Proposing a Zero Trust and Blockchain Approach to Tackle Software Supply Chain Attacks

Ripunjay Singh[1], Aspen Olmsted[2]
New York University, New York
Wentworth Institute of Technology, Boston
USA

Abstract

This paper will discuss some statistics related to software supply chain attacks and understand their effects in today's technology landscape. We will focus on newer technologies like Zero Trust and Blockchain, which can help safeguard organizations against such attacks. The paper will develop on previous research and emphasize creating a Zero Trust and Blockchain deployment, allowing organizations to establish a decentralized, immutable, and secure software supply chain.

1. Introduction

According to the Cybersecurity and Infrastructure Security Agency, a software supply chain attack occurs when a cyber threat actor infiltrates a software vendor's network and employs malicious code to compromise the software before the vendor sends it to their customers. Software supply chain attacks are increasingly becoming a significant security concern for the modern software supply chain. Bad actors have become increasingly sophisticated in their attacks, demanding more innovative and robust security measures to tackle these modern-day nuisances.

A Zero Trust Architecture allows organizations to emphasize the principle of "never trust, always verify," taking a step back from unquestioningly trusting all supply chain actors. Zero Trust minimizes the attack surface by continuously validating user identities, devices, and applications. Moreover, Blockchain technology will allow for a decentralized, immutable infrastructure. The distributed ledger system will ensure that the software artifacts are not tampered with, making it effective in verifying the authenticity and integrity of each component in the supply chain cycle. This paper will explore how combining these two measures can allow organizations to safeguard themselves and defend against software supply chain attacks.

The paper has been divided as follows: Section 2 will discuss related research to combating software supply chain attacks and what methods exist today. Section 3 will take the reader through a motivating example. Section 4 will discuss the hypothesis, while Section 5 will discuss the proposed solution of how implementing a Zero trust and Blockchain solution can help protect organizations against software supply chain attacks. Section 6 will depict a reference architecture, along with the flow of information. The discussion will conclude with Section 7, describing some future ideas.

2. Related Research

The SolarWinds attack in 2020 is by far the most known supply chain attack. SolarWinds's network management system, Orion, was jeopardized by attackers and then used to acquire and steal sensitive data from over 500 customers [3]. In September 2017, attackers compromised Avast's CCleaner tool, which hackers then used to target significant technology and telecommunications companies worldwide with a second-stage payload [1]. These are just examples of how organizations have been affected by software supply chain attacks.

Organizations face several challenges in tackling software supply chain attacks. The first challenge is prevention. Currently, security measures only protect against untrusted sources, but they are ineffective against supply chain attacks that exploit customers' inherent "Trust" in chosen suppliers [2]. The second challenge is detection. Since contemporary systems are enormously large and have a sophisticated dependency on third-party software components, a cyber defense mechanism can't know the inner functionality of all components [2].

Some researchers have come close to devising ways or best practices to tackle supply chain attacks. Jefferson Martinez et al. mention various good practices that can be used, starting with Software Billing of Materials (SBOM). SBOM allows the builder to ensure the software components are up to date and respond quickly to new vulnerabilities. Other ways include using Multi-Factor Authentication, ML algorithms to analyze anomalies in data flows, etc [5].

Udit Agarwal et al. explored using blockchain in the supply chain through a literature review. Their research depicted that blockchain is helpful in the supply chain in four major areas: (1) Transparency and traceability, (2) Information sharing, (3) Product Anti-counterfeiting, and (4) Build Trust [3].

Muhammad Zeeshan Malik et al. devised a novel

 | DOI: 10.20533/jitst.2046.3723.2023.0100

solution called "Validation" that will be executed on a user's machine to test the legitimacy of update techniques such as honeytokens, privileged pathways, and data loss ratio or any specific algorithm implemented by end users [6].

Another study investigated using Zero Trust for supply chain attacks. Thiago Melo Stuckert do Amaral et al. proposed integrating Zero Trust in the cyber supply chain in a series of steps: (1) identifying the critical components of the cyber supply chain, (2) checking adherence to the Zero Trust principles, which will enable a gap analysis for the implementation of the controls, and (3) identifying the design of a roadmap of security improvements [7]. The authors also provide tools that facilitate this integration and implementation.

Although the above works have focused separately on Zero Trust and Blockchain to safeguard against software supply chain attacks, this research paper takes a step further and studies how the deployment of a combined Zero trust and blockchain solution can be fruitful in preventing and detecting software supply chain attacks, thereby allowing for the confidentiality, integrity, and availability of the various involved components.

3. Motivating Example

Software supply chain attacks have increasingly become a nuisance to organizations, evidenced by the statistics and the scale of supply chain attacks that have affected organizations in the last few years. Sonatype reported in its State of the Software Supply Chain report that software supply chain attacks have jumped drastically to 742% from 2019 to 2022. Capterra found that a software supply chain threat over the past year has directly impacted three-fifths (61%) of US businesses.

4. Hypothesis

The software supply chain has long suffered supply chain attacks, primarily due to vulnerabilities that target software and hardware components involved in the development, distribution, and maintenance. These can be attack vectors like malware injection, backdoor insertion, compromised updates, etc. These attack vectors lead to exploitation opportunities for bad actors to steal sensitive information or launch further attacks. A compromised software supply chain can affect the security of entire systems and the users.

Zero Trust provides a strict security model by implementing strong authentication and access controls and monitoring using a Policy Enforcement Point (PEP) to reduce tampering within the software supply chain. Blockchain provides an immutable and decentralized component to the framework, ensuring increased integrity of software components. Previous related research has utilized the efficacy of these measures individually. In this paper, we hypothesize that integrating these two measures to establish a resilient software supply chain ecosystem will reduce the impact of supply chain attacks.

In this conceptual research, we propose a real-world deployment designed to assist organizations in mitigating software supply chain attacks. In this approach, a web application will be deployed to serve as the central hub through which all components of the supply chain ecosystem pass.

At a high level, the application will be designed to enforce a zero-trust architecture, with the Policy Enforcement Point (PEP) conducting security checks on each component, guided by the NIST 800-161 framework. The PEP will make informed decisions to allow or deny access to the user's computer, ensuring that trust is not granted solely based on the component's source. Simultaneously, the blockchain system will be pivotal in storing critical component data.

Our approach involves a two-phased evaluation to assess the effectiveness of this deployment: Phase 1 logs incidents without the deployment and Phase 2 logs incidents with the deployment in place. This structured assessment will enable us to gauge the impact of our proposed solution.

5. Proposed Approach

The proposed solution has the following components:

1. A web application platform with the following parts:

   Web interface: The web interface is the component that facilitates the user interaction with the software websites to request updates and patches.

   Blockchain: The blockchain stores critical information about the software components. We advise using Ethereum, given its widespread use and standardization.

   Policy Enforcement Point (PEP): The PEP serves the dual role of authenticating and authorizing user access to the web application and verifying software updates against the NIST SP 800-161 controls.

   Below is a depiction of a PEP based on the NIST SP 800-207 [9]. Note that although all the traffic passes through the PEP in the Data Plane, the actual decision to allow/deny access is made by the Policy Engine and the Policy Administrator,

 | DOI: 10.20533/jitst.2046.3723.2023.0100

collectively called the Policy Decision Point or PDP, which resides in the Control Plane. We will not be going deeper into the model in this paper (see Figuer 1).

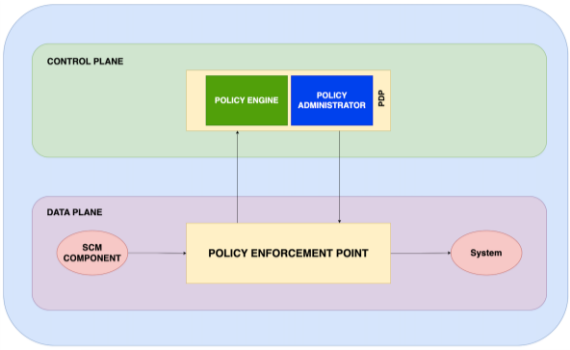

Figure 1. A depiction of a Policy Enforcement Point

2. To bolster overall security, we propose integrating additional tools for analyzing updates, identifying malware, and detecting vulnerabilities. The proposal includes SIEM/SOAR, Threat Intelligence, NSM, and DLP tools.

3. The entire approach unfolds in three interactions (see Figure 2):

   a. The software supply chain and the web application

   b. The PEP and the blockchain

   c. The user and the web app

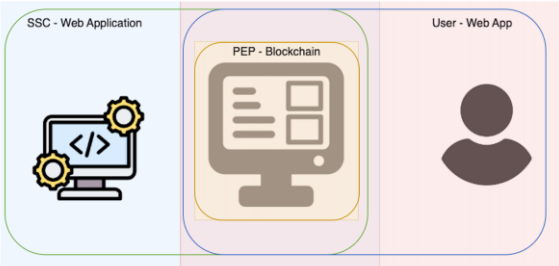

Figure 2. Interactions in the proposed model

The NIST SP 800-161 is a unique publication that provides cybersecurity supply chain risk management practices for systems and organizations. It guides managing the security of the supply chain for federal information systems. The special publication includes a set of security controls that will be included to enhance the proposed approach. Some of these controls include Access Control, Baseline Configuration, Identity and Access Management, etc [8].

Figure 3 shows a blockchain diagram that will be a crucial part of the information flow. The blockchain will store critical data about the software and its updates/patches, allowing integrity, transparency, and immutability.

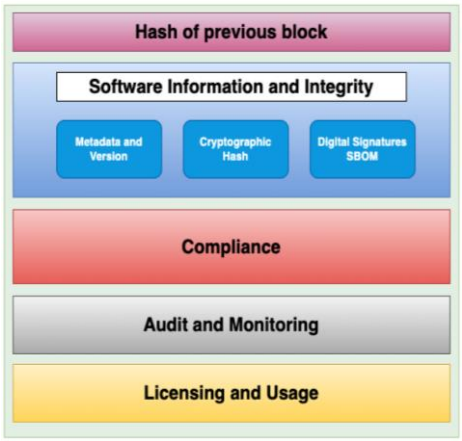

Figure 3. Diagram depicting the blockchain structure

1. The Hash block of the previous block ensures the continuity and integrity of the blockchain.

2. Software Information and Integrity Block contain crucial data about the software, including details such as metadata, cryptographic hash, and the Software Billing of Materials (SBOM).

3. The Compliance Block stores information regarding the access controls implemented for each software update, specifying which user can download a software update (e.g., admin, employees at different clearance levels, etc.).

4. The Audit and Monitoring block captures event logs and records all interactions between each software update. In the future, these logs can be a crucial part of risk assessment and provide insights to security teams about potential security threats.

5. The Licensing and Usage Block contains licensing information, ensuring that each software update is used in compliance to prevent legal troubles.

6. Reference Architecture

The Figire 4 shows a reference architecture for the proposed approach, along with a description of the flow of information.

STEP 1
The flow of Information: System (User) -> Web Interface:

 | DOI: 10.20533/jitst.2046.3723.2023.0100

Description: The user requests a software update through the web application and undergoes access control evaluation like Multi-Factor Authentication (MFA) to ensure the user claims who they are.

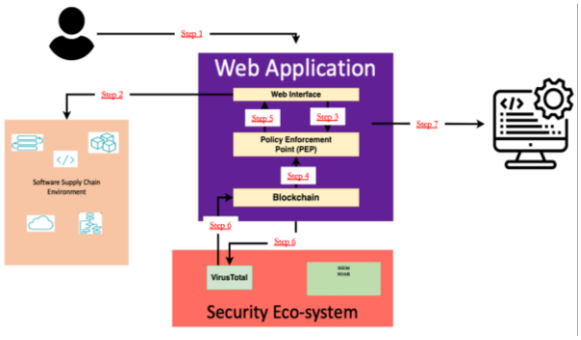

Figure 4. Reference Architecture of the Proposed Approach

STEP 2
The flow of Information: Web Interface -> Software Website:

Description: The Web Application receives the download request and is the central component for managing the software update process.

STEP 3
Flow of Information: Web Interface -> Policy Enforcement Point (PEP):

Description: The Web Application forwards the request to the PEP, which then runs authentication and authorization checks, as well as an assessment of the software update against the NIST 800-161 controls.

STEP 4
Flow of Information: Policy Enforcement Point (PEP) <-> Blockchain:

Description: In the meantime, the PEP interacts with the blockchain to confirm the authenticity and integrity of the software update. It checks metadata, hashes, digital signatures, Software Bill of Materials (SBOM), audit and monitoring data, and license information recorded on the blockchain.

STEP 5
Flow of Information: Policy Enforcement Point (PEP) -> Web Interface:

Description: Based on the rigorous evaluation, the PEP makes one of three **initial decisions** and communicates it to the web interface:

1. Rejected: The software update doesn't meet the guidelines and controls and cannot be downloaded.
2. Flagged Suspicious: Further security checks needed.
3. Accepted: The software update is ready to be implemented

STEP 6
Flow of Information:

1. Software Update -> VirusTotal
2. VirusTotal -> PEP
3. VirusTotal -> Blockchain

Description: After the initial decision, security tools such as VirusTotal can be used to run scans on the software update and perform real-time analysis to identify any malware or security threats. The PEP receives the VirusTotal results and considers these findings when making **its final access control decision**. The scan results are also stored in the Blockchain for future use.

STEP 7
The flow of Information: Software Update -> System:

Description: The software update is downloaded and ready to use by the user.

Note: that after a file is downloaded, and if organizations wish to add an extra layer of security, they can monitor the behavior of the flagged or accepted software updates via SIEM/SOAR, EDR tools, etc. The main difference between "Flagged" and "Accepted" is that it provides users with a clear understanding of what types of updates/patches to watch and do more security analysis for.

7. Conclusion and Future Work

This research paper has focused on addressing the growing threat of attacks on the software supply chain. It proposes a novel approach involving the integration of a robust Zero Trust and Blockchain application reinforced by comprehensive, standardized policies for the PEP. This framework aims to enhance the security and resilience of the software supply chain, promoting decentralization and immutability. Currently, a limitation we face is the lack of an actual real-world deployment that can assist us in understanding whether the proposed approach is pragmatic and successful.

Given the widespread use and scalability, our future work will build upon these foundations, specifically focusing on developing a real-world on-premises web application deployment and potentially a cloud-based deployment moving forward. Furthermore, we intend to address critical metrics,

 | DOI: 10.20533/jitst.2046.3723.2023.0100

such as time to detect and contain attacks. By continuously evolving our research, we aspire to contribute to the ongoing efforts to secure and fortify organizations' software supply chains in the face of dynamic and evolving cyber threats.

8. References

[1] Coufalikova, A., Klaban, I., & Slajs, T. (2021, June 8). Complex strategy against supply chain attacks. 2021 International Conference on Military Technologies (ICMT). DOI: 10.1109/icmt52455.2021.9502768.

[2] Wang, X. (2021, November 29). On the Feasibility of Detecting Software Supply Chain Attacks. MILCOM 2021 - 2021 IEEE Military Communications Conference (MILCOM).10.1109/milcom52596.2021.9652901.

[3] Agarwal, U., Rishiwal, V., Tanwar, S., Chaudhary, R., Sharma, G., Bokoro, P. N., & Sharma, R. (2022). Blockchain Technology for Secure Supply Chain Management: A Comprehensive Review. IEEE Access, 10, 85493–85517. DOI: 10.1109/access.2022.3194319.

[4] Gokkaya, B. (2023, May 23). Software supply chain: review of attacks, risk assessment strategies and security controls. https://arxiv.org/abs/2305.14157 (Access Date: 3 December 2023).

[5] Martínez, J., & Durán, J. M. (2021, October 31). Software Supply Chain Attacks, a Threat to Global Cybersecurity: SolarWinds' Case Study. International Journal of Safety and Security Engineering, 11(5), 537–545. DOI: 10.18280/ijsse.110505.

[6] Malik, M. Z., & Bukhari, S. Z. A. (2023, March 15). Protection Mechanism against Software Supply Chain Attacks through Blockchain. 2023 International Conference on Communication Technologies (ComTech). DOI: 10.1109/comtech57708.2023.10164932.

[7] do Amaral, T. M. S., & Gondim, J. J. C. (2021, November 18). Integrating Zero Trust in the cyber supply chain security. 2021 Workshop on Communication Networks and Power Systems (WCNPS). DOI: 10.1109/wcnps53648.2021.9626299.

[8] Boyens, J., Smith, A., Bartol, N., Winkler, K., Holbrook, A., & Fallon, M. (2022, May 5). Cybersecurity supply chain risk management for systems and organizations. DOI: 10.6028/nist.sp.800-161r1.

[9] Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020, August 11). Zero Trust Architecture. DOI: 10.6028/nist.sp.800-207

 | DOI: 10.20533/jitst.2046.3723.2023.0100