International Journal of Computer Trends and Technology
ISSN: 2231–2803 / https://doi.org/10.14445/22312803/IJCTT-V72I8P110

Volume 72 Issue 8, 70-79, August 2024
© 2024 Seventh Sense Research Group®

Original Article

Chain Reaction: Analyzing Trends and Crafting Defenses Against Software Supply Chain Attacks

Varadharaj Varadhan Krishnan

Independent Researcher, Washington USA.

Corresponding Author : varadharaj.krishnan@gmail.com

Received: 13 June 2024 Revised: 18 July 2024 Accepted: 08 August 2024 Published: 29 August 2024

***Abstract* - *This paper comprehensively analyses the software supply chain attack. Software supply chain attacks have increased in frequency and sophistication in recent years and have already caused widespread impact. This paper outlines the anatomy of such attacks, detailing various techniques used at different supply chain stages, from development to software distribution. The paper delves into notable incidents, including the SolarWinds attack and other significant breaches from 2020 to 2023, showing the widespread impact and TTPs and exploring the strategies that could have prevented or minimized the impact. The study uses open-source software supply chain security incident data sets to analyze trends and investigate the root cause and mitigation strategies. By performing thematic and empirical analysis of past incidents, this paper aims to produce critical actionable insights and equip organizations with the knowledge and strategies to mitigate and defend against these software supply chain attacks in the future.*

***Keywords* - *Cybersecurity, Software supply chain security. Supply chain attack, SSCA mitigation strategies, SSCA trend analysis open-source supply chain attack.*

1. Introduction

Organizations around the world face an accelerating threat from supply chain attacks on software. Software supply chain attacks involve compromising software updates, inserting malicious code into legitimate software packages, or exploiting third-party services and tools. These attacks target modern software development's interconnected, global nature, compromising one link of the software supply chain to reach numerous downstream organizations and consumer targets [1]. Software development today commonly involves multiple layers of contractors and providers all around the world, who may or may not fully appreciate the trust customers place in the software from them and in the work product customers consume downstream. In the past, software supply chain breaches were few and rare occasions. They were also usually executed by sophisticated attackers often linked to geopolitical adversaries. One of the most notable supply chain breaches in recent years was the SolarWinds attack, attributed to the Russian APT group named APT29, also known as "Cozy Bear" [2]. But in the last three years, almost two-thirds (61%) of U.S. businesses were affected by such an attack, with at least one of their key suppliers being hacked. These attacks have become a common and serious issue for organizations and businesses all around the world. The barrier for a successful software supply chain attack was further lowered in 2023 and has increased throughout 2023 and 2024. They are found across an array of popular open-source projects, most notably npm and PyPI. Open-source package repositories were a major pathway for software supply chain attacks in 2023, with a staggering 1,300% increase compared to 2020 [5][6][27]. In particular, the Python Package Index (PyPI) experienced a 400% rise in threat instances in just 2023 [10]. Now, the landscape of supply chain attacks has broadened, and both sophisticated nation-state actors and less resourceful beginner threat actors can perform such attacks through open-source projects, as seen in the Operation Brainleeches campaign targeting Microsoft 365 users. Federal efforts to raise the bar for software security are still in their infancy and remain confined to federal contractors. The burden of securing software supply chains falls to the private sector and individual software makers. This paper aims to help organizations understand the dimensions and landscape of software supply chain risk and provides a comprehensive strategy to build defenses against these pervasive threats.

2. Anatomy of Software Supply Chain Attack

Software supply chain attacks exploit software development and distribution vulnerabilities to compromise systems and gain unauthorized access [2]. These attacks occur at various stages within the supply chain, from the initial development of the software to its deployment and use. Below is an illustration of how a threat actor compromises the software source code to inject malicious code that gets built and distributed via the software vendor's legitimate software.

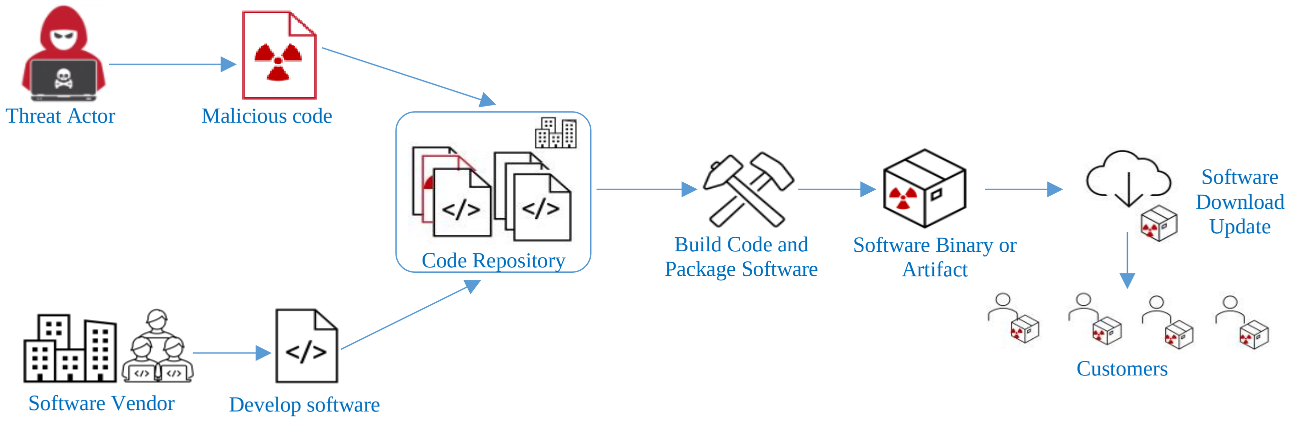

Fig. 1 Illustration of a software supply chain attack

The attack has two phases. The first phase is "Supplier Attack," which focuses on compromising one or more suppliers. The second phase, called the "Customer Attack," targets the final victims of the attack. Although linked by access to the supplier, these two phases differ significantly in techniques used, attack vectors exploited, and time spent on the attack [3][4]. While supply chain attacks are executed in quite a few different ways, most of them fall into one of three categories: target development, target deployment, or target usage. However, within each category are a handful of attack techniques.

2.1. Midstream Attacks

Attacks that take place during development; these focus on components that act as intermediaries in the software development lifecycle, like tools or build pipelines. An example is when Click Studios' Passwordstate software, an enterprise password manager, was compromised during development when an attacker injected bad code into its "in-place upgrade" feature, causing updates pushed to customers to deliver their payload to those customers' networks. [12][17].

2.2. Dependency Confusion Attacks

Dependency confusion attacks happen by exploiting the use of private, internally created software dependencies. They do this by registering a dependency with the same name as the one used internally and putting the malicious one with an incremented version number out on public software repositories.Then, the threat actor waits for it to be downloaded by an unsuspecting user [17]. The software build systems are likely to download the latest version, and the threat actor can inject malicious code successfully and then perform further attacks.

2.3. Compromise SSL and Code-Signing Certificates

Attacking the digital certificates used to establish the identities of computers and the software they run is another way threat actors, especially nation-state threat actors, carry out attacks. There are two types of digital certificates that are of interest to attackers: Secure Sockets Layer/Transport Layer Security (SSL/TLS) certificates, which are used to secure Web traffic, and code-signing certificates, which are used to ensure the software is not tampered with and can be traced back to a trusted vendor. Once stolen, threat actors will be able to sign their malicious payload with the same certificates [12][13][14].

2.4. CI/CD Pipeline Infrastructure Attacks

This technique aims to attack the continuous integration and delivery processes with the intention of embedding malicious payloads in the built artifact.

2.5. Open-Source Software Attacks

Open-source software attacks occur when threat actors insert harmful code into an open-source software package. This code then spreads to users who utilize the package. Because of the sheer larger number of open-source software users, such an attack can have significant real-world consequences [14][15][18]. The taxonomy, as presented in Figure 2, has one section for the supplier and one section for the customer. In the supplier table, the first column, "Attack Technique Used to Compromise the Supply Chain," shows how the supplies would be attacked. The second column, "Supplier Assets Targeted by the Supply Chain Attack," shows the supplier's target. For the customer table, the first column, "Attack Techniques Used to Compromise the Customer," shows how the customer would be attacked.

The second column, "Customer Assets Targeted by the Supply Chain Attack," shows what the target of the attack would be from the customer side [19][20][21]. Figure 3 illustrates the different attack vectors within a software supply chain from development to distribution. Subsequently, at the packaging stage, compromised packages or dependencies can be introduced. Finally, attackers might manipulate distribution channels during software distribution, affecting the software delivered to end-users. Each stage, from developer input to software distribution, presents critical points where robust security measures are essential to prevent compromises, underscored by the red hazard symbols indicating compromised dependencies.

SUPPLIER	
Attack Techniques Used to Compromise the Supply Chain	Supplier Assets Targeted by the Supply Chain Attack
Malware Infection	Pre-existing Software
Social Engineering	Software Libraries
Brute-Force Attack	Code
Exploiting Software Vulnerability	Configurations
Exploiting Configuration Vulnerability	Data
Open-Source Intelligence (OSINT)	Processes
	Hardware
	People
	Supplier

CUSTOMER	
Attack Techniques Used to Compromise the Supply Chain	Customer Assets Targeted by the Supply Chain Attack
Trusted Relationship [T1199]	Data
Drive-by Compromise [T1189]	Personal Data
Phishing [T1566]	Intellectual Property
Malware Infection	Software
Physical Attack or Modification	Processes
Counterfeiting	Bandwidth
	Financial
	People

Fig. 2 ENISA Taxonomy for supply chain attacks [19]

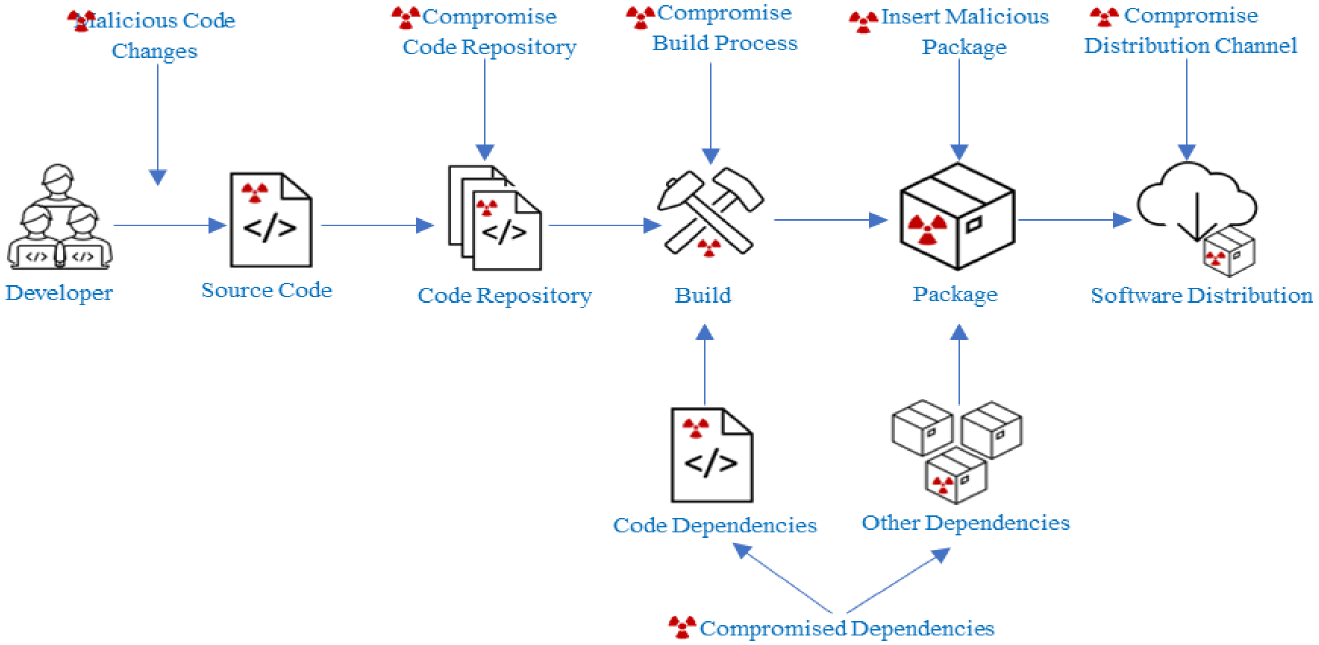

Fig. 3 Illustration of software supply chain threats

Table. 1 Software supply chain threats

Threat	Real World Attack - Example
Malicious Code Changes	SushiSwap [26]
Compromise Code Repository	Malicious NPM Packages [27]
Compromise Build Process	SolarWinds [28]
Compromised Dependencies	event-stream [29]
Insert Malicious Package	Codecov [30]
Compromise Distribution Channel	Attacking Azure Container Registries [31]

Table. 2 Notable attacks and their special characteristics

Date	Attack	Description
Dec 2023	Hackers Exploit JetBrains Vulnerability	In December 2023, the Russian Foreign Intelligence Service-backed group CozyBear exploited a critical vulnerability in JetBrains TeamCity servers. This sophisticated attack infiltrated the development environments of numerous organizations, potentially compromising the integrity of software produced and distributed using TeamCity. The breach demonstrated the significant risks posed by nation-state actors in software supply chains [3][5].
Nov 2023	Protestware on npm	In November 2023, ReversingLabs researchers identified npm packages containing scripts broadcasting peace messages related to conflicts in Ukraine, Israel, and the Gaza Strip. This protestware leveraged the popularity and widespread use of npm to disseminate political messages, highlighting the potential for open-source repositories to be used for purposes beyond traditional cyberattacks, including social and political statements [6][14]
Jul 2023	Operation Brainleeches	Operation Brainleeches, uncovered in July 2023, marked a significant "dual-use" campaign on npm. Over a dozen malicious packages were identified, targeting both application end users and supporting email phishing campaigns aimed at Microsoft 365 users. This attack demonstrated the evolving strategies of cybercriminals, combining supply chain attacks with other forms of cybercrime.
June 2023	MOVEit	In June 2023, the MOVEit file transfer management program by Progress Software was compromised, impacting over 600 organizations globally. The breach involved the exploitation of vulnerabilities within the MOVEit software, leading to unauthorized access and significant data breaches. This incident was considered one of the most severe supply chain attacks to date due to the scale and sensitivity of the data affected [14]
Mar 2023	3CX Desktop App	During the software build stage, the 3CX Desktop App, an enterprise voice over IP solution, was compromised in March 2023. Malicious code was inserted, leading to the distribution of a tainted version of the application to users. This attack illustrated the dangers of compromised build environments and the widespread reach such breaches can have on end users [3]
Apr 2021	CodeCov	In April 2021, attackers targeted CodeCov's Bash Uploader, a tool used for uploading code coverage reports. By compromising this tool, attackers gained access to sensitive environment variables from the CI/CD environments of CodeCov's customers, affecting hundreds of networks. The attack was particularly concerning due to its stealthy nature and the critical access it provided to attackers [4].
Dec 2020	SolarWinds	The SolarWinds attack, discovered in December 2020, remains one of the most infamous software supply chain attacks. Attackers believed to be backed by the Russian government, inserted malicious code into the Orion Network Management System software, distributed to around 18,000 customers. The breach affected numerous high-profile organizations, including U.S. government agencies, and showed the world the extensive reach and impact of supply chain compromises [12]
Sep 2017	CCleaner	In September 2017, hackers compromised the development and distribution systems of the popular CCleaner software managed by Piriform. The attackers inserted malicious code into legitimate versions of the software, which were downloaded by millions of users. The malware was designed to collect data from infected systems and potentially deliver a second-stage payload to specific targets. This attack demonstrated how even well-regarded and widely used software could become a vector for extensive cyber espionage [4]

3. Notable Attacks

In recent years, the number of software supply chain attacks has surged dramatically. In 2023, Almost two-thirds (61%) of U.S. businesses reported being directly impacted by such an attack. This growing threat is not confined to the United States; it is a global challenge. Table 2 shows the most notable attacks and their special characteristics.

4. Trends in Software Supply Chain Attacks

The software supply chain attack dataset provided by the Digital Forensic Research Lab (DFRLab) [25], a pioneering entity within the Atlantic Council, is used to perform trend analysis. The dataset shared by DFRLab has a record of software supply chain attacks since 2010. DFRLab is an entity of the Atlantic Council, renowned for its meticulous digital forensic research and investigation and documentation of instances of online misinformation, cyber threats, and state-sponsored information operations. Their datasets, derived from open-source intelligence (OSINT), are used in this research to analyze key patterns and trends of software supply chain attacks. A dataset titled SCRM-database-v3 [25] is used for this study. Figure 4 above shows a steady increase in the total number of software supply chain attacks over the last decade.

Further grouping the incidents by the distribution vector, a clear trend emerges. An increase in open-source dependency-based attacks has been on the rise since 2018. The increase in popularity of open-source projects and the very nature of the collaboration mode for OSS development paved the way for threat actors to use the OSS dependency vector. Figure 5 shows the distribution by vector since 2010. Further grouping the incidents by the type of code compromised on the supplier side, a similar trend emerged, showing the increase in the number of OSS projects that have been the target since 2016. Figure 6 shows the code type attacked by year, and Figure 8 shows a comparison of the codebase category. Both clearly show that open-source projects are being increasingly used to perform such attacks.

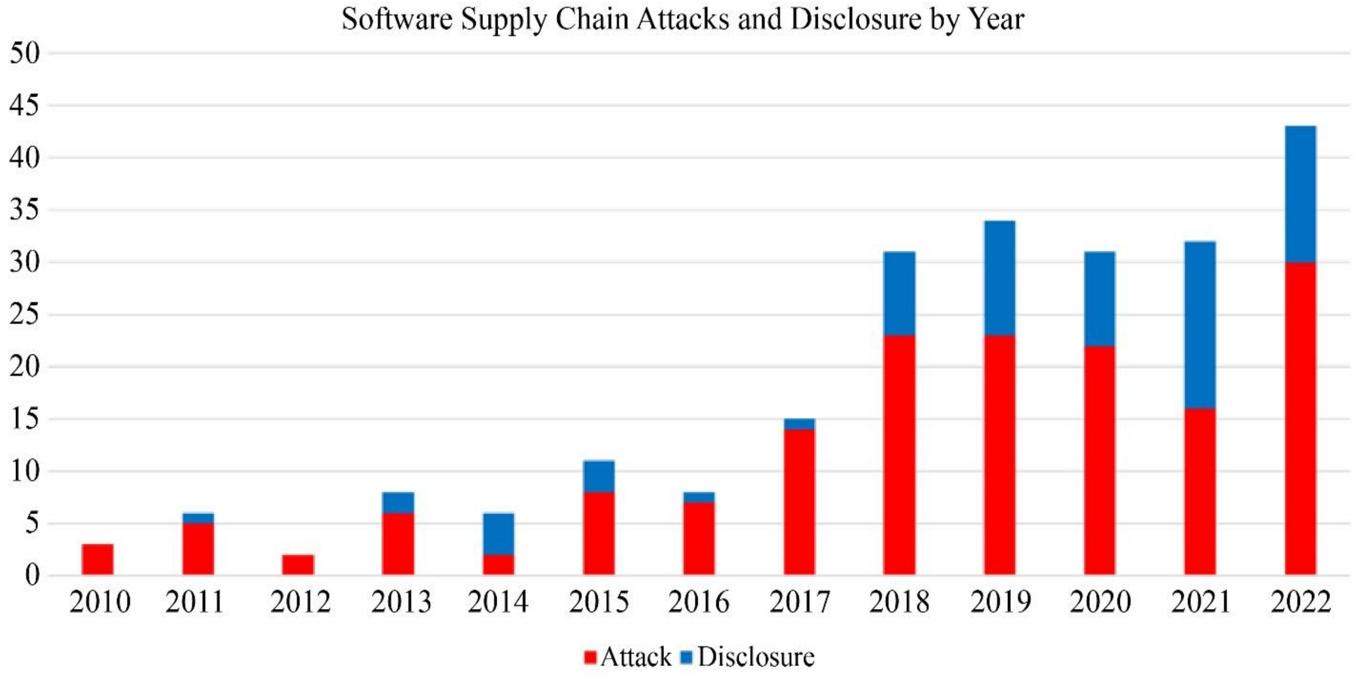

Fig. 4 Software supply chain attacks and disclosure by year

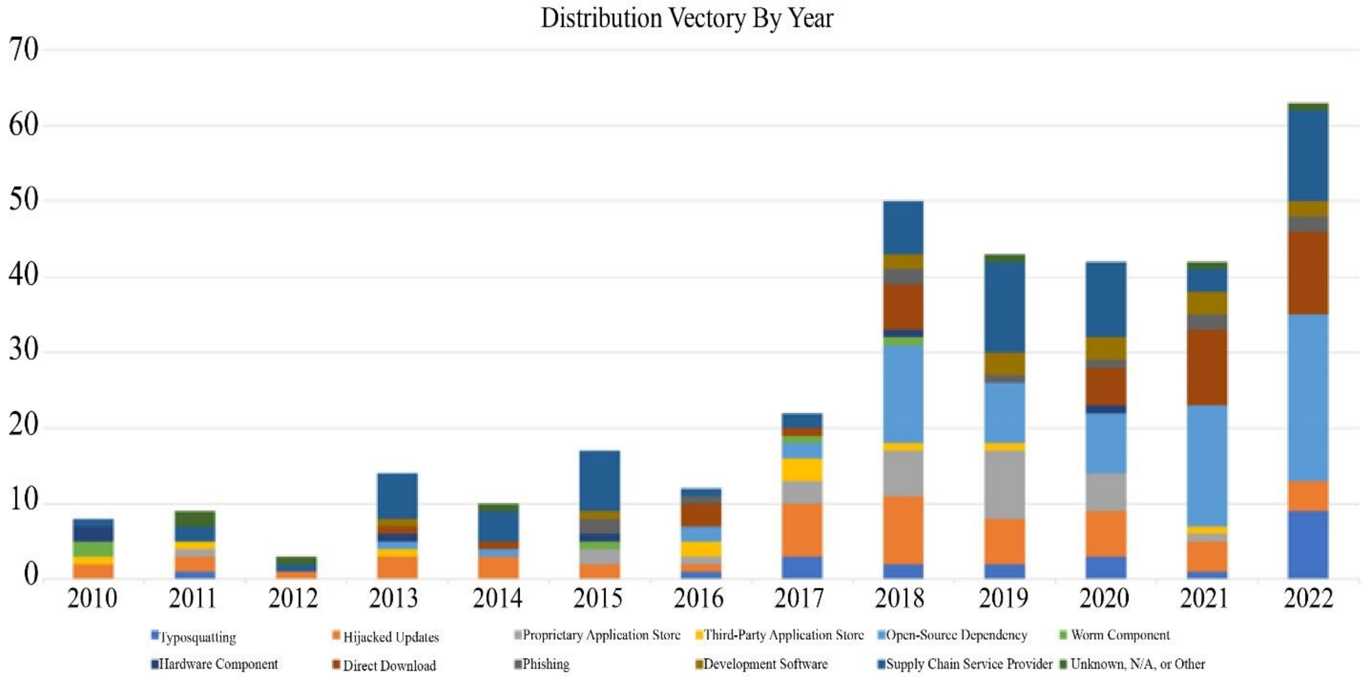

Fig. 5 Distribution vector by year

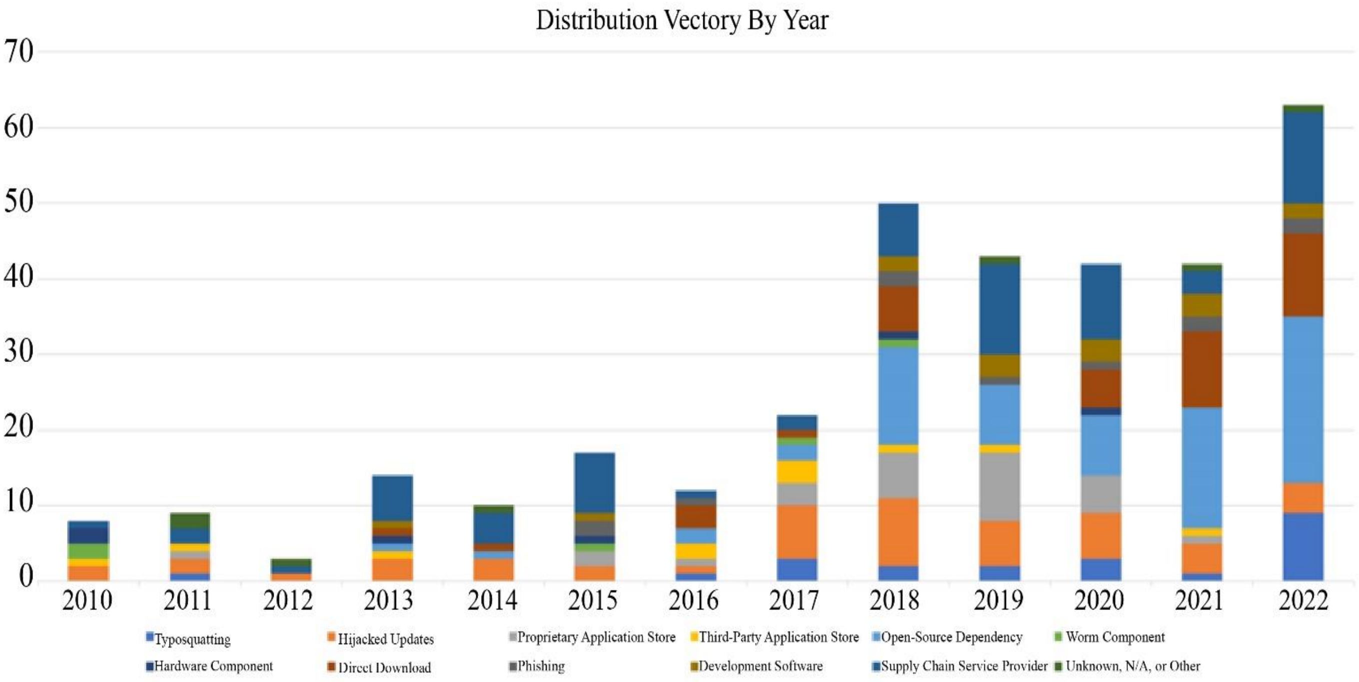

Fig. 6 Affected code types by year

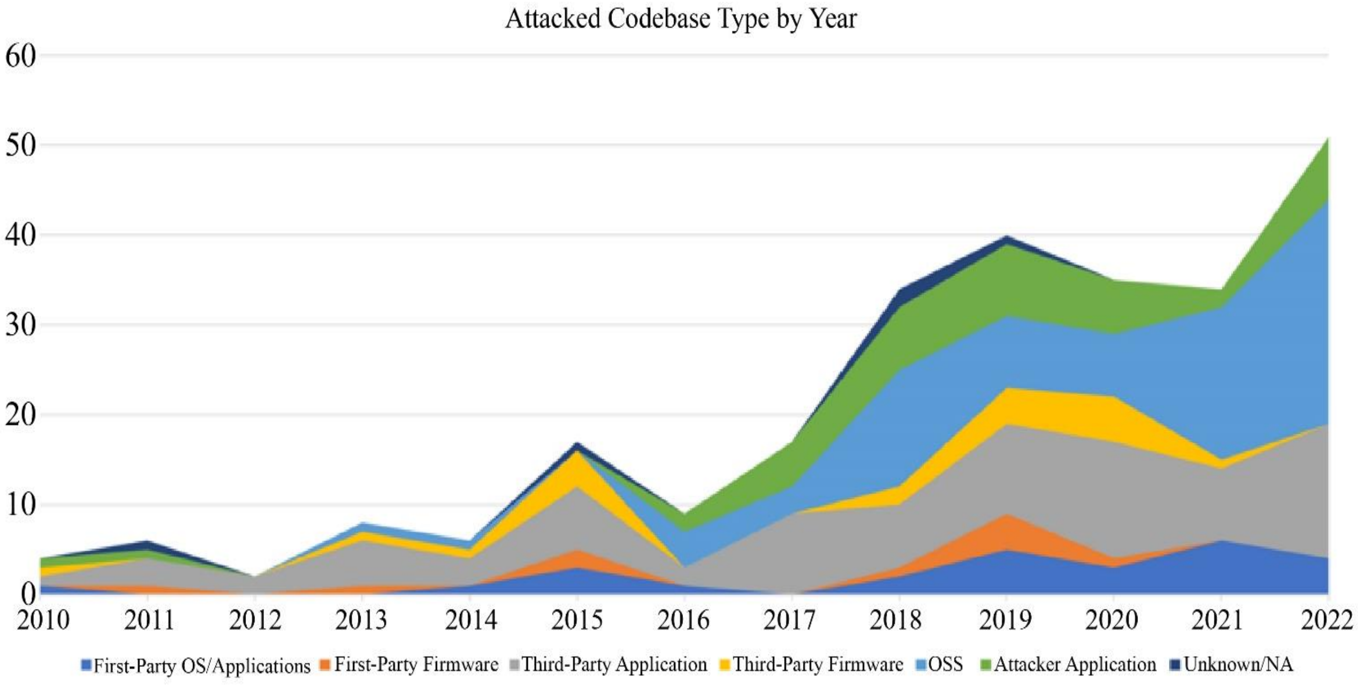

Fig. 7 Attacked codebase type by year

5. Increased Targeting of Open-Source Software

Recently, open-source projects have become increasingly lucrative targets for threat actors. The increased adoption and reliance on open-source software development have made open-source projects a lucrative target. Open-source software is ubiquitous in modern software development; one of the estimates shows open-source makes up 90% of the software in use today [33] [37]. Such extensive use creates a large attack surface for cybercriminals looking to exploit vulnerabilities or introduce vulnerabilities. The open-source development model also encourages global collaboration; while it promotes innovation and rapid development, it also introduces security challenges. The decentralized nature of open-source projects leads to nonstandard security practices and makes it easier for attackers to introduce malicious code without immediate detection. Dependency confusion and typosquatting are the other two attack techniques used by the threat actors in addition to inserting malicious code directly into OSS projects. Dependency confusion involves uploading malicious packages with names like legitimate internal packages to public repositories, which are then downloaded into software builds. Typosquatting exploits slight variations in package names to deceive developers into downloading malicious packages [29]. These methods are relatively low effort but have extensive and far-reaching impacts. Several high-profile incidents, such as the Log4j vulnerability in December 2021, demonstrated how a single vulnerability in a widely used open-source logging library could have catastrophic consequences affecting millions of devices and countless organizations worldwide. Another example is the npm incident 2022, where over 200 malicious packages were discovered. These incidents show the potential scale and impact open-source software compromises can have. The compromise of a popular open-source library propagates quickly through the software supply chain, resulting in widespread exploitation and breaches.

5.1. Mitigation Strategies

Here are some mitigation strategies to combat the increasing targeting of open-source software. (a) Enhanced Security Practices for OSS usage: Organizations should adopt robust security practices for managing open-source project usage in their organization. Including regular vulnerability assessments, automated security scanning, and the use of tools like Software Composition Analysis (SCA) to identify and remediate vulnerabilities in software dependencies. (b) Supply Chain Transparency: Implementing a Software Bill of Materials will enhance transparency in an organization's software supply chain. SBOMs give a detailed account of all components used in each software, enabling organizations to quickly identify and address vulnerabilities when they are discovered [21][22]. (c) Community and Vendor Collaboration: Collaboration between open-source communities, vendors, and security researchers is vital to combat software supply chain attacks. Shared intelligence and coordinated response efforts will help detect threats. Initiatives like the Open-Source Security Foundation (OpenSSF) are playing an important role in encouraging such collaboration and improving the overall security posture of open-source projects [32].

6. Rise of State-Sponsored Attacks

State-sponsored attacks have dramatically changed the landscape of cyber threats. National government-backed groups orchestrate these attacks. Geopolitical conflicts and rivalries have driven many nations to leverage cyber capabilities as part of their strategic arsenal. Generally, state-sponsored attacks can be closely linked to geopolitical tensions; for example, the ongoing conflict between Russia and Ukraine resulted in a surge in cyber operations aimed at destabilizing critical infrastructure and spreading misinformation [34]. State-sponsored threat actors have advanced skills and resources and are capable of conducting highly organized and sophisticated attacks. These groups often employ zero-day exploits and develop their own custom malware to infiltrate and remain undetected within the target's network for years. The primary targets for these nation-state-backed groups are critical infrastructure, government agencies, and key industries. Software supply chain attack techniques are lucrative for these actors since they are an easier route to get past the generally fortified defenses at these targets. The SolarWinds attack in 2020 was an ultimate example where the attackers inserted malicious code into the Orion software and compromised thousands of organizations globally, including U.S. federal agencies [35]. They are believed to be linked to Russia, and the aftermath of the incident showed the planning and execution excellence. Typical cybercriminals are motivated by financial gain, but state-sponsored attackers have strategic objectives like espionage, sabotage, and disruption of critical services. Their typical targets include sectors of national importance, like defense departments, energy generation and storage, finance, and healthcare. One another great example is the Microsoft Exchange Hack in early 2021. A state-sponsored group known as Hafnium, believed to be linked to China, exploited vulnerabilities in Microsoft Exchange Server. The attackers gained access to email accounts and installed backdoors. This attack affected thousands of organizations worldwide, including government agencies, defense contractors, and educational institutions.

6.1. Mitigation Strategies

There is no silver bullet to counter against state-sponsored threats. General security hygiene and enhanced detection and response posture are foundational to defend against these nation-state actors. (a) Implementing advanced threat detection and response systems is a must. Applying machine learning and using user behavioral analytics-based detection helps defend against sophisticated threats in real-time. (b) Once again, strengthening collaboration between the public and private sectors can enhance information sharing and collective defense efforts. Initiatives like the Cybersecurity and Infrastructure Security Agency (CISA) in the United States promote such partnerships to bolster national cybersecurity resilience. (c) Conducting regular security audits, patching vulnerabilities promptly, and updating software and hardware can mitigate the risk of exploitation by state-sponsored actors.

7. Proliferation of Dependency Confusion Attacks

Dependency confusion attacks, known as namespace confusion or substitution attacks, have become popular in recent years. They particularly affect the software supply chain. These attacks exploit weaknesses in how software dependencies are managed, especially in systems that utilize private and public repositories. Dependency confusion attacks happen when an attacker knows the names of internal packages used by an organization but does not publish them in public repositories. The attacker then creates a malicious package with the same name and publishes it to public repositories with higher version numbers. Because of the way many package managers resolve dependencies, these malicious packages are inadvertently pulled into the software build process. A notable instance of this attack method was demonstrated by ethical hacker Alex Birsan in 2021.

Birsan successfully breached the internal systems of over 35 major companies, including Microsoft, Apple, and Tesla, by exploiting dependency confusion. His proof-of-concept attack involved uploading packages with names matching those used internally by these companies to public repositories like npm, PyPI, and RubyGems. When a company’s automated build system pulled these public packages, the malicious code was executed, allowing Birsan to gain access to the customer environment. Modern software development often involves complex dependency trees with numerous direct and transitive dependencies. This complexity can obscure the visibility of individual packages being used, making it difficult to ensure that all dependencies are correctly sourced and verified.

7.1. Mitigation Strategies

Organizations can implement the following strategies to defend against dependency confusion attacks. (a) Registering private package names in public repositories can prevent attackers from creating malicious packages with those names. (b) Enforcing strict version control and pinning specific versions of dependencies can prevent building systems from inadvertently pulling newer, potentially malicious versions from public repositories. This practice will ensure that only trusted versions of packages are used in the organization’s build process. (c) Automate security scanning tools to monitor and verify dependencies.

This will help detect and block malicious packages before they are included in the build. (d) Lastly, robust auditing and monitoring systems should be implemented to track the usage and source of dependencies used within a project. Regular audits can help identify and address potential security gaps.

8. Best Practices for Organizations

8.1. Maintain Current Software Asset Inventory

A comprehensive asset inventory is crucial for any cybersecurity strategy, particularly supply chain security. Organizations should aspire and pursue to keep track of all software installations, though it can be challenging. Automated tools can be used to track software installs and utilizations. A centralized inventory will help proactively hunt for vulnerabilities and enable organizations to respond swiftly to any emerging situation.

8.2. Assess Vendor's Security Posture

Commonly, software vendors often lack the same level of protection against supply chain attacks as enterprises that procure them for use. Organizations should conduct third-party risk assessments to understand their software vendor's security posture. Before engaging their services, trust and transparency with vendors regarding information access and usage should be established. Third-party risk management assessments should be used with a vendor security rating system to independently validate cyber risk assessment responses. Vendors' security practices should be regularly reviewed and audited.

8.3. Audit Unapproved Shadow IT

Shadow IT refers to any IT infrastructure that hasn't been vetted by a company's security team or provisioned by the IT team responsible for that function. The shift to remote working during the global pandemic led many employees to use personal devices for work. In large enterprises, business teams and software development teams sometimes bypass organizations’ IT processes and use public cloud infrastructure or SaaS solutions to accelerate their POC or MVP development. Monitoring approved devices and shadow IT, especially those connected to the internet, helps detect and prevent some types of software supply chain attacks.

8.4. Continuous Supplier Risk Validation

Supplier risk should be assessed continuously, not just at the initial engagement. Organizations should engage their suppliers at crucial points in the supply chain, including those involved in production, manufacturing, and delivery. Regular consultations, security practice reviews, and audits are essential to maintain a secure supply chain. Identifying root causes and addressing them periodically will significantly reduce threats.

8.5. Endpoint Detection and Response (EDR) Solutions

Use endpoints are often the entry points for supply chain attacks due to inadequate security and user behavior. EDR systems protect endpoints by detecting and responding to threats, preventing them from spreading across the network. EDR solutions enhance security by analyzing user behavior and identifying anomalies. They help confine threats, provide detailed attack analysis, and offer insights into how to eliminate them.

8.6. Implement Strong Code Scanning and Peer Review Process

A robust code peer review process and scanning process can prevent both unintentional and intentional errors that might be introduced into the code base and reduce the risk of supply chain attacks.

8.7. Secure CI/CD Infrastructure

Regularly install security patches for operating systems and CI/CD software stack to ensure secure builds. Access to CI/CD infrastructure should be strictly controlled and audited regularly. Appropriate secret management solutions should be used alongside the CI/CD platforms to prevent secret leaks.

8.8. Secure Software Development Life Cycle (SDLC)

Adopting secure software development life cycle (SDLC) models, such as the Microsoft Security Development Lifecycle or the National Institute of Standards and Technology's Secure Software Development Framework, enhances software security. Secure socket layer encryption, digital signatures, and strict input validation should be implemented to build secure software updates.

9. Conclusion

As the attack on software supply chains continues to grow and evolve, it is crucial for organizations to take proactive measures to strengthen their defenses. This paper offers an in-depth look at the current landscape of software supply chain attacks, showcasing the tactics, techniques, and procedures used by both advanced nation-state actors and less well-equipped threat actors. By delving into how these attacks work and examining real-world examples, this paper uncovers insights needed for threat mitigation. The trends identified from the historical events, like a rise in attacks on open-source software, further reinforce the importance of improved open-source software security measures and collaboration within the open-source community. By stressing the significance of maintaining an inventory of software assets, regularly evaluating supplier risks, and securing endpoint devices, this paper offers actionable insights for organizations to enhance their cybersecurity defenses. In conclusion, this document serves as a tool for organizations aiming to defend themselves against the challenges posed by software supply chain attacks.

References

[1] Defense Technical Information Center, Securing the Supply Chain from Cyber-Attack: Challenges and Best Practices, (Report No. AD1108057), 2020. [Online]. Available: https://apps.dtic.mil/sti/trecms/pdf/AD1108057.pdf

[2] Jon Boyens et al., "Supply Chain Risk Management Practices for Federal Information Systems and Organizations," *National Institute of Standards and Technology Special Publication 800-161*, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[3] Haydn Brooks, Risk Ledger, Top Supply Chain Cyber Security Trends 2024, 2024. [Online]. Available: https://riskledger.com/resources/cyber-security-trends

[4] Supply Chain Trends, Critical Infrastructure, and Cyber Security in 2024, CyberTalk, 2023. [Online]. Available: https://www.cybertalk.org/2023/12/29/supply-chain-trends-critical-infrastructure-and-cyber-security-in-2024/

[5] Kevin Townsend, Cyber Insights 2024: Supply Chain, SecurityWeek, 2024. [Online]. Available: https://www.securityweek.com/cyber-insights-2024-supply-chain/

[6] Carolynn van Arsdale, The State of Software Supply Chain Security Report 2024: Key takeaways, ReversingLabs, 2024. [Online]. Available: https://content.reversinglabs.com/state-of-sscs-report/the-state-of-sscs-report-24

[7] ReversingLabs, Gartner Report: Mitigate Software Supply Chain Risk - Key Takeaways. [Online]. Available: https://content.reversinglabs.com/gartner-sscs-risk-mitigation/gartner-report-mitigate-sscs-risk-takeaways

[8] Carolynn van Arsdale, How NIST CSF 2.0 and C-SCRM Help Manage Software Supply Chain Risk, ReversingLabs, 2024. [Online]. Available: https://content.reversinglabs.com/special-nist-csf-cscrm-sscs-risk

[9] Mitigate Enterprise Software Supply Chain Risks, Gartner Research, 2023. [Online]. Available: https://www.gartner.com/en/documents/4893131

[10] 9th Annual State of the Software Supply Chain, Modernizing Open-source Dependency Management, Sonatype. [Online]. Available: https://www.sonatype.com/state-of-the-software-supply-chain/modernizing-open-source-dependency-management

[11] Zach Capers, Three in Five Business Affected by Software Supply Chain Attacks in Last 12 Months, Capterra, 2023. [Online]. Available: https://www.capterra.com/resources/software-supply-chain-attacks/

[12] Bart Lenaerts, What is Supply Chain Attacks? CrowdStrike, 2023. [Online]. Available: https://www.crowdstrike.com/cybersecurity-101/cyberattacks/supply-chain-attacks/

[13] The Software Supply Chain is Under Attacks, Cloudflare. [Online]. Available: https://www.cloudflare.com/the-net/supply-chain-attacks/

[14] Paul Roberts, A Partial History of Software Supply Chain Attacks, ReversingLabs, 2024. [Online]. Available: https://www.reversinglabs.com/blog/a-partial-history-of-software-supply-chain-attacks

[15] Matt Kapko, Cybersecurity Dive, Costs of Software Supply Chain Attacks Could Exceed $46B this Year, 2023. [Online]. Available: https://www.cybersecuritydive.com/news/software-supply-chain-attacks/650148/

[16] Sumeet Wadhwani, Attacks on Software Supply Chains to Increase in Severity in 2023: Report, Spiceworks, 2022. [Online]. Available: https://www.spiceworks.com/it-security/security-general/news/software-supply-chain-attacks-rising/

[17] Ax Sharma, 6 Most Common Types of Software Supply Chain Attacks Explained, CSO, 2023. [Online]. Available: https://www.csoonline.com/article/570743/6-most-common-types-of-software-supply-chain-attacks-explained.html

[18] Snyk, "*2023 Supply Chain Attacks Report*," Cybersecurity Ventures, 2023. [Publisher Link]

[19] Threat Landscape for Supply Chain Attacks, European Union Agency for Cybersecurity (ENISA). [Online]. Available: https://www.enisa.europa.eu/publications/threat-landscape-for-supply-chain-attacks

[20] Mackenzie Jackson, Supply Chain Attack: 6 Steps to Protect Your Software Supply Chain, GitGuardian, 2021. [Online]. Available: https://blog.gitguardian.com/supply-chain-attack-6-steps-to-harden-your-supply-chain/

[21] SLSA, Threats and Mitigations. [Online]. Available: https://slsa.dev/spec/v0.1/threats

[22] AxSharma, Codecov Hack Aftermath: Hundreds Breached, Many more to Follow, SecurityReport, 2021. [Online]. Available: https://securityreport.com/codecov-hack-aftermath-hundreds-breached-many-more-to-follow/

[23] Justin Bhar, Top Software Supply Chain Security Solution Approaches: Pros and Cons, Security Boulevard, 2022. [Online]. Available: https://securityboulevard.com/2022/11/top-software-supply-chain-security-solution-approaches-pros-and-cons/

[24] Dr. Trey Herr, Breaking Trust: Shades of Crisis Across an Insecure Software Supply Chain, Atlantic Council, 2020. [Online]. Available: https://www.atlanticcouncil.org/in-depth-research-reports/report/breaking-trust-shades-of-crisis-across-an-insecure-software-supply-chain/#improve

[25] Will Loomis et al., DFRLab, Software Supply Chain Security: The Dataset, 2023. [Online]. Available: https://dfrlab.org/2023/09/27/software-supply-chain-security-the-dataset/

[26] Ax Sharma, Cryptocurrency Launchpad Hit by $3 Million Supply Chain Attack, Ars Technica, 2017. [Online]. Available: https://arstechnica.com/information-technology/2021/09/cryptocurrency-launchpad-hit-by-3-million-supply-chain-attack/

[27] Juan Aguirre, NPM Hijackers at it Again: Popular 'COA' and 'RC' Open-Source Libraries Taken Over to Spread Malware, Sonatype, 2021. [Online]. Available: https://blog.sonatype.com/npm-hijackers-at-it-again-popular-coa-and-rc-open-source-libraries-taken-over-to-spread-malware

[28] Microsoft Security Response Center, Customer Guidance on Recent Nation-state Cyber Attacks, Microsoft, 2020. [Online]. Available: https://msrc.microsoft.com/blog/2020/12/customer-guidance-on-recent-nation-state-cyber-attacks/

[29] Thomas Hunter II, Compromised NPM Package: Event-Stream, Medium, 2018. [Online]. Available: https://medium.com/intrinsic-blog/compromised-npm-package-event-stream-d47d08605502

[30] Post-Mortem: April 2021 Incident, Codecov, 2021. [Online]. Available: https://about.codecov.io/apr-2021-post-mortem/

[31] Karl Fosaaen, Attacking Azure Container Registries with Compromised Credentials, NetSPI, 2020. [Online]. Available: https://www.netspi.com/blog/technical/cloud-penetration-testing/attacking-acrs-with-compromised-credentials/

[32] Sonatype, State of the Software Supply Chain. [Online]. Available: https://www.sonatype.com/state-of-the-software-supply-chain/introduction

[33] Marc Ohm et al., "Backstabber's Knife Collection: A Review of Open Source Software Supply Chain Attacks," Detection of Intrusions and Malware, and Vulnerability Assessment, pp. 23-43, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[34] Jeferson Martinez, and Javier M. Duran, "Software Supply Chain Attacks, A Threat to Global Cybersecurity: SolarWinds' Case Study," *International Information and Engineering Technology Association*, vol. 11, no. 5, pp. 537-545, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[35] William J. Heinbockel, Ellen R. Laderman, and Gloria J. Serrao, "*Supply Chain Attacks and Resiliency Mitigations*," MITRE Technical Report, pp. 1-78, 2017. [Publisher Link]

[36] Robert J. Ellison et al., "*Evaluating and Mitigating Software Supply Chain Security Risks*," Software Engineering Institute, pp. 1-50, 2010. [Publisher Link]

[37] Cailean Osborne, "Public-private Funding Models in Open Source Software Development: A Case Study on Scikit-learn," *arXi*v, pp. 1-15, 2024. [CrossRef] [Google Scholar] [Publisher Link]