International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor 2024: 7.101

The Persistent Threat of Software Supply Chain Attacks: Lessons Still Unlearned

Karthikeyan Thirumalaisamy

Independent Researcher, Washington, USA.
Email: *kathiru11[at]gmail.com*

Abstract: ***The software supply chain represents a major cybersecurity threat which has become one of the most significant dangers in today's digital world. Organizations persist in making basic security mistakes despite major incidents such as SolarWinds, Kaseya and Log4j which exposed fundamental weaknesses in software development and distribution processes. Lessons from past incidents are often not fully understood and integrated to protect critical infrastructure and enterprise software ecosystems from dangerous exposure. This paper examines software supply chain attacks from the previous year through incident analysis to extract valuable lessons from these events. The paper presents practical recommendations to boost software supply chain security resilience and accountability.***

Keywords: Software supply chain security, Supply chain attack, Cybersecurity threats, Infrastructure resilience, Security accountability, Incident analysis

1. Introduction

Software supply chains are arguably more complex, distributed, and third-party dependent than ever before, and while the interconnections foster creativity and efficiency, they also present an extraordinary amount of risk. One of the most significant risk areas is, of course, software supply chain attacks, where threat actors access trusted software providers or participate in trusted development processes to compromise the end user's systems. Despite notable incidents like the SolarWinds Orion compromise (2020) and the Log4j incident (2021), which had profound impacts across many sectors, software supply chain attacks have been increasing yearly. Unfortunately, things only get worse with the NPM Typosquatting Campaign, Blue Yonder Ransomware Attack, and Polyfill.io and BootCDN compromise (2024); additionally, in 2025, the GitHub Action "tj-actions/changed-files" exploit, and two significant occurrences involving the infiltration of NPM packages by the Lazarus Group. While it is extremely disheartening to see these events unfold, it is more upsetting because there appears to be little momentum for long-term responses and action from the cybersecurity community and software industry as a whole.

These attacks are not new in the grand scheme of software supply chains; however, their continued ability to find success signals an unresolved fundamental issue. There remains to be a high lack of visibility to software dependencies, no security standards in development and deployment, and clearly a lack of basic standards for comprehension around software integrity. As adversaries become even bolder in their methods in the years to come, we must have a comprehensive awareness of weaknesses in the software supply chain that could disrupt IT; organizations, national and world infrastructure cannot continually be compromised by the same threats, no matter the sophistication.

In this paper, we review some of the key software supply chain attacks we experienced last year, providing case-by-case analysis of each incident and what vulnerabilities surfaced; we will then take some lessons away from these case studies, and conclude with some actionable tools in efforts to enhance better resiliency, transparency, and accountability in the software supply chain space.

2. Understanding Software Supply Chain Attacks

A software supply chain attack is a cyberattack in which malicious actors compromise a trusted element within the software development or delivery process in order to distribute malware or gain unauthorized access to systems. Rather than targeting an organization directly, attackers exploit third-party software components, such as open-source libraries, commercial software vendors, build tools, or update mechanisms. Because modern software often relies heavily on external dependencies, compromising a single component can give attackers access to multiple downstream targets.

3. Key Characteristics and example of Software Supply Chain Attacks

Below is a brief introduction to the key characteristics of software supply chain attacks.

3.1. Indirect Entry Point

Attackers do not focus on going after the end victim directly; instead, they compromise a trusted third-party supplier, developer, or software component that the victim would be reliant on. By compromising a trusted third-party, attackers can bypass standard perimeter defenses.

For example, **the SolarWinds Orion compromise (2020)** is one of the most obvious examples of a supply chain attack with an indirect entry point. The attackers did not penetrate target organizations directly; they compromised a trusted vendor (SolarWinds), and that vendor's software was installed and trusted in over 18,000 organizations. The attackers leveraged that third-party trust to access the networks of major US federal agencies, Fortune 500 companies, and critical infrastructure providers without raising alarms.

3.2. Abuse of Trust

These attacks take advantage of the established trust that exists between an organization and its software upstream, vendors, partners or open-source contributors. There is an unwritten rule that once a component is trusted, and incorporated, the code of that component is rarely examined, unless in comparison to things perceived as external threats.

For example, **the Event-Stream NPM Package Attack (2018)** is the best example of abusing trust. The attacker was able to gain the trust of the original maintainer, by offering to work with them to help maintain it. The open-source ecosystem trusted the package because it was old, and it had a good reputation. Developers and users inherited the malicious code, unknowingly trusting the rest of the software supply chain to be coded safely. No one thought a trusted, mature package would be leveraged to deliver malware, and it was not observed as risky or suspect.

3.3. Widespread Impact

One compromised asset can impact multiple upstream assets. This creates a very scalable and appealing attack approach to nation-state and adversarial actors.

For example, the **Log4Shell Vulnerability in Log4j (2021)** is the perfect representation of the breadth of impact. Log4j is so ubiquitous with almost all enterprises, cloud, and consumer-facing software products relying on it to execute necessary programming code. The vulnerability impacted millions of servers and applications around the world, including Apple, Amazon, Google, Microsoft, Cloudflare and the U.S. government. Since Log4j is often bundled as a transitive dependency, many organizations did not even realize they were utilizing it. The attack surface was enormous any input field, HTTP header, or log message could trigger exploitation.

3.4. Stealth and Persistence

Frequently, these attacks can be undetectable for long periods of time, as the malicious code is likely to be embedded deep within otherwise legitimate software. They can also be configured so that they allow backdoor access or those kinds of stealth as a persistence mechanism.

For example, the **SolarWinds Orion Attack (2020)** is also an example for stealth and persistence attack. The attackers inserted a stealthy backdoor - called SUNBURST - into the company's regular updates of Orion software platform. The SUNBURST malicious code was digitally signed and distributed legitimately as part of the SolarWinds software updates that spanned March through June of 2020. The attackers were able to go undetected for a minimum of nine months in a very stealthy manner, leveraging the compromised software for access to the networks of approximately 18,000 (16,000 agency clients and private companies) customers. The SUNBURST backdoor was configured to act like legitimate network traffic as far as backdoor software goes, and it was configured to execute only under certain conditions in order to go undetected. The attacker established persistence via legitimate administrative tools after the initial access, and lateral movement detected the victim network. The attack was discovered only in December 2020 and was not by SolarWinds, but by a security researcher from FireEye when FireEye noticed some suspicious activity in their own network.

3.5. Complex Attack Surface

Today's software supply chain has multiple components and an entire ecosystem: libraries (some open source), build tools, CI/CD pipelines, API programs, and third-party integrations, leading to an extensive and difficult-to-secure attack surface.

For example, the **Codecov Bash Uploader Compromise (2021)** illustrates a complex attack surface. In this case, an attacker modified a script that exfiltrated sensitive environment variables, including API tokens, credentials, and keys, from customers' build environments to a remote server they controlled. The compromise ran for approximately 2 months before customers discovered the breach.

A simple yet powerful change in a Bash script, used by so many, had gone undetected, as there was no integrity check in the download and execution process of the script. The script was run in the CI/CD environments of thousands of users, so it was difficult to know every system impacted. The attacker was able to pull secrets from cloud environments, which illustrated the interconnectedness of modern build pipelines, and how small tools that might not seem important in the software development lifecycle can lead to company-wide risk. This incident also illustrated how build systems, dependency managers, cloud secrets, and environment variables intersect with each other in ways that are difficult to monitor holistically.

3.6. Difficult Attribution

Because of third party involvement, it can be difficult to determine where and how a breach occurs. This task becomes even more complex when it comes to attribution arising from open-source contributors or compromised insiders. Often, investigations include a variety of vendors which delay timeline response and coordination.

For example, the **NotPetya Malware Attack (2017)** is an excellent example in the context of attribution. The NotPetya malware incident spread quickly across the world and was initially appearing as ransomware looking to mislead analysts and slow down response times. It utilized legitimate hacking tools like Mimikatz and the EternalBlue exploit, which further confused attribution to specific actor compromises. Some technical artifacts appeared criminally motivated, and some appeared to suggest nation's state involvement. Conversely, upon deep analysis, security experts concluded that it was a destructive wiper meant to widely cause destruction, not extort funds.

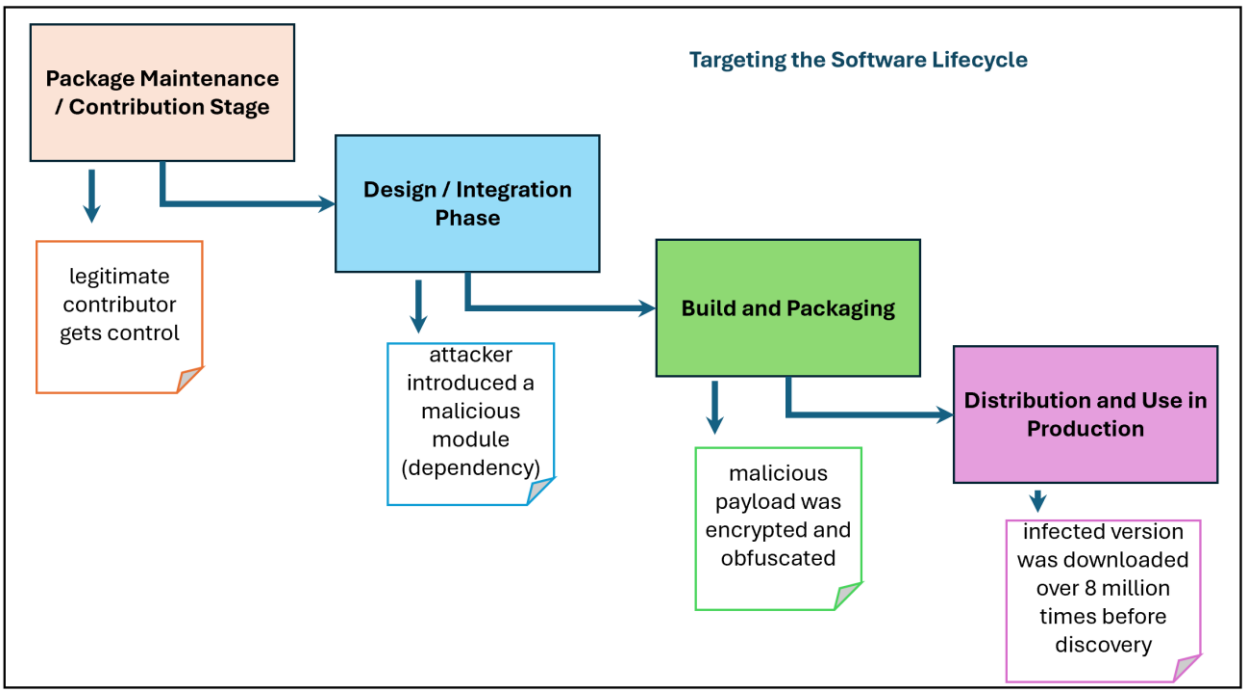
3.7. Targeting the Software Lifecycle

Attacks can happen at any time within a software development lifecycle, whether during coding, compiling, packaging, or updating. Attackers may seek to compromise build servers, repositories, or even developer machines.

Security needs to be implemented from development to delivery to deployment.

For example, the **Event-Stream NPM Package Attack (2018)** is a commonly referenced example of targeting the software lifecycle. In this case, event-stream (a popular Node.js package utilized by millions of JavaScript developers) was compromised after a new maintainer was added to the project. The attacker added a malicious dependency (flatmap-stream) that was aimed at a specific downstream application: the cryptocurrency wallet Copay.

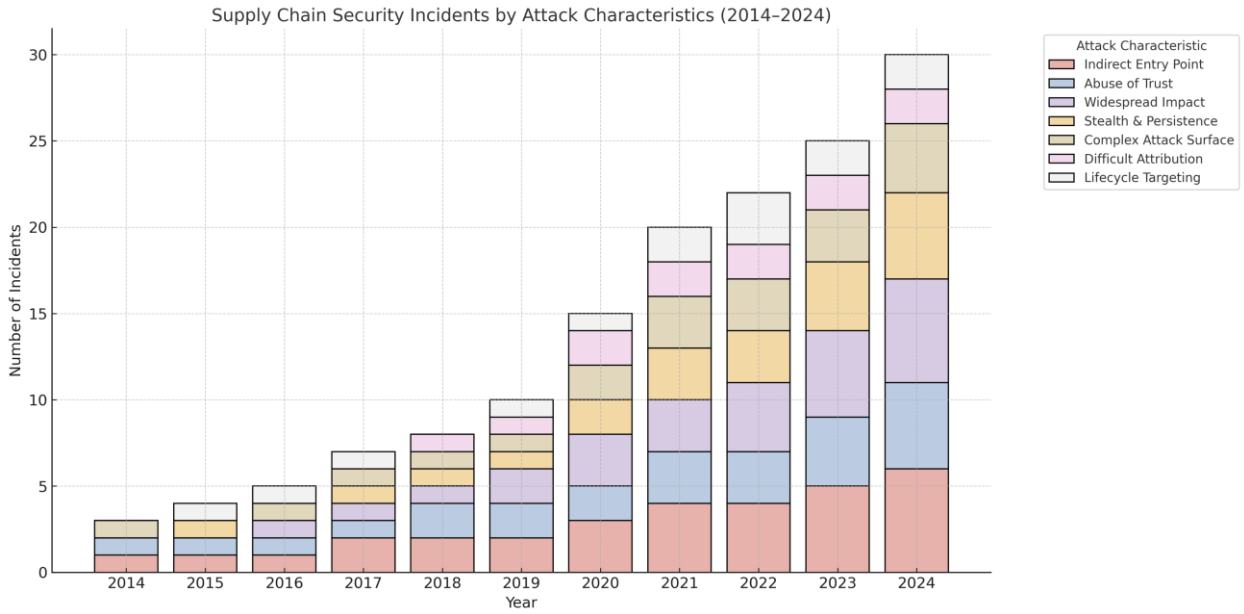
This attack purposefully targeted many phases of the SDLC within an open-source context.



4. Persistent increase in Supply Chain Attacks

Software supply chain attacks have steadily increased in both frequency and complexity over the last 10 years. This is not an arbitrary increase but a persistent one, representing a durable change in the landscape of cybersecurity. After a large number of software supply chain attacks and data breaches in the last decade, it becomes clear that these are not an anomaly or temporary trend, but rather the future framework of increasingly complex cyberattack strategies and tactics. If we do not fundamentally reshape our security culture, policies, and technology, we are destined to see an increase in this behavior.

The chart below illustrates the evolution of software supply chain security incidents from 2014 to 2024, broken down by key attack characteristics. Each colored section of the stacked bar represents the number of incidents associated with a particular characteristic for that year



The chart showing the nature of supply chain attacks reveals that there are still the same types of events happening which indicates that much of the software industry and cybersecurity professionals have failed to learn or take appropriate action based on past attacks.

Knowing these characteristics of each attack helps:
- Make informed defenses at various points of the supply chain by security teams.
- Establish more relevant policies by policymaker and vendors.
- Encourage developers to harden CI/CD pipelines and better check third-party dependencies.

5. Comprehensive Analysis of Recent Supply Chain Attacks

The number and sophistication of supply chain attacks in recent years, especially in 2024, emphasizes the urgent need for a comprehensive analysis of their tactics, targets, and outcomes. This section provides an in-depth examination of critical incidents that occurred in 2024, uncovering common patterns, emerging trends, and systemic vulnerabilities.

5.1. XZ Utils Backdoor (March 2024)

XZ Utils is a popular open-source compression/decompression library that is commonly available in Linux distributions for managing .xz files. Because it is part of the core utilities of many systems, it is a high-value asset for attackers.

An individual who used the alias Jia Tan began contributing to the XZ Utils project. Over time, this individual became credible in the community by regularly submitting patches that people found useful and volunteering to do additional maintenance work. Ultimately, Jia Tan was given commit access to the project by the original maintainer who had become spammed and burnt out. Jia Tan injected malicious code into XZ Util versions 5.6.0 and 5.6.1.

The added backdoor was highly obfuscated, such that it was designed to:
- Activate exclusively in certain environments (e.g., specific build environments).
- Leverage the liblzma library (an integral part of XZ Utils) to provide unauthorized remote access via OpenSSH.
- Allow a remote attacker to bypass authentication and establish shell access on a target system.

5.1.1. Key Characteristics of the Attack

The following table shows the characteristics of the attack to facilitate better understanding.

Characteristic	Description
Abuse of trust	Attacker gained trust over several years.
Stealth and Persistence	Backdoor was hidden in compressed data and complex code.
Widespread Impact	XZ Utils is used across nearly all Linux systems.
Complex Attack Surface	The backdoor activated only in specific environments and certain conditions

5.1.2. Key Learnings of the Attack

- **Maintainer Vulnerability:** With the project being solely maintained by a single overstretched developer, a malicious actor could exploit that situation and provide help and then gain power/influence.
- **Long-Term Social Engineering:** The malicious actor worked for up to years on the project, and this demonstrates how having social engineering for a long time, trust-based development can be seriously compromised.
- **Insider Threats are Real:** The attack came from a trusted contributor, which is a reminder that insider threats are indeed real threats even in open-source communities.
- **Build System Exploits:** Malicious code was hidden in build scripts and not main source code, which highlights how security analysis/reviews need to include the entire build chain, not just the source code.
- **Manual Detection is Not Sustainable:** The backdoor was discovered through manual analysis, due to chance not due to automated tools, this is gut wrenching and suggests inadequacies in current detection systems.
- **The Risk of Dependencies that are Widely Used:** XZ Utils is used in nearly all Linux systems, so a compromise could have resulted in immense havoc. This illustrates the systemic risk that foundational software components can present.
- **Need for Ecosystem Help:** The problem with this project was that there was no institutional support. This makes the project vulnerable. Fundamental open-source infrastructure needs sustainable funding, governance, and shared responsibility.

5.2. NHS Supplier Synnovis Ransomware Attack (June 2024)

Synnovis, a pathology services organization has been involved with Guy's and St Thomas' NHS Foundation Trust, King's College Hospital NHS Foundation Trust, and SYNLAB, experienced a major ransomware attack. It impacted the National Health Service (NHS) in south east London and caused disruption to valuable health services and risked sensitive patient data.

The attack was perpetrated by Qilin, a Russian criminal ransomware group that locked the organisation's I.T. systems, encrypted data and demanded a ransom from Synnovis. The attack brought down virtually all of Synnovis's digital mechanisms and they were reliant on manual systems including how to identify samples and relaying test results digitally to clinicians. The manual system significantly limits the quantity and speed of pathology services.

The disruption led to postponed thousands of elective procedures and outpatient appointments, including approximately 200 cancer-related surgical procedures. Blood transfusion and other critical diagnostic services were affected. The attackers published approximately 400 GB of sensitive data on their darknet site that contained patient names, dates of birth, NHS numbers and blood test results.

5.2.1. Key Characteristics of the Attack

The following table shows the characteristics of the attack to facilitate better understanding.

Characteristic	Description
Indirect Entry Point	Attack targeted Synnovis, a pathology service supplier, not NHS directly.
Abuse of Trust	NHS trusts relied on Synnovis for critical services, assuming robust cybersecurity posture.
Widespread Impact	Over 7 NHS trusts, including multiple hospitals and clinics, faced diagnostic delays and surgery cancellations.
Stealth and Persistence	The attackers gained deep access and encrypted large volumes of data, indicating persistence before detection.
Complex Attack Surface	The integration of Synnovis with hospital IT and diagnostic systems increased the attack surface.

5.2.2. Key Learnings of the Attack

- **Third-Party Risk Management is Essential:** As organizations rely on third-party vendors there is a need for rigorous security assessments, audits, and monitoring of vendor systems in real-time.
- **The Healthcare Sector is a Target-Rich Environment:** High-value targets (diagnostics and pathology) that must operate in urgent circumstances (our patients) are vulnerable to attack.
- **The extent of damage increases when detection is delayed:** Active access from attackers before encryption demonstrates the potential for detection and response systems to improve threat detection and response arrangement.
- **Business Continuity Planning is Required:** The events that transpired clearly illustrated the need to implement robust backup systems and manuals to fallback on if systems could not return.
- **Zero Trust must also apply to third-party vendors:** Even if you trust your partners, they should only have access to systems at the least privilege level and avoid access to sensitive systems and data altogether.
- **Data encryption and backups should not be optional:** All system and data should be backed up robustly and offline and encrypted and kept in secure offline storage - hacking attempts should be as hard as possible to extract any leverage for ransom negotiations and speed the recovery.

5.3. Cisco Data Breach by IntelBroker (October 2024)

Cisco Systems experienced a noticeable data breach involving its publicly facing DevHub platform; the threat actor known as IntelBroker who identified themselves as from the cybercrime forum BreachForums made claims surrounding this breach. Cisco indicated that all of its internal systems were safe however there was still significant exposure due to having a public development resource.

IntelBroker claimed to have exfiltrated potentially up to 4.5TB of information and some of the leaked data related to Cisco's B2B clients, with reports showing over 800 unique companies.

5.3.1. Key Characteristics of the Attack

The following table presents the characteristics of the attack to facilitate better understanding.

Characteristic	Description
Indirect Entry Point	Attackers exploited misconfigurations in a publicly accessible DevHub, not Cisco's core infrastructure.
Abuse of Trust	Cisco's platform hosted sensitive development data that was assumed to be securely managed.
Widespread Impact	Data included information from Cisco's clients, impacting over 800 organizations.
Stealth and Persistence	Attackers accessed data in early October, but leaks occurred weeks later, suggesting undetected access.
Targeting the Software Lifecycle	Attack focused on development assets: source code, build artifacts, config files — core parts of SDLC.

5.3.2. Key Learnings of the Attack

- **Public-Facing Development Platforms Require Security Assessments**: Development platforms like DevHub that are not configured properly can provide insights into sensitive information that could be leveraged for a breach. Ongoing audits and reviews of access are always necessary.
- **Code and Configuration Data are Valuable Assets**: Data obtained from a breach of a software product can lead to value from source code snippets, API keys, and deployment scripts compromising the ability of downstream systems. As a long-term strategy, these assets need to be managed as sensitive data, like user data.
- **Supply Chain Exposure Impacts Customers**: The visible data exposed and also showed some degree of information about Cisco's partners and customers that should have remained confidential. The breach shows that everybody has the risk of shared exposure across the software supply chain.
- **Threat Actors Target Build and Deployment Pipelines:** The breach highlights that a development environment can also be a target leading to a larger breach or data exfiltration.
- **Visibility and Monitoring are Necessary:** The delayed identification and public announcement of the breach shows the current state of monitoring of developer platforms or asset exposure.

5.4. Blue Yonder Ransomware Attack Affecting Multiple Retailers

In November 2024, Blue Yonder organized a ransomware attack, which exemplifies the ways in which vulnerabilities in supply chain software can create broad, systemic impacts that disrupt multiple line of businesses. Blue Yonder, which is a leading supply chain management software supplier owned by Panasonic, faced a ransomware attack that affected its entire managed services hosted environment during the attack. The Termite ransomware group took responsibility, claiming a theft of about 680 GB of data including: database dumps, email files, and PDFs/Word documents. In addition to the Blue Yonder incident, this attack caused staggering secondary or cascade impact on Blue Yonder's top clients,

including Starbucks, Morrisons, Sainsbury's and BIC and Waterstones.

5.4.1. Key Characteristics of the Attack

The following table presents the characteristics of the attack to facilitate better understanding.

Characteristic	Description
Abuse of Trust	Clients relied on Blue Yonder's cloud infrastructure; the breach exploited this centralized trust.
Complex Attack Surface	Blue Yonder's broad service offering—logistics, planning, inventory—created a wide and hard-to-secure surface.
Widespread Operational Impact	Impacted payroll, scheduling, and product distribution for major retailers.
Stealth and Persistence	Indicators suggest attackers moved laterally within systems before triggering ransomware.

5.4.2. Key Learnings of the Attack

- **Holistic Vendor Risk Management:** View EVERY third-party service vendor as a risk. Do a risk assessment of the vendor's security, independently verify the controls with artifact evidence (for example penetration testing report or SOC 2 Type II report), and require them to uphold their contractual obligations to the security controls.
- **Robust Incident Response and Business Continuity Planning:** Good managed services can still make a mistake from time to time. Document and drill on playbooks that consider a ransomware containment plan, a data recovery plan, and a business-process "manual" plan (e.g. payroll via paper check in the case of a compromised service).
- **Zero Trust Segmentation of Managed Services:** A compromise of one module could harm several client consequences. There must be strict segmentation between network layers and application layers based on strict separation of service tenants and services with least privilege.
- **Proactive Monitoring and Threat Detection:** Implement continuous attack monitoring by using behavior analytics of scripted attacks (i.e. anomalous service-to-service calls) and alerts to quickly respond and detect attacks.

6. Consolidated Mitigation Strategies for Supply Chain Attacks

This is a consolidated summary of risk mitigation strategies from all the relevant 2024 supply chain security incidents. We organized the recommendations into categories of factors and recommended actions to build a comprehensive defensive posture.

6.1. Third Party and Vendor Risk Management

- **Vendor Security Assessment:** Always assess the third vendor security posture to be used, such as their information security policies, compliance certifications (i.e. ISO 27001, SOC 2), and prior security incidents before contracting out a service.
- **Cybersecurity Contractual Requirements:** Specify cybersecurity requirements within contracts you want to see in place, for example, data protection standards, data breach notification timelines, and audit rights.
- **Security Monitoring:** Monitor your vendors regularly and in accordance with the risk they present, for example, security practices, systems integrity, performance. Use tools and services that can offer a rating or score of your vendor risk that is continuously updated in real-time.
- **Access Control, and Least Privilege:** Only provide access relevant to the vendor to perform their work. Do not provide extended access (that may be defined as "vendor access") to internal systems or sensitive data.
- **Incident Response Assets:** The Incident Response process should have a clear Incident Response Plan that integrates vendors into it. The plan should outline the vendor's roles and clearly set out communication guidelines for your organization and prevailing communications for the vendor group in the event of an incident.
- **Supply Chain Mapping:** Understand where your vendor's own vendors are providing a service as this risk is not captured in your initial risk.
- **Security Awareness Training:** Encourage or require third parties to provide their employees with cybersecurity best practice training, particularly in relation to third parties interacting with your organization's systems or data.

6.2. Implement Zero Trust Architecture

- **Identity and Access Management (IAM):** Use robust multi-factor authentication (MFA) and enable single sign-on (SSO) using role-based access control (RBAC). Monitoring for anomalous users and login activity
- **Device Trust:** Verify devices meet your security policies before logging in and consider leveraging your endpoint detection and response (EDR) capabilities. Periodically run device health checks.
- **Network Segmentation and Micro-Segmentation:** Break your network into small, segregated sections, which can help prevent lateral movement on your network, as well as restrictively manage access to need-based access. You should always monitor user behavior, device health, and network activity, while also making use of machine learning and threat intelligence to identify anomalies.
- **Data Security:** Use encryption both in transit and at rest, and properly classification and labeling schemes. Develop granular access policies based on data characteristics plus incorporate with Security Automation and Orchestration. Automate the application of policies and the incident response process to create efficiencies for threat detection and incident response.

6.3. Enhanced Monitoring and Threat Detection

- **Security Information and Event Management (SIEM):** Establish a centralized log management system using a SIEM. A SIEM platform collects, correlates, and analyzes security events from throughout your IT environment (servers, endpoints, network devices, cloud, etc.), while also providing the capability analyze real time events as well as configuration alerts and notifications.
  Examples: Splunk, IBM QRadar, Microsoft Sentinel, LogRhythm

- **Endpoint Detection and Response (EDR):** EDR tools monitor and collect activities from endpoints (laptops, servers, workstations) to detect and respond to threats. It uses behavior based threat detection and utilizes it for incident investigation, root cause analysis, and remote containment and remediation.
  Examples: CrowdStrike Falcon, SentinelOne, Microsoft Defender for Endpoint
- **Network Detection and Response (NDR):** Monitors network traffic to identify suspicious patterns, lateral movement, or unauthorized data exfiltration.
  Examples: Darktrace, Vectra, ExtraHop
- **User and Entity Behavior Analytics (UEBA):** Use Machine Learning to establish baselines of normal behavior and detect deviations that may indicate malicious activity or suggest insider threats.
- **Threat Intelligence Integration:** Integrates threat intelligence (TI) feeds into your monitoring tools to enhance detection accuracy by providing contextual information on known threats (IPs, domains, malware hashes).
- **Integration of monitoring with Incident Response (IR):** Integrate all alerts into an established IR process and leverage Security Orchestration, Automation, and Response (SOAR) to automate response actions you define in playbooks.

6.4. Data Protection and Loss Prevention

- **Enforcement of Encryption Standards:** Utilize AES-256 or equivalent encryption capabilities for data in rest and TLS 1.2+ for data in transit.
- **Separate Locations for Sensitive Data:** Store data categorized as high risk (i.e. customer PII, financials, source code) in the form separate locations or pieces with limited access.
- **CSPM:** Use capability tooling (i.e. Wiz, Prisma Cloud) to track information in cloud services along with enforcing access and encryption policies.
- **Data Masking and Tokenization:** Protect fields of sensitive data using data masking in environments that will be used for testing or analytics that do not need to include fields with sensitive data exposed.

6.5. Ensure Backup and Business Continuity

- **Immutable Backups (WORM):** Implement write-once-read-many (WORM) storage to completely prevent attackers from modifying or deleting your backups.
- **Backup Segregation:** Store your backups in environments separate from your production network to avoid contaminating your backups.
- **Geo-Redundancy:** Replicate your backups across different geographic regions with the goal of lessening the impact from physical or geographic disruptions.
- **Ransomware Recovery Planning:** Create tested playbooks as a means of recovering your core systems quickly without paying ransoms.

6.6. Improve Supply Chain Resilience

- **Multi-Vendor Dependency:** The absolute worst thing you can do is lock yourself in with a vendor. You must design your systems to allow for vendor flexibility, from cloud, logistics, payroll.
- **Supplier Business Continuity Assurance:** Ensure your suppliers can show their own disaster recovery and cyber resilience.
- **Supply Chain and Risk Mapping:** Map out your suppliers ecosystem and bill critical points along the way.
- **Scenario Based Planning:** Simulate that one of your suppliers, or your logistics supplier gets hit by a cyber-attack and test your internal capability to respond.

7. Conclusion

With the increase in both the frequency and sophistication of software supply chain attacks, it has become ever more apparent that organizations need to adopt holistic and forward-looking approaches to improve their cybersecurity posture. These breaches often start with attacks on the trusted software provide and then exploit the trust inherent in the software development and distribution processes, making them especially difficult to detect and respond to. The impacts of these breaches can be devastating, not only for the targeted organization but also for its downstream partners, customers, and ecosystems. In this paper, we review recent supply chain attacks, provide a detailed examination of the typical features and a number of example tactics, techniques, and procedures (TTPs) employed by a variety of threat actors, ranging from highly resourced and recommended nation state adversaries to less sophisticated or opportunistic cyber criminals. By examining some of the real-world incidents, the paper will show the evolving nature of these threats, and the vulnerabilities that attackers frequently take advantage of. Furthermore, in addition to the technical examination of the incidents, we revisit some prior incidents, some of which go back as far as 2017, to recap the continuous trends as well as warning signs that attackers have repeatedly overlooked or have ignored. Although the prior incidents have proven to be informative, many of the lessons learned are just lessons learned. A lot of organizations continue to not implement many of the lessons learned from their breaches regarding managing their risk and security postures, leaving defensive gaps. To further this ongoing outstanding challenge, we collated and synthesized a set mitigation strategy based on past cases and based on current best practices. The aim of the recommendations is to be actionable and helpful for companies of varying sizes and maturity levels in order to better protect themselves going forward and to mitigate damages. At the end of the day, this document is a strategic resource intended to help an organization understand the full scope of software supply chain risk, how to learn from prior exploits, and how to take meaningful steps to avoid similar attacks in the future.

References

[1] Jon Boyens, Angela Smith, Nadya Bartol, Kris Winkler, Alex Holbrook, Matthew Fallon, Cybersecurity Supply Chain Risk Management Practices for Systems and Organizations, 2024. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-161r1-upd1.pdf
[2] Varadharaj Varadhan Krishnan, Chain Reaction: Analyzing Trends and Crafting Defenses Against

Software Supply Chain Attacks, (IJCTT-V72I8P110), 2024. [Online]. Available: https://www.ijcttjournal.org/2024/Volume-72%20Issue-8/IJCTT-V72I8P110.pdf
[3] Bao Tran, Patent Attorney, Surge in Software Supply Chain Attacks Demands Heightened Third-Party Vigilance, 2024. [Online]. Available: https://cyble.com/blog/surge-in-software-supply-chain-attacks-heightens-third-party-vigilance/
[4] Supply Chain Attacks: Frequency and Severity Stats, 2025. [Online]. Available: https://patentpc.com/blog/supply-chain-attacks-frequency-and-severity-stats
[5] Tim Freestone, Analyzing the supply chain risks behind the top data breaches in 2024. [Online]. Available: https://www.scmr.com/article/analyzing-the-supply-chain-risks-behind-the-top-data-breaches-in-2024
[6] Security Staff, Software supply chain experiences almost 1 attack every 2 days, 2024. [Online]. Available: https://www.securitymagazine.com/articles/100985-software-supply-chain-experiences-almost-1-attack-every-2-days
[7] OpenBuckets Support, Supply Chain Attacks: The Biggest Cybersecurity Nightmare – 3 Intriguing Case Studies, 2024. [Online]. Available: https://opensecuritylabs.com/blog/2024/03/supply-chain-attacks-top-cases/
[8] Eldan Ben-Haim, Lessons From the Largest Software Supply Chain Incidents, 2024. [Online]. Available: https://www.darkreading.com/vulnerabilities-threats/lessons-largest-software-supply-chain-incidents
[9] Bao Tran, Patent Attorney, Supply Chain Attacks: Frequency and Severity Stats, 2025. [Online]. Available: https://patentpc.com/blog/supply-chain-attacks-frequency-and-severity-stats
[10] Akamai Security Intelligence Group, XZ Utils Backdoor — Everything You Need to Know, and What You Can Do, 2024. [Online]. Available: https://www.akamai.com/blog/security-research/critical-linux-backdoor-xz-utils-discovered-what-to-know
[11] Wikipedia, XZ Utils backdoor, 2024. [Online]. Available: https://en.wikipedia.org/wiki/XZ_Utils_backdoor
[12] Marek Columby, The Blue Yonder Ransomware Attack: A Wake-Up Call for Supply Chain Resilience, 2025. [Online]. Available: https://www.semantic-visions.com/insights/the-blue-yonder-ransomware-attack-a-wake-up-call-for-supply-chain-resilience#:~:text=In%20November%202024%2C%20Blue%20Yonder,by%20the%20Termite%20ransomware%20gang
[13] Andrea Little Limbago, Blue Yonder Outage Could Impact 3.5 Million Companies though Extended Supply Chain, 2024. [Online]. Available: https://www.interos.ai/blue-yonder-ransomware-cyber-attack-finastra-breach/
[14] Secure Blink, Ransomware Attack Cripples Blue Yonder, Disrupting Global Supply Chains, 2024. [Online]. Available: https://www.secureblink.com/cyber-security-news/ransomware-attack-cripples-blue-yonder-disrupting-global-supply-chains
[15] Cyber Management Alliance, How a Ransomware Attack on Synnovis led to chaos at NHS UK: A Timeline, 2024. [Online]. Available: https://www.cm-alliance.com/cybersecurity-blog/how-a-ransomware-attack-on-synnovis-led-to-chaos-at-nhs-uk-a-timeline
[16] Balaji, Cisco Data Breach – Intel Broker Group Stolen 4.5 TB of Data, 2024. [Online]. Available: https://cyberpress.org/cisco-data-breach/
[17] Adi Bleih, Cisco Data Leak: The Facts on the Ground, 2024. [Online]. Available: https://cyberint.com/blog/research/cisco-data-leak-the-facts-on-the-ground/
[18] CYFIRMA, Data Breach Investigation on Cisco, 2024. [Online]. Available: https://www.cyfirma.com/research/data-breach-investigation-on-cisco/
[19] Alanna Titterington, Supply-chain attacks in 2024. [Online]. Available: https://me-en.kaspersky.com/blog/supply-chain-attacks-in-2024/23773/
[20] Paul Schnackenburg, Understanding Supply Chain Attacks and Protecting Your Business, 2025. [Online]. Available: https://www.hornetsecurity.com/en/blog/supply-chain-attacks
[21] Alexander Liskin, Vladimir Kuskov, Igor Kuznetsov, Story of the Year: global IT outages and supply chain attacks, 2024. [Online]. Available: https://securelist.com/ksb-story-of-the-year-2024/114883
[22] zvelo, Countering the Rising Tide of Supply Chain Attacks. [Online]. Available: https://zvelo.com/countering-the-rising-tide-of-supply-chain-attacks
[23] SocRadar, Biggest Manufacturing Industry Attacks 2024, [Online]. Available: https://socradar.io/biggest-manufacturing-industry-attacks-2024