2022 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW) | 978-1-6654-7679-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ISSREW55968.2022.00081

Investigating Novel Approaches to Defend Software Supply Chain Attacks

Md Jobair Hossain Faruk*, Masrura Tasnim†, Hossain Shahriar‡, Maria Valero‡
Akond Rahman§, Fan Wu¶

*Department of Software Engineering and Game Development, Kennesaw State University, USA
†Institute for Cybersecurity Workforce Development, Kennesaw State University, USA
‡Department of Information Technology, Kennesaw State University, USA
§Department of Computer Science and Software Engineering (CSSE), Auburn University, USA
¶Department of Computer Science, Tuskegee University, USA

{*mhossa21, mtasnim1*}@students.kennesaw.edu, {*hshahria, mvalero2*}@kennesaw.edu,
{*akond*}@auburn.edu & {*fwu*}@tuskegee.edu

***Abstract*—Software supply chain attacks occur during the processes of producing software is compromised, resulting in vulnerabilities that target downstream customers. While the number of successful exploits is limited, the impact of these attacks is significant. Despite increased awareness and research into software supply chain attacks, there is limited information available on mitigating or architecting for these risks, and existing information is focused on singular and independent elements of the supply chain. In this paper, we extensively review software supply chain security using software development tools and infrastructure. We investigate the path that attackers find is least resistant followed by adapting and finding the next best way to complete an attack. We also provide a thorough discussion on how common software supply chain attacks can be prevented, preventing malicious hackers from gaining access to an organization's development tools and infrastructure including the development environment. We considered various SSC attacks on stolen code-sign certificates by malicious attackers and prevented unnoticed malware from passing by security scanners. We are aiming to extend our research to contribute to preventing software supply chain attacks by proposing novel techniques and frameworks.**

***Index Terms*—Software Supply Chain Attacks, Cybersecurity, Software Security, Software Reusability**

I. Introduction

The software supply chain (SSC) is a complex, multi-faceted network of companies, individuals, and organizations involved in developing and distributing software products with a desire to develop software products predominantly through the composition of existing software components [1]. SSC plays an important part in software applications' development, distribution, and consumption worldwide and becoming increasingly important due to the increasing complexity of software development projects over the years. With the help of technological advancement, organizations are highly interconnected with product development and services across the world, software is playing a major role in diversifying operations. Particularly, focusing on software development, open-source software is everywhere, and the reusability of developed code is a new trend that eases the development process easier [2].

While software supply chain security refers to securing the components, activities, and practices involved in the creation and deployment of software by the stakeholders. SSC includes open source, third-party and proprietary code, developer practices, development frameworks, tools, deployment methods, infrastructure, interfaces, and protocols [3]. Organizations are responsible for protecting the software supply chain from malicious attacks by performing security activities, often providing security efforts to stakeholders [4]. The SSC security aims to prevent unauthorized access to or modification of software products during their development or final delivery.

Securing software supply chain is a challenge and vulnerabilities in SSC can exponentially increase the damage that hackers can cause. Adopting a comprehensive and novel set of security and quality tools for detecting and preventing SSC attacks and compliance issues in the proprietary program, open source, and third-party dependencies or even deployment configurations can be the solution [5]. The fundamental priority for the organization should be to improve security through every stage of the software development life cycle (SDLC) including initiation, development, configuration/deployment, operations/maintenance, and disposal [6]. Evidence of SSC risks must be gathered to determine whether the risks have been appropriately mitigated. Scores of U.S. institutions and agencies including the presidential office and the national institute of standards and technology (NIST) have emphasized addressing the security risks throughout the software supply chain indicating the level of necessity to address the security risk in SSC [7].

Software supply chain Security is growing as the software industry grows and becomes more complex which helps organizations and companies in daily operations by automating the creation, distribution, and use of software in business processes. Therefore the focus should be on addressing the security concern for the software supply chain by providing

978-1-6654-7679-9/22/$31.00 ©2022 IEEE
DOI 10.1109/ISSREW55968.2022.00081

infrastructure and policy to protect, deterrence, prevention, and respond to the SSC threats for organizations and businesses. The primary contributions of the paper are as follows:

- We study the existing approaches, frameworks, and applications of software supply chain security.
- We discuss the opportunities to improve existing applications and processes to mitigate SSC threats.
- We provide suggestions on how organizations may utilize novel techniques to prevent, detect, assess, and remediation of software supply chain security incidents.

The rest of this paper will be organized as follows: In Section II, we discuss the related work about software supply chain security. Section III provides a brief description of the software supply chain with applications, frameworks, and security breaches on SSC security. While Section IV presents scores of software supply chain attacks and Section V discusses the opportunities to improve the current framework, processes, and suggestions to utilize resources to prevent, detect, and remediation of software supply chain security. Section VI concludes the paper.

II. Related work

The software engineering institute at Carnegie Mellon University published a white paper on software supply chain security [6]. In the paper, SEI studies SSC security risks from an evaluation and mitigation perspective. The researchers also define SSC and analyze its security risk supply and address its focus that including minimizing opportunities for unauthorized changes and having appropriate methods for gaining confidence that such opportunities have been minimized, particularly by lower-level participants in the supply chain. A reference model was introduced that helps suppliers and organizations to follow practices that reduce SSC security risks.

The Linux Foundation studies open-source software supply chain security by prioritizing the open contribution framework [8]. SSC was defined as a massively complex mechanism that covers how software store, retrieve, and analyze the implicate additional factors along with the software deliberation using various medium. The paper emphasized OSS that is primarily written by individuals or companies or even distributed developers. The paper shows how malicious code can be distributed using the software supply chain.

Vdoo, a company that works on software security has published a practical guide on the software supply chain to understand and prevent attacks. The author provides a statistic that indicates almost 99% of software products use open source software [9]. According to the paper, organizations that utilize open source or third-party software components should focus on the provenance and security of the code added to the products because end-users little about whether to exploit through vulnerability that the developing team programmed the software.

A number of US federal agencies including the national institute of standards and technology (NIST) and office of the national counterintelligence executive (NCSC) published articles to address software supply chain security [10], [11]. The agencies introduced a Cyber Supply Chain Risk Management (C-SCRM) and Secure Software Development Framework that provides recommendations on how to manage supply chain risk. The articles also provide details on how SSC attacks can target software products in various development lifecycles. Interestingly, most of the attacks on SSC were listed with descriptions including incidents, entry points, compromised stage, affected software, and impact. The agencies also describe the type of SSC attacks including hijacking updates, undermining code signing, and compromising open-source code.

III. Software Supply Chain

Considering the traditional supply chain, a number of differences can be noted in the software supply chain, Fig. 1. Fig. 2 explains the characteristics of the software supply chain and challenges comprising distributed, assembled, siloed data, and stateful characteristics. Intangibility for instance where SSC is an intangible object made up of a series of virtual and digital components, making it very challenging to identify, quantify, and comprehend how they work from beginning to end. Besides, mercuriality is another difference where more teams reduce their time to market and it is realistic to anticipate that a software supply chain's form and organization may evolve over time. On the other hand, iterative reuse is another key aspect of SSC that allows developers or organizations to reuse the existing code or dependencies effectively.

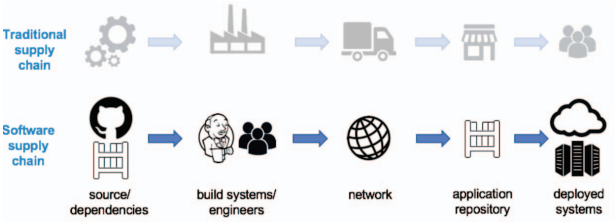

Fig. 1. illustrates the flow of software supply chain and traditional supply chain

From the beginning of 2020, organizations have experienced a tremendous paradigm shift switching to remote work. This situation has led companies to exponentially increase their Cloud technologies access, which includes less visibility into the security ecosystem, less control of access points, and a varied attack surface for adversaries to target. In Spring 2020, a Texas-based company called SolarWinds, which supports its clients by supplying software called Orion to monitor and manage IT networks and data visualization made a software update available to its eighteen thousand customers [13]. The update which seemed routinely at the beginning was used by hackers as a vehicle for a massive cyberattack against America [14], [15]. The lessons learned from this attack include the increasing problem with software supply chain (SSC) security and the enormous monetary losses that it

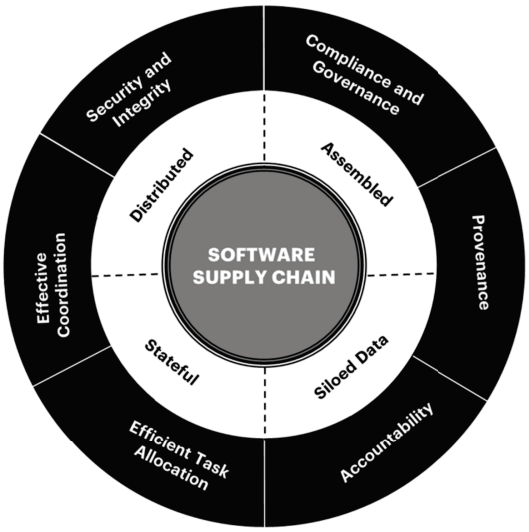

Fig. 2. characteristics of software supply chain and challenges [12]

generates, for instance, stocks of Yahoo dropped 5% after the 2016 SSC attack [4], and Equifax reported a 31% drop after 2017 SSC attack [5], and even Marriot had 6% loss after SSC attacks [6]. After the SolarWinds hacks, SSC security gained national visibility and political interest as everybody is worried about enemies who might use current trusted parties to exploit the nationâs systems and networks.

Software supply chain (SSC) security is considered as poor being the cause of more than 50% of cybersecurity attacks. This kind of "one-target, multiple-victims" scenario has turned SSC attacks into a lucrative business model for hackers, particularly when coupled with ransomware. According to the National Institute of Standards and Technology (NIST), breaches in SSC security tend to emanate more often from poor human decisions than technical failure. The security in this area highly depends on SSC stakeholders relying on implementing secure practices. The SSC workforce of the future needs to be capable of systematically implementing various aspects of cybersecurity and artificial intelligence (AI) such as secure intelligent automation, cyber-physical systems security, digital transformation, supply-chain security analytics, and more.

A. Application Platforms of Software Supply Chain Security

The application of software supply chain security is a vital part of the modern world and the software supply chain entails the process that involves all stages of production, distribution, and consumption of software. The function of this process is to ensure that products are produced according to specific requirements and standards. It also ensures that products are delivered on time to meet customer demands [16]. Notably, supply chain security is important in an online business model, where many third parties are involved. This increases the risk of supply chain attacks because this model has different points of failure. In addition, if any of those companies were compromised or hacked, it could lead to major damage for businesses and individuals.

The security threats are both external and internal. The external threats include theft, damage, or loss of goods at the end of the supply chain. These can be caused by human error or natural events such as floods, earthquakes, fires, hurricanes, and other weather events. Internal threats include data breaches that occur when hacking or cyberattacks compromise sensitive information stored on computers or networked devices. This can include customer credit card information or intellectual property information such as blueprints or manufacturing processes [17]. Moreover, when it comes down to logistics and transportation, many ways can increase the security of supply chains and processes involved therein, such as using satellite tracking systems on every shipment sent out by trucking companies to keep track of every single shipment as well as its location at all times so that it can be tracked easily if something goes wrong during transportations or if someone tries to steal something from within one's company facilities.

In the modern era of technology, there is a great deal of emphasis on data security as it is crucial for business growth. It can be said that data security is one of the most important aspects of any business [18]. If a company does not have strong data security policies and practices, it may experience some significant risks. The software supply chain is a major component of information security. It is a chain of software development, quality assurance, and distribution activities that involve creating, testing, and maintaining applications. The first step in the supply chain is creating a new application by an organization. Once created, this application is enhanced by adding functionality and improving performance through updates.

Quality assurance personnel typically test these updates before being distributed to end-users for deployment on the production network. Once deployed to production environments, applications are subject to attacks from hackers looking to steal sensitive data or change system settings without authorization. Organizations have kicked off these attacks by bypassing firewalls and intrusion detection systems (IDSs) with port scanning tools. Since these tools do not require authentication or authorization to run on a target host, they can be used as a stealthy means of compromising vulnerable systems.

B. Security Breaches in The Software Supply Chain

Security threats include theft, damage, or loss of software products at the end of the supply chain. Data breaches occur when hacking or cyberattacks compromise sensitive information stored on computers or networked devices. This can include customer credit card information or intellectual property information such as blueprints or manufacturing processes [4]. A software supply chain is a series of steps where one company provides software to another company. The software supply chain is the mechanism by which software is transferred from its creation to deployment. The people who control this chain are uniquely positioned to influence software

quality and security. The nature of the supply chain makes it difficult for developers and testers to detect and prevent vulnerabilities before they're introduced into production environments. Vulnerabilities can be introduced at any point along the supply chain: from the developer creating their software to a third-party supplier who integrates it with a business application that runs on the business's infrastructure.

The degree to which these vulnerabilities can impact operations varies depending on where in the supply chain they were introduced and what type of controls exist within that environment. In many cases, organizations have controls that prevent any code from being added or modified without explicit approval from management or legal counsel. Still, these controls are often ineffective because they focus on preventing unauthorized changes rather than ensuring only authorized changes to occur [19]. Security threats in the software supply chain have been a major challenge to software vendors, service providers, and end-users. The threat of cybercrime is increasing and becoming more difficult to detect and mitigate. As technology advances and becomes more complex, so too do the types of cybercrimes that occur. Hackers can compromise code repositories or source code management systems used by developers to gain access to sensitive data and software source codes [20]. An attacker may also find vulnerabilities in third-party libraries used by developers or even hijack entire applications before releasing them on public platforms such as Google Play Store or Apple App Store. Cybercriminals can exploit security flaws in penetration testing tools and automated vulnerability scanners designed for penetration testing purposes during software testing phases [5]. They can also use automated scripts that scan websites for vulnerable plugins, such as cross-site scripting (XSS) vulnerability scanners.

There are two kinds of malicious code in the software supply chain, first is known as malicious code, and it is used to cause damage or harm to a system or user. Malicious code can be spread through infected files, emails, and other means. It can also be introduced into a system via malware that infects a computer with a virus. The second type of malicious code is known as a back door which refers to a method used by hackers to gain access to an organization's network without permission from the owner and users. This malicious code can be installed on computers to gain unauthorized access to an organization's network, including internal web pages and email accounts. Compliance with Legal Requirements; In addition to having processes in place to address vulnerabilities, there should also be policies regarding compliance with legal requirements. These requirements include encrypting data at rest, storing encryption keys on an isolated system, and properly handling them when they are needed for authentication purposes [21]. Everyone who has access to any system should be authenticated so that only those who should have access can access it.

IV. Software Supply Chain Attacks

Software development processes with modern technology follow open-source approaches and this is where the software supply chain emerges that refers to anything that involves and affects software that was neglected precisely for decades. Today's software engineers do not develop software products from the ground up, rather relying on third-party resources, pre-built libraries, or open-source codes in proprietary codebases and community projects including Github, Docker Hub, and StackOverflow [22]. The process of ensuring the security of the software supply chain is exhausting and complicated that lead to necessitates the attention of the software community.

There are some common attacks that occur in software supply chain, dependency confusion or typosquatting, and malicious source code injection for instance depicted in Fig. 3. A number of attacks were recorded including Solarwinds, CodeCov, Passwordstate, SITA, Kaseya, Apple Xcode, and XcodeGhost. Attackers hunt for vulnerable network protocols, unsecured server infrastructures, unsafe coding practices, change source codes, and hidden malware in build and update processes by malicious attackers. The source code is the underlying architecture of any software supply chain and establishing and verifying the source code's integrity is the first step in securing a supply chain [23]. In the context of "integrity", it refers to the transparency of the source code and repository that shall be identical to the original codes written by developers and have never been altered. Organizations must take efforts to validate the source of code introduced to their first-party products and libraries in order to ensure the integrity where any malicious changes made on the developer's system as a result of a local intrusion shall be addressed.

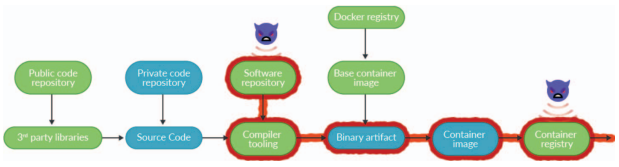

Fig. 3. Examples of software supply chain attacks

Software supply chain attacks are gradually increasing parallelly with the advancement of technology and the most common being related to dependency confusion or typosquatting, followed by malicious source code injection [24]. The ever-increasing complexity of software, infrastructure, and dependencies is making organizations and software engineering community increasingly worried about its security. Software supply attacks occur during the processes of producing software are compromised, resulting in vulnerabilities that target downstream customers. While the number of successful exploits is limited, the impact of these attacks is significant. Despite increased awareness and research into software supply chain attacks, there is limited information available on mitigating or architecting these risks, and existing information is focused on singular and independent elements of the supply chain. As the field matures in response to adversarial tactics, a more holistic approach is needed. The following modules can be developed:

- SSC attacks using software development tools and infrastructure [25]: During software supply chain attacks, attackers prefer the path that is least resistant followed by adapting and finding the next best way to complete an attack. One of the common software supply chain attacks happens when malicious hackers gain access to an organization's development tools and infrastructure including the development environment. The organization shall need to consider providing sufficient training to employees or project teams to maintain used software tools and infrastructure.
- SSC attacks on stolen code-sign certificates [25]: Stealing code signing certificates by malicious attackers is one of the common issues in software supply chain attacks that help malware pass unnoticed by security scanners. Attackers acquire code-signing certificates including trusted machine identities from various sources and organization's networks for instance that leads the system vulnerable to malware attacks. Often the computer, mobile or IoT devices look for a code signing certificate to ensure the software runs is trusted and malicious attackers take advantage of this trust system to spread malware that appears trusted because it is legitimately signed by code signing certificates [26].
- SSC attacks by infiltrating a software vendor's network [27]: Most modern software receives routine updates to improve the features, user experience, address bugs and security issues where software vendors typically distribute updates from centralized servers to customers as a routine part of product maintenance. Malicious attackers can hijack an update by infiltrating the vendorâs network and either inserting malware into the outgoing update or altering the update to grant the threat actor control over the softwareâs normal functionality. In 2017, Russian hackers launched the NotPetya malware that spread across the globe and caused disrupted critical businesses internationally.
- SSC attacks by employing malicious code in product deliverable [27]: A software supply chain attack occurs when open-source code compromises threat actors by inserting malicious code into publicly accessible code libraries. Software developers utilize free blocks of code from third-party platforms including GitHub to perform specific functions for their own projects. However, malicious libraries can contain the same code and functionality that developers impersonated; however, the code also can contain additional functionality, the ability to obtain boot persistence and open a reverse shell on remote workstations for instance. Because proprietary code authors frequently incorporate blocks of open-source code in software products, malicious code can disrupt the software supply chain, resulting in severe ramifications for government, critical infrastructure, and private sector software users.
- SSC attacks by delivering viruses and malicious software via suppliers or vendors [28]: Viruses or malicious software can also occur through a software provider or vendor that can lead to a software supply chain attack. For instance, a keylogger hidden on a USB stick may infiltrate a huge retail organization, logging keystrokes to determine passwords for specific accounts, and allowing hackers to access valuable corporate information, customer data, payment information, and more.

V. Discussion

Software supply chain security is the process of protecting the integrity of software. It involves various software development, testing, and deployment levels to ensure that the product is delivered in its final form. The most common method for ensuring the integrity of software supply chains is through a combination of quality assurance (QA), software development life cycle (SDLC), software supply chain management system, and gatekeeper. The software supply chain management system is the backbone of software development, testing, and deployment. It provides a single-source view of the entire development process and enables organizations to track all aspects of a project from inception to delivery [2]. This signifies that before a product goes into production or through testing, it has been assessed by an independent party to ensure it meets all requirements. The QA should ensure that any changes are done correctly and without introducing any bugs or issues into the final product. It is a vital part of software supply chain security and helps in testing the product before its release to ensure that it meets all the quality standards the customer sets. It is also used to detect any bugs that may be present in the product. QA teams will also perform code reviews and test scripts for customers and vendors to ensure no errors or issues with their software products [12].

The SDLC follows a similar process but with more control over each stage by adding controls such as automated testing tools and version control systems for managing changes to code. A good SDLC will reduce errors during development and allow for rapid updates if needed; it is an enterprise-wide approach to managing all aspects of software projects from conception to delivery, including requirements analysis and traceability, design, coding, testing, and quality testing. Significantly, the software development life cycle involves designing, developing, testing, and deploying a software system through multiple steps. It is divided into seven phases: initiation, analysis, design, coding, testing, and deployment. These phases are further broken down into sub-phases based on the type of products being developed. Each phase has unique responsibilities which help ensure quality control at every stage of development. In the context of information security, a gatekeeper is a person who controls access to resources such as computer systems or networks by deciding who may enter and exit those resources.

A software supply chain management system has been designed to help the organization manage and monitor its supply chain effectively. Through this, the business can better understand the current status of its products and services, thus making it easier for them to determine what needs

to be done in terms of quality control. It also helps in identifying opportunities and risks when it comes to managing and monitoring the supply chain [5, 10]. In addition, it allows businesses to easily identify potential problems, which can then be fixed before they become bigger problems. The main objective of this software is to provide complete visibility into the whole supply chain by integrating various tools and data sources. This helps effectively manage all activities related to the software product life cycle. Furthermore, this software helps companies control costs by ensuring that they are only spending money on necessary things and not on overheads or unnecessary expenses. Gatekeepers are usually responsible for protecting their organizations against threats from inside or outside their organizations. The Gatekeeper will perform tests on each component or functionality as part of its job description. It will report any problems back to developers who may be working on those particular components or functionality so that they can fix them before releasing them into production.

VI. Conclusion

Security is a key concern in the software supply chain that can be vulnerable to exploitation by cybercriminals throughout the software development lifecycle has several stages. Often one can rely on open source or third parties code and dependencies to complete software products and organizations or developers must ensure the security concern in protecting and securing the software supply chain from malicious attacks. In this paper, we addressed the key issues related to software supply chain security to better understand the importance of software supply chain security for an organization and how to implement best practices in this area. We also studied existing approaches, frameworks, and applications to ensure software supply chain security. Discussion on overcoming security issues in the software supply chain would help organizations and developers prevent, detect, assess, and remediation of SSC security concerns.

Acknowledgment

The work is partially supported by the U.S. National Science Foundation Awards 2209638, 2209636, and 2209637. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] G. C. Murphy, "Software supply chains (keynote)," in *2015 ACM/IEEE 18th International Conference on Model Driven Engineering Languages and Systems (MODELS)*, 2015, pp. 2–2.

[2] N. Zahan, T. Zimmermann, P. Godefroid, B. Murphy, C. Maddila, and L. Williams, "What are weak links in the npm supply chain?" in *2022 IEEE/ACM 44th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, 2022, pp. 331–340.

[4] W. Enck and L. Williams, "Top five challenges in software supply chain security: Observations from 30 industry and government organizations," *IEEE Security Privacy*, vol. 20, no. 2, pp. 96–100, 2022.

[3] Y. Ma, "Constructing supply chains in open source software," in *2018 IEEE/ACM 40th International Conference on Software Engineering: Companion (ICSE-Companion)*, 2018, pp. 458–459.

[5] D. Yan, Y. Niu, K. Liu, Z. Liu, Z. Liu, and T. F. Bissyandx00E9;, "Estimating the attack surface from residual vulnerabilities in open source software supply chain," in *2021 IEEE 21st International Conference on Software Quality, Reliability and Security (QRS)*, 2021, pp. 493–502.

[6] R. J. Ellison, J. Goodenough, C. B. Weinstock, and C. Woody, "Evaluating and mitigating software supply chain security risks," 2010.

[7] Y. Matsuno, Y. Yamagata, H. Nishihara, and Y. Hosokawa, "Assurance carrying code for software supply chain," in *2021 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, 2021, pp. 276–277.

[8] L. Foundation, "Open source software supply chain security," 2020.

[9] Vdoo, "The practical guide for understanding and preventing software supply chain attacks," 2021. [Online]. Available: https://www.energy.gov/sites/default/files/2021-06/Shane%20DeLair%20Vdoo-A1.pdf

[10] N. I. of Standards and Technology, "Defending against software supply chain attacks," 2021.

[11] N. C. Executive, "Software supply chain attacks," 2021.

[12] K. Singi, J. C. B. R P, S. Podder, and A. P. Burden, "Trusted software supply chain," in *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2019, pp. 1212–1213.

[13] F. Massacci, T. Jaeger, and S. Peisert, "Solarwinds and the challenges of patching: Can we ever stop dancing with the devil?" *IEEE Security Privacy*, vol. 19, no. 2, pp. 14–19, 2021.

[14] M. Willett, "Lessons of the solarwinds hack," *Survival*, vol. 63, pp. 7 – 26, 2021.

[15] G. Wolff, W. Lerner, and C. Gruden, "Navigating the solarwinds supply chain attack," *The Procurement Lawyer*, vol. 56, p. 3â11, 2021.

[16] V. Hassija, V. Chamola, V. Gupta, S. Jain, and N. Guizani, "A survey on supply chain security: Application areas, security threats, and solution architectures," *IEEE Internet of Things Journal*, 2021.

[17] C. Cao, S. Cecilia, H. Chia, M. C. Chou, C. Tan, M. T. Teh, S. M. Sim, H.-q. Ye, and X.-m. Yuan, "Key issues of a software focused supply chain," in *2006 4th IEEE International Conference on Industrial Informatics*, 2006, pp. 747–752.

[18] M. Ohm, A. Sykosch, and M. Meier, "Towards detection of software supply chain attacks by forensic artifacts," *Proceedings of the 15th International Conference on Availability, Reliability and Security*, 2020.

[19] Y. Hou, J. Such, and A. Rashid, "Understanding security requirements for industrial control system supply chains," in *2019 IEEE/ACM 5th International Workshop on Software Engineering for Smart Cyber-Physical Systems (SEsCPS)*, 2019, pp. 50–53.

[20] K. Garrett, G. Ferreira, L. Jia, J. Sunshine, and C. KÃ€stner, "Detecting suspicious package updates," in *2019 IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER)*, 2019, pp. 13–16.

[21] R. A. Martin, "Assurance for cyberphysical systems: Addressing supply chain challenges to trustworthy software-enabled things," in *2020 IEEE Systems Security Symposium (SSS)*, 2020, pp. 1–5.

[22] M. Kaczorowski. (2020) Secure at every step: What is software supply chain security and why does it matter? [Online]. Available: https://github.blog/2020-09-02-secure-your-software-supply-chain-and-protect-against-supply-chain-threats-github-blog/

[23] C. N. C. Foundation. (2020) Software supply chain best practices.

[24] A. Iradier, "Secure software supply chain: why every link matters." Sysdig, Inc., 2021. [Online]. Available: https://sysdig.com/blog/software-supply-chain-securit

[25] B. Woodbury, D. Simpson, D. Halfin, and A. M. Gorzelany, "Supply chain attacks." Microsoft, 2021. [Online]. Available: https://docs.microsoft.com/en-us/windows/security/threat-protection/intelligence/supply-chain-malware

[26] C. S. Writer, "Stolen code-signing certificate used in malware attack." New Statesman Media Gr. Ltd., 2018. [Online]. Available: https://techmonitor.ai/techonology/cybersecurity/eset-internet-security-firms

[27] Hyperproof, "Defending against software supply chain attacks," 2021. [Online]. Available: https://hyperproof.io/resource/software-supply-chain-attacks

[28] K. Security, "What is a supply chain attack? keeperâs guide to supply chain cyberattacks." [Online]. Available: https://www.keepersecurity.com/threats/supply-chain-attack.html