argon
an aqua company

2021 Software Supply Chain Security Report

A Turning Point Year for Software Supply Chain Security

In 2021, the world woke up to a surge in an attack vector that had been a security risk for many years; one that the security community could no longer neglect: software supply chain attacks. Following the SolarWinds Sunburst attack in late 2020, software companies of all sizes, across all industries, began facing an augmented number of targeted and organized supply chain attacks. This enhanced threat resulted in significant system downtime, monetary loss, and reputational damage of businesses worldwide.

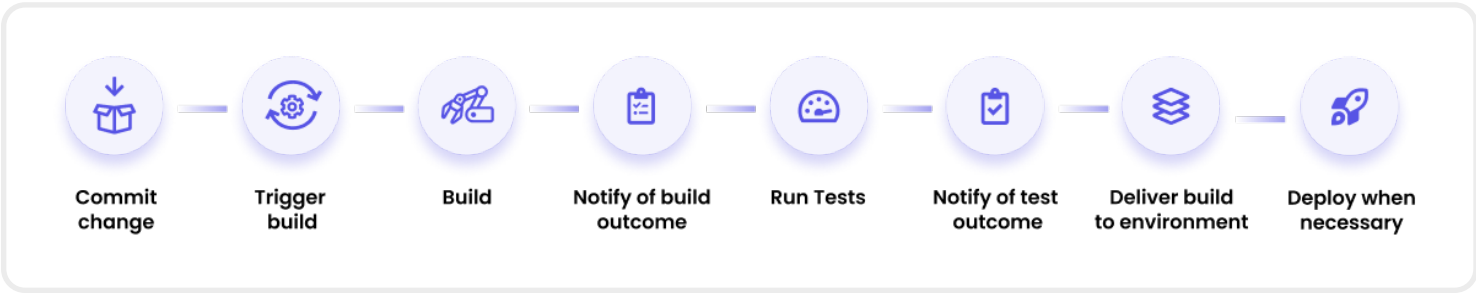

Figure 1: The software supply chain process

When Did Random Attacks Become a Pattern?

Throughout 2021, supply chain attacks increased in number and sophistication. This represents a notable shift in attackers' approach, now focusing their efforts on breaching software suppliers. This allows them to leverage paths that are implicitly trusted, yet less secure, and to establish a way to breach many victims with one attack, by proxy.

> **The high risk of software supply chains is, in part, attributed to the fact that a successful attack may impact a large number of companies who make use of the affected supplier's software.**

The SolarWinds Sunburst attack is a good example of the potential damage of software supply chain attacks. In this nation-state attack against the networking tools' vendor SolarWinds, about 18,000 of their customers were exposed as a consequence of using SolarWinds' breached software. As many as 250 of these exposed organizations suffered targeted attacks, including governmental agencies, such as the U.S. Pentagon, and top enterprises, such as Microsoft and FireEye.

Looking Back at 2021

The SolarWinds attack is considered one of the largest and most sophisticated supply chain attacks to-date and exemplifies the devastating potential of supply chain attacks. The attack directed attackers' attention to the software supply chain's comparatively low security status among organizations and was the opening shot for a wave of supply chain-based attacks that followed. The SolarWinds attack received a lot of media coverage and inspired a global wave of security awareness and improvement initiatives focused on reducing the risk of supply chain attacks.

In February that year, Alex Birsan tested the exposure of enterprises to a supply chain attack technique that leverages automated DevOps practices to compromise pipelines. This tactic, known as dependency confusion, results in malicious public libraries being incorporated into projects instead of the trusted private libraries of the same name. Birsan was able to hack into Apple, Microsoft, and dozens of other top companies during this experiment, illustrating that even companies that highly prioritize security can fall victim to implicit shortcomings in the software supply chain.

SolarWinds was followed by a similar build-time code-manipulation attack in which attackers penetrated the Codecov product 's software supply chain, manipulating the build process to inject malicious code into its software and using the software update mechanism to distribute the malware to Codecov customers.

Not long after, on May 12, 2021, President Biden's Executive Order on Improving the Nation's Cybersecurity was released, emphasizing – for the first time – the need to enhance software supply chain security.

In July 2021, the attack on Kaseya raised awareness of the immediate and downstream effects of supply chain attacks. In this attack, a managed service provider software was used to distribute the REvil ransomware to the managed service provider's customers, causing significant downtime and revenue loss.

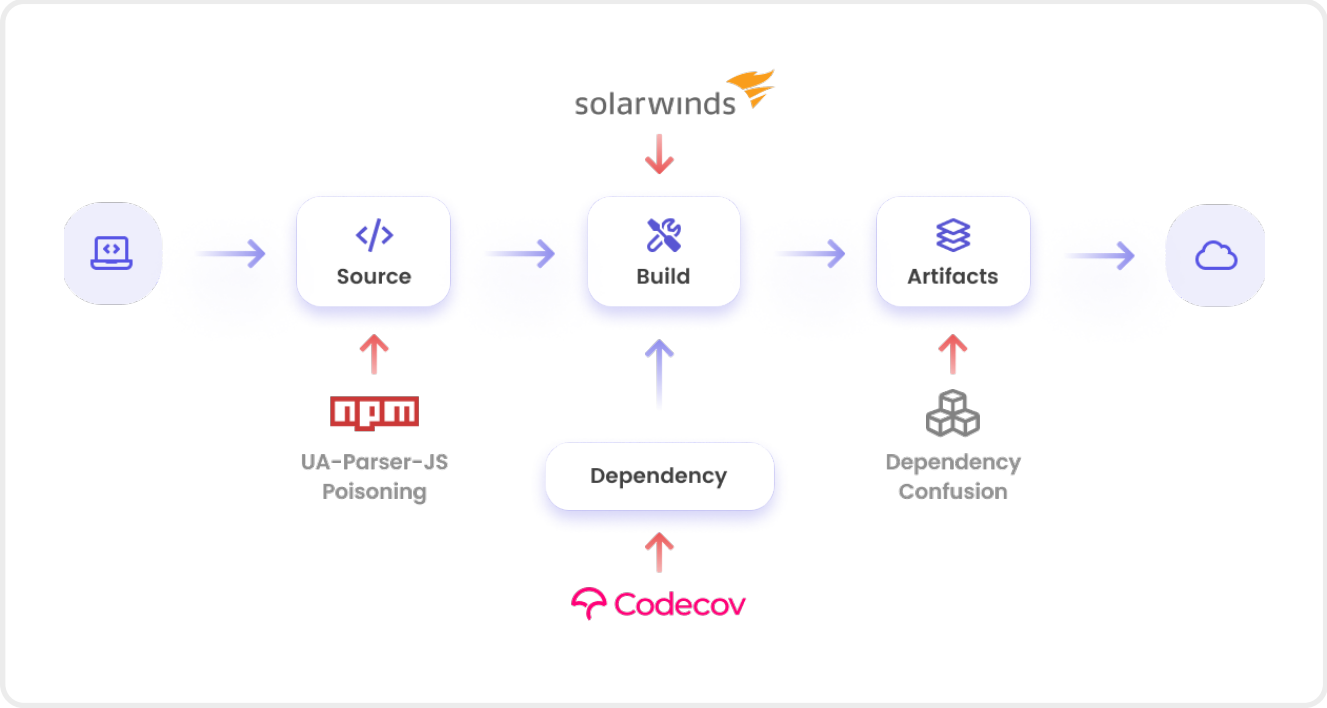

Figure 1: Visualizing where the biggest attacks compromise the software supply chain

Unfortunately, these examples are not isolated cases and the number of supply chain attacks has since steadily increased, with the most popular approach being software dependency poisoning. In November alone, we saw three attacks against highly popular packages (UA-Parser-JS, COA, and RC), each with millions of downloads per month. This malicious tactic has proven quite effective and further stresses the need for the security community to shift their attention to and address this highly damaging potential attack vector.

The year's final incident would come on December 9th, when the Log4Shell vulnerability was discovered and forced software vendors into a patching frenzy. Shortly after the discovery, attackers started to exploit this popular package and take advantage of this vulnerability to launch their attacks.

Main Lesson Learned from 2021 Attacks' Analysis

Examining the success rate and consequent damage of the many attacks in 2021, one of the most evident details is that current security tools and practices are not adequate for preventing software supply chain attacks. Traditional application security testing cannot detect supply chain attacks which, often, exploit trusted software artifacts rather than the vulnerabilities targeted by such tools. Additionally, established CI/CD and DevOps pipelines rely on implicit permissions to enable rapid commits and deployment, implementing security controls at the end of this process – far too late to preclude malicious activity.

For organizations to stay secure, there is an increasing need for new protective methods and solutions that are built to address the unique characteristics of supply chain attacks.

Supply Chain Attack Vectors Still Waiting for a Solution

The number and impact of the year's attacks highlight the fact that application security teams face a new challenge that will require innovative thinking. Most AppSec teams lack the resources, budget, and knowledge to sufficiently address the risk of supply chain attacks. This is further complicated by the need for cooperation from development and DevOps teams.

During the past six months, Argon's security experts analyzed dozens of customer security assessments to identify the state of enterprise security and to qualify its readiness to defend against software supply chain attacks.

Key Findings

This analysis discovered that the level of security across software development environments remains low, with all evaluated companies having vulnerabilities and misconfigurations that expose them to supply chain attacks. Below are the three main risks that each of these companies have in common, forming the recommended areas of focus for any organization that develops software:

1. Use of Vulnerable Packages: Open-source code is part of almost all commercial software. Many of the open-source packages in use have existing vulnerabilities, and the process of upgrading to a more secure version requires effort from development and DevOps teams. As a result, most companies are lagging in vulnerability remediation, even when it comes to high severity vulnerabilities. It is, therefore, not surprising that this is one of the fastest- growing methods of carrying out supply chain attacks.

Attacks leveraging vulnerable packages come in two flavors:

- Exploiting existing vulnerabilities: Exploiting packages' existing vulnerabilities to obtain access to the application and execute the attack. An example of this is the recent Log4j cyberattacks. Having vulnerable packages within your code can provide attackers with access to applications in production.

- Package poisoning: Planting malicious code in popular open-source packages, or private packages, to trick developers or automated pipeline tools into incorporating them into the application build process. An example of this is the ua-parser-js package poisoning.

2. Compromised Pipeline Tools: Taking advantage of privileged access, misconfigurations, and vulnerabilities in the CI/CD pipeline infrastructure (e.g., source code management system, build agent, package registries, and service dependencies) can provide attackers with access to critical IT infrastructure, development processes, source code, and other application artifacts.

A compromised CI/CD pipeline can expose source code, which is the blueprint of your application, your development infrastructure, and your processes. It enables attackers to change the code or inject malicious code during the build process and tamper with the application (as was the case of SolarWinds). This type of breach is hard to identify and can cause a lot of damage before it is detected and resolved. Attackers can also use compromised package registries to upload malicious artifacts instead of legitimate ones. In addition, there are dozens of external dependencies (e.g., services, tools) that are connected to the pipeline and can be used to gain access to the pipelines or to launch attacks (as was the case of Codecov).

In every development environment that was reviewed, Argon researchers identified critical misconfigurations and vulnerabilities in the pipeline tools that reduced the overall security posture of the environment.

3. Code/Artifact Integrity: One of the main risk areas identified in Argon's research is the upload of bad code to the source code repositories, which directly impacts the artifact quality and security posture. Common issues that were found in most customer environments are:

- Sensitive data in code (secrets),
- Code quality and security issues,
- Infrastructure-as-code (IaC) issues,
- Container image vulnerabilities, and misconfigurations.

In many cases the number of issues discovered were overwhelming and required dedicated cleanup projects to reduce exposure, such as secret cleaning, standardizing container images, and other activities.

According to ENISA, the European Union Agency for Cybersecurity, 66% of emerging supply chain attacks focus on the software supplier's code to further compromise targeted customers of that supplier.

The software supply chain is a core component of modern application development and leaving such a wide attack vector unprotected threatens to severely lower companies' security posture, potentially exposing sensitive data and creating additional entry points for attackers to the application in runtime. In many cases, security teams lack visibility into the risks that threaten the software supply chain, only to discover them when it is too late. Additionally, most companies do not have preventative capabilities within CI/CD tools and process to compensate for this lack of risk visibility.

Summary of 2021 Findings

Software supply chain attacks grew by 3X in 2021, as compared to 2020, with more vulnerabilities and attacks discovered every month. Attackers focused on open-source vulnerabilities, dependency poisoning, code issues, insecure software supply chain processes, or implicit trust in software suppliers to distribute malware or establish backdoors in the resources of unsuspecting application users.

Considering the number of issues found in company, AppSec teams are challenged by a lack of resources, visibility, and access, exacerbated by inadequate security tools. At this point, the attackers have the upper hand and, unless security measures are implemented to control risk and prevent supply chain attack vectors, enterprises are likely to fall victim to attacks such as those levied against SolarWinds, Kaseya, and Codecov.

Looking Ahead at 2022

The Log4j vulnerability discovered in December 2021 reminded everyone of the potential impact that single vulnerability in a popular package can have across the tech industry. Not surprisingly, shortly after the CVE was discovered, we started to see attackers exploiting this new attack path using bots to find it, inject their malicious code, and launch major attacks. Looking ahead to 2022, the massive effect of these attacks will continue to put software companies on attackers' priority list, and we should expect this trend to accelerate in the frequency and sophistication of software supply chain attacks.

The wave of software supply chain attacks and the massive damage they inflicted during 2021 did not go unnoticed by the board members, management teams, and security executives of enterprises worldwide. We are seeing growing awareness in board rooms and within the security community for preparations against supply chain attacks and, according to a recent survey, over 60% of security leaders say they plan to deploy software supply chain security measures in 2022.

There is also more support in the software and cloud communities with initiatives like Google SLSA and the Cloud Native Computing Foundation (CNCF) Supply Chain Security Forum that promote awareness and help set standards and guidelines for a secure SDLC.

Enterprise challenges in securing the software supply chain will remain high. The main challenges for AppSec teams in 2022, according to security leaders, revolve around:

- Lack of resources, with most AppSec teams being understaffed and overloaded without a way to close this gap.
- Collaboration and coordination with DevOps and development teams, which is required to implement effective security as part of the software development process.
- Gaps in supply chain security knowledge and expertise, which is required to enable contributors from development, DevOps, and security teams to define and implement a holistic security program over their development processes and CI/CD pipelines.
- Traditional security tools that are not designed to address the complexity and dynamic nature of the software supply chain, which provide limited risk visibility and inadequate security controls for the diverse CI/CD and DevOps tools used for modern applications.

Security teams and DevSecOps practitioners need to start working together, define and execute new security strategy and initiatives that account for risks inherent in the software supply chain. They must bolster the security of their development environments to better protect their application infrastructure, processes, and deployed software to be ready for the next wave of these advanced attacks.

Collaboration with DevOps teams and automation of security within development workflows should play a major part in these strategies. When paired with new security solutions that are designed to secure the software development process against sophisticated attacks, the result will be an elevated security posture suitable for cloud native applications. Without these measures and failing to enforce a dedicated security practice around the software supply chain that can prevent such attacks from proliferating environments, software vendors are risking their business, their reputation, and their customers' trust.

About the Author

Eran Orzel is Chief Customer and Revenue Officer and founding member of Argon security, the leader in software supply chain security. Prior to joining Argon, he held several roles at Check Point Software Technologies, most recently as the Global Head of Strategic Sales and Partnerships, where he led and played a significant role in the rapid growth of Check Point's major business growth engines. Eran is an experienced and innovative business leader with over 20 years of experience in sales leadership and go-to- market operational roles in cybersecurity and enterprise software.

About Argon

Argon enables security and DevOps teams to protect their software supply chain against vulnerabilities, security risks, and supply chain attacks. Argon's first-to-market security solution delivers visibility, security, and integrity to your CI/CD pipeline, ensuring trust in your software releases. At Argon, we believe that the fast release of code should not come at the expense of security. We are here to secure the new world of software development enabling our customers to build, test and deploy software securely.

For more information, visit: www.argon.io or contact us at info@argon.io