarWinds, customers received software that was signed, but compromised. In following this best practice, you just did just what the attacker wanted by installing the compromised software on your systems.

There's also the best practice of monitoring software behavior. Unfortunately, as the SolarWinds supply chain compromise shows, the attackers were stealthy and patient so that they could wreak damage on SolarWinds customers before discovery of the compromise by an external security vendor.

Source Code Reviews are Insufficient

The SolarWinds compromise shows source code reviews aren't a certain defense against a software supply chain attack either. Some reports point out that the attackers had control of the SolarWinds build environment, making it possible for them to insert malicious code without the knowledge of the SolarWinds Orion development team.

When traditional security practices and solutions such as these fail on such a grand scale, it becomes time to reevaluate how software supply chain security works in organizations of all types.

The SolarWinds hack exposes many of the major drawbacks that come with today's software supply chains to the light of new and changing cybersecurity realities.

Changing software supply chain security means mustering not only your own teams but also counterpart teams in all the commercial partners and vendors that touch your supply chain to become true partners with open lines of communication, collaboration, and knowledge sharing.

While larger corporations may vet the security of software vendors through questionnaires or independent assessments, more still needs to be done to reduce risks across the software supply chain. Work beyond that initial questionnaire and subsequent onboarding means focusing on automated vulnerability scans and other methods to shore up your process for bringing in software components or applications.

Security, development, and IT teams must collaborate to ensure sufficient security checks and remediation of issues at each stage of the software supply chain. Those compliance processes must apply to software from all sources, whether open source, commercial vendors, or internal developers.

7 Best Practices for Improving Software Supply Chain Security

There isn't a single security solution that can secure your software supply chain from attacks. The gravity of the SolarWinds attack is an invitation for you and your software supply chain partners to collaborate and reassess the security, governance, communications, and collaboration needs across your software supply chain. Here are some best practices you are bound to see and experience in the post-SolarWinds world:

1

Improve Relationships and Collaboration

There's a human angle to supply chain security that's easy to ignore. Relationships with your supply chain partners are integral to building trust, collaboration, and communications.

Put tools and processes in place that enable your DevSecOps team to collaborate with your software supply chain partners.

Options include a federated collaboration solution that enables you to communicate and collaborate with your vendors in real-time. It's also important to share threat intelligence and best practices amongst your supply chain participants.

"Like their counterparts in Ops, security, and test, a majority of developers (29%) think "soft skills," including communication and collaboration, are going to be most important for their future career development."

— Mapping the DevSecOps Landscape 2020 Survey, GitLab

2

Improve Governance of Software Onboarding

A major challenge in the software supply chain is that it's nearly impossible to have any practical visibility or control over your third-party software vendors' security and compliance practices or other partners in your supply chain. Thus it's imperative to have an onboarding process for software components or applications that includes collaboration, communication, and security best practices that all participants in the supply chain must follow.

Here's a typical onboarding scenario for a small company — such as a SaaS startup — that does business with a larger company such as a telecommunications company. The customer must sign a contract before they can purchase the software for their corporate enterprise. The small company vendor must self-certify that they comply with Sarbanes Oxley, NIST 590, or Cybersecurity Maturity Model Certification (CMMC) to complete the contract. Of course, the small company can answer a compliance question as "No." In that case, the prospective customer and the vendor set up a phone call where the customer's compliance security specialists walk through the compliance questionnaire. The call will focus on many security practices, including the vendors' usage of multi-factor authentication (MFA), access control lists for their corporate accounts, and other security-related information about their technology infrastructure.

When it comes to a commercial software supply chain, it's difficult to justify opening up visibility technologically across vendors in a software supply chain. Exposing system access and cybersecurity strategies produces a high level of risk that the average enterprise may choose to ignore or defer to another time. You have to look for ways to compensate for this shortcoming as both a software supply chain owner and a partner through collaboration, communication, and information exchange on threats, vulnerabilities, and other security-related issues. Open source and commercial container scanning tools can help close this gap in some scenarios.

The situation is often better in government agencies, where contractual and security regulations draw firmer lines that can give the government more control over their vendors' security and compliance. However, even the US federal government was far from immune from the SolarWinds compromise. Government contracts can include enforceable security requirements with financial penalties up to and including being barred from doing business with the government if your technology solution and infrastructure don't meet compliance requirements.

When your organization sets up a software supply chain, you can create a set of security best practices that serve as the baseline requirements for joining the software supply chain as an allowed software supply chain partner.

Security requirements for a software supply chain partner typically include:

- **Approved practices for code sourcing** from OSS projects and commercial vendors
- **Standardized manual and automated code reviews** at designated build stages
- **Consistent software security testing** in their development and production environments
- **Best practices for detecting anomalous insertions of code** or software components during a partner's software development lifecycle

Putting in such guidelines for supply chain partners can lead to mixed results with commercial vendors. However, systems integrators and other businesses working for the government are used to following stringent guidelines for doing business. The most obvious option to set down such guidelines is through your purchasing agreement or as part of a contractual addendum. The vendor would have to agree to follow those guidelines as a requirement for doing business with your organization.

Guidelines can become challenging at validation and enforcement time. There needs to be an enforceable penalty such as indemnification when a security issue occurs.

Onboarding OSS to your Supply Chain:

If OSS is part of your software supply chain, it's up to the supply chain owner to create a process to scan OSS code for vulnerabilities. The same rule applies if you're a supply chain partner using OSS components in your products. If the OSS comes from a vendor such as Red Hat, then you can onboard them as you would any other supply chain partner.

Software supply chain governance is a team sport. You need support and agreement from stakeholders across your organization, including executives, cybersecurity leadership, line-of-business managers, and others involved in the software delivery process that your software supply chain supports.

3

Harden your Build Environment

The DevOps and DevSecOps communities are paying more attention to hardening their DevOps toolchains against external attacks. The Linux Foundation recommends hardening of build environments.[1] It's now time for supply chain owners to extend this discussion to the entire software supply chain.

Treat all build systems up and down your software supply chain as the critical systems that they are. Apply the same or higher security requirements to your build systems as your production system. While the Linux Foundation admits it's not clear if such a step would've made a difference in the SolarWinds compromise specifically, omitting this step opens the door for malicious actors to achieve the same outcome by infiltrating your build system.

Regardless, it's time to take action to better secure your build environments across your software supply chain. While you have control over the build environments in your own corporate or agency domain, you will need to work with your supply chain partners to ensure their build environments meet your security standards.

4

Require an SBOM for all Partners and Vendors

The software bill of materials (SBOM) will play a key role in defending software supply chains in the future. Look for SBOMs to become a requirement at contract signing time for firms of all sizes aiming to join the software supply chain of a large enterprise.

OSS — a critical component of cloud-native software — doesn't normally ship with an SBOM. That leaves the enterprises and vendors using OSS to build critical components of enterprise software to generate their own SBOMs.

Typically, large company contracts require a written record of all the open source dependencies of their vendor's software. They'll also want to see a correlation of open source licenses to those dependencies. Large corporations call this open source license compliance. The question to ask is whether the company requesting the information is taking that data to create an SBOM and then run a vulnerability analysis.

The SBOM by itself isn't the end of the story. However it provides critical data that is needed to assist with your security efforts Software supply chain owners can use the SBOM provided by their supply chain partners to conduct their own vulnerability analysis and to gain a view into their risk as new threats emerge.

1. https://www.linuxfoundation.org/en/uncategorized/preventing-supply-chain-attacks-like-solarwinds/

5

Implement Defense in Depth

Defense in depth (DiD) is a security design concept where multiple security control layers are placed throughout a complex IT system.

Defense in depth provides redundancy through multiple layers of security so that you are still protected in the event of a security controls failure or an attacker exploits a vulnerability. For example, if you check your source code, your container image, and your running container for vulnerabilities, you're going to get the same report from all three checks. But if an attacker exploits just one of those software development artifacts or activities, you can miss it without a multi-layered approach. However, when you have multiple security controls with overlapping functionality at different levels or with a software supply chain at stages, you're resilient if one of them suffers a malfunction, exploit, or compromise.

In the future, major federal government agencies and large corporations will seek as much redundant security coverage for their software supply chains as possible, at or under budget to secure areas of their software supply chains that are susceptible to attack.

6

Apply Zero Trust to Software Supply Chain Security

Originating in network security, Zero Trust is the concept that organizations shouldn't trust anything inside or outside their networks, and instead should verify all systems before allowing access. Such principles are easily adaptable to the larger security issues that the software supply chain poses.

Both public and private sector cybersecurity experts are identifying the need to apply zero trust principles to software supply chain security.

Leading IT analyst firm Gartner advises that "Longer term, acknowledging that anyone can be breached and that there is no inside vs outside of the network (i.e., zero trust), security teams should adapt their security and risk management roadmaps to better reflect supply chain attack exposure."

William Evanina, former director of the National Counterintelligence and Security Center, pushed for more scrutiny by the federal government over the software supply chain. He has said publicly that the federal government must put a supply chain risk mitigation program in place that takes advantage of zero trust.[2]

You can expect more news about Zero Trust adoption in many industries in the aftermath of SolarWinds.

2. https://www.nextgov.com/cybersecurity/2021/01/counter-intelligence-chief-calls-zero-trust-software-supply-chain-policy/171355/

7

Create a “Kill Chain” for your Software Supply Chain

The news of recent software supply chain attacks needs to spark innovation to defend against the next such attack. One innovation sure to play a growing role in software supply chain security is the cyber kill chain or just kill chain, which creates as many opportunities as possible to prevent, disrupt, or at least quickly detect such incidents before weaponized software causes damages.[3] A cyber kill chain traces the stages of a cyberattack from the earliest reconnaissance through data exfiltration. The kill chain helps enterprises target and combat ransomware, security breaches, and advanced persistent attacks (APTs).

The concept of a cyber kill chain comes out of the DoD and is based on risk management practices such as National Institute for Technology (NIST) Cyber Supply Chain Risk Management (C-SCRM) guidance.[4]

The concept of the cyber kill chain continues to evolve since its inception to detect and counter insider threats, social engineering, and more emerging threats making it more than ideal for software supply chain security.

3. https://www.darkreading.com/omdia/solarwinds-hack-lessons-learned-finding-the-next-supply-chain-attack/a/d-id/1339871
4. https://csrc.nist.gov/projects/cyber-supply-chain-risk-management

Next Steps

The gravity of supply chain security takes a major effort between the software supply chain owner and their software supply chain partners to level up their security, threat intelligence, and situational awareness to mitigate risks. Next steps to combat software supply chain attacks may vary depending on the size of your organization and its security budget.

For the Software Supply Chain Owner

Here are some potential next steps for you to improve your own supply chain security:

- Take steps to improve collaboration and communication amongst your internal teams that work with your software supply chain vendors.
- Account for open source and commercial software as part of your onboarding, security, and ongoing maintenance processes.
- Review your software supply governance judiciously and an going forward.
- Build upon collaboration and communication improvements to help ensure that your software supply chain partners are aware of security and operations processes and compliance.
- Harden your build environment and create documented standards to guide your partners through hardening their build environments.
- Implement defense in depth if you are a large organization with the budget, staffing, and business case to justify putting in redundancies to secure the major systems serving your supply chain.
- Apply a zero trust framework across your software supply chain that treats your supply chain as if it has already been breached.
- Create a "kill chain" for your software supply chain that creates as many opportunities as possible to disrupt attacks on your software supply chain.

For Software Supply Chain Partners

Here are some potential next steps to help improve supply chain security if your organization is a software supplier:

- Look for ways to improve collaboration and communications with the supply chain owner. Remember that personal relationships can be important to build trust with your customer (the supply chain owner). Open communication channels promote information and threat sharing, a necessity.
- Make security part of your corporate culture from day 1 putting tools, processes, and training in place. Open your security processes to internal and external auditors regularly.
- Harden your build environments and even consider and look to the supply chain owner for any guidance or standards you can follow as part of your contract or through collaboration and knowledge sharing.
- Take a continuous improvement approach to software supply chain security when it comes to toolchain hardening, strategies, communications, and collaboration.

About Anchore

Anchore accelerates the development of secure and compliant cloud-native applications. Our suite of container security solutions seamlessly embeds in the DevOps lifecycle with continuous security and compliance checks early in the software development process. From sourcing to CI/CD pipelines to production, Anchore's solutions protect the software supply chain and prevent container security risks from reaching production. Using Anchore as part of the DevSecOps toolchain creates a reliable way to detect issues earlier, save developers time and lower the cost to fix vulnerabilities. Built with an open source foundation, Anchore solutions provide transparency into source code and the benefit of peer reviews.

Headquartered in California with offices in Virginia and the UK, Anchore customers include large enterprises and government agencies that require secure and compliant cloud-native applications. To learn more about Anchore's solutions, visit **www.Anchore.com**.

anchore

info@anchore.com

anchore.com