DOI: 10.1561/9781638285076.ch11

Chapter 11

Securing the Software Supply Chain: Innovations and Approaches

By Apostolis Zarras, Evangelos Haleplidis, Christos Xenakis and Apostolos Fournaris

In modern software development, applications are built using diverse components sourced externally, such as open-source libraries, cloud services, and hardware, which introduce significant security risks into the software supply chain. High-profile incidents like the Log4j and SolarWinds attacks have demonstrated the severe impact of vulnerabilities in this interconnected ecosystem. Traditional security measures focus heavily on open-source components, often overlooking hardware and firmware, which are equally susceptible to threats. This chapter introduces RESCALE, a comprehensive framework designed to address these challenges through a security-by-design approach. RESCALE integrates advanced static and dynamic security testing tools with blockchain technology, creating a Trusted Bill of Materials that provides both hardware and software components transparency, traceability, and immutability. By incorporating cutting-edge security assessments and leveraging blockchain for the secure recording of results, RESCALE ensures a holistic and resilient supply chain, safeguarding against known and emerging threats in today's complex cybersecurity landscape.

11.1 Introduction

In modern software development, applications are no longer built entirely from scratch; they consist of various components from multiple sources integrated into the *Software Development Lifecycle (SDLC)*. Many of these components are not developed in-house. Still, they are used as-is, which, while accelerating time to market by leveraging pre-existing functionality, introduces significant security risks. This software supply chain, comprised of external hardware, infrastructure, operating systems, drivers, open-source scripts, CI/CD tools, and even cloud services, is inherently complex and often untrustworthy. The traditional security assessment methods applied to in-house development must address the intricate vulnerabilities posed by this external integration.

High-profile incidents, such as the Log4j vulnerability [1] and the SolarWinds supply chain attack [2], have starkly illustrated the potential devastation of supply chain breaches. Attacking a single component in the supply chain can compromise many downstream software products, exponentially magnifying the impact. As cybersecurity experts increasingly anticipate future large-scale attacks to target the software supply chain, there is an urgent need for a holistic approach that addresses both hardware and software security [3].

Much of the attention on securing the software supply chain focuses on open-source solutions, perceived as a primary vulnerability. However, this perspective overlooks the broader scope of the problem. A comprehensive view must account for all aspects of the SDLC, including hardware and firmware, which also follow distinct supply chains. Hardware vulnerabilities, such as side-channel attacks [4] and transient execution attacks [5], as well as hardware Trojans [6], can compromise not only individual systems but also the software built upon them. The interconnectedness of hardware and software supply chains demands that security-by-design principles be applied across the entire ecosystem to mitigate risks.

One emerging solution is the *Software Bill of Materials (SBOM)* [7]. This formal document tracks the components of a software product and their supply chain relationships. SBOMs provide visibility into third-party components, particularly open-source libraries, and help organizations manage compliance and dependency tracking. Although some of the existing SBOM standards can disclose vulnerabilities, the majority fail to provide comprehensive security information for the listed components and fail to address the hardware supply chain. Furthermore, robust mechanisms are lacking to protect SBOMs from *Confidentiality, Integrity, and Availability (CIA)* attacks.

This chapter proposes RESCALE, an innovative approach to securing software supply chains through a holistic, security-by-design methodology. By integrating cutting-edge security assessment tools with blockchain technology, RESCALE aims

to create a resilient and trustworthy supply chain framework that can withstand the evolving threat landscape.

11.2 Background

The ever-evolving software and hardware development landscape presents new challenges in maintaining a secure supply chain. The potential for vulnerabilities grows as the software industry increasingly relies on third-party components, hardware integrations, and open-source solutions. These weaknesses, if exploited, can disrupt entire ecosystems, as evidenced by high-profile incidents like the Log4j and SolarWinds attacks. Ensuring the security of both hardware and software components across the SDLC is crucial. This section will explore various approaches and methodologies for testing vulnerabilities, ensuring firmware security, applying formal verification techniques, and addressing hardware vulnerabilities at the processor level for a comprehensive supply chain security framework.

11.2.1 Security Testing and Vulnerability Assessment

Security testing is essential for uncovering vulnerabilities and weaknesses within software, networks, and systems. By employing various techniques, security experts can identify potential threats, estimate the likelihood of exploitation, and evaluate the overall risk landscape of an application. For example, vulnerability scanning detects known vulnerabilities and loopholes, helping establish a security baseline. Popular tools include OpenVAS, Nessus, Burp Suite, Tripwire, Qualys, and Metasploit.

Security scanning, another critical approach, is designed to uncover misconfigurations and vulnerabilities within applications, networks, and systems. Both manual and automated tools, such as Nmap, Wireshark, and Zenmap, are commonly utilized for this type of testing. In contrast, penetration testing simulates real-world cyberattacks against an application, system, or network in a controlled environment. Conducted by certified security experts, this test is critical for identifying previously unknown vulnerabilities, including zero-day threats and business logic flaws. Tools frequently used for penetration testing include W3af, Aircrack-ng, John the Ripper, OWASP ZAP, and the comprehensive Kali Linux suite.

Once vulnerabilities are identified, security risks are classified (i.e., Critical, High, Medium, and Low) through a risk assessment, usually using the *Common Vulnerability Scoring System (CVSS)*. Tools such as SpiraPlan, A1 Tracker, RM Studio, Isometrix, and CheckIt assist in this process. Based on these assessments, mitigation strategies and security controls are prioritized and recommended.

11.2.2 Firmware Security

Firmware serves as the crucial interface between a computer's hardware and software, abstracting many low-level, hardware-specific details to enable the seamless development and execution of software across multiple systems. However, positioning firmware below the software layer poses significant security challenges, often requiring dedicated solutions. Several tools and frameworks have been developed to detect vulnerabilities at the firmware level, thereby reducing the potential attack surface for hackers.

One such solution is *Avatar* [8], an event-based arbitration framework that facilitates communication between an emulator and a physical target device. Avatar enables complex dynamic analysis of embedded firmware, supporting various security tasks such as reverse engineering, malware analysis, vulnerability discovery, vulnerability assessment, backtrace acquisition, and root-cause analysis of known vulnerabilities.

Another solution, *Charm* [9], enhances the dynamic analysis of device drivers in mobile systems. Charm's key innovation is remote device driver execution, allowing device drivers to run in a virtual machine. In contrast, the mobile device is used solely for servicing low-level and infrequent I/O operations through a low-latency, customized USB channel. This approach isolates the analysis from the mobile device, enabling more efficient and thorough testing.

Similarly, *PROSPECT* [10] supports dynamic code analysis of embedded binary code within arbitrary analysis environments. By transparently forwarding peripheral hardware access requests from the original host system to a virtual machine, security analysts can run and analyze the embedded software without requiring intimate knowledge of the embedded peripheral hardware components.

While these tools have collectively advanced the state of dynamic firmware analysis, they still face significant challenges. Dynamic testing or fuzzing of embedded firmware is often constrained by hardware dependencies and limited scalability, contributing to the ongoing vulnerability of IoT devices. Additionally, these solutions demand substantial expert knowledge and manual effort to set up and operate, further hindering their scalability and practical application in real-world scenarios.

11.2.3 Formal Verification

Formal verification plays a pivotal role in the development of complex information systems. It is a systematic process that uses mathematical reasoning to ensure that a system's design specification remains consistent throughout the implementation process. One of the most widely adopted techniques for formal verification is the *Symbolic Model Verifier (SMV)*. Despite its success in commercial designs,

SMV has limitations, particularly in handling the size and complexity of verifiable designs [11].

Formal verification requires engineers to adopt a different mindset from traditional testing methods. While simulation relies on empirical testing—using trial and error to explore combinations of inputs and uncover potential errors—this approach can be time-consuming and incomplete. Engineers typically create numerous input scenarios in simulation, focusing on trying to break the design rather than ensuring it behaves as intended in all situations. In contrast, formal verification is both mathematical and exhaustive, allowing engineers to verify the correct behavior of the design comprehensively.

Several tools are commonly used in formal verification. *Coq*[1] is a formal proof management system that provides a formal language for writing mathematical definitions, executable algorithms, and theorems and an environment for developing machine-checked proofs semi-interactively. Similarly, the *HOL* interactive theorem prover [2] is a proof assistant for higher-order logic, offering a programming environment where theorems can be proved and proof tools developed. *ACL2*[3] is another software system comprising a programming language, an extensible theory in first-order logic, and an automated theorem prover. *Isabelle*[4] is a generic proof assistant that enables the expression of mathematical formulas in a formal language, providing tools to prove these formulas in a logical calculus. Initially developed at the University of Cambridge and the Technical University of Munich, Isabelle has since benefited from global contributions from various institutions and individuals. Despite the maturity of formal verification tools, some technologies need to catch up in this area.

11.2.4 Identify Vulnerabilities at the Processor Level

For many years, secure system design has been built on a foundational assumption: hardware is inherently trustworthy. However, recent research has revealed that this assumption is flawed and that hardware often represents the weakest link in the security of commodity supply chains. Worse still, the hardware/software interface is inadequately specified, with a notable absence of microarchitectural contracts. This gap frequently leads to new security vulnerabilities as software unintentionally violates assumptions silently made by hardware.

1. https://coq.inria.fr/
2. https://hol-theorem-prover.org/
3. https://www.cs.utexas.edu/users/moore/acl2/
4. https://isabelle.in.tum.de/

From Rowhammer [12] to cache side-channel attacks [13] and from Spectre [14] and Meltdown [15] to RIDL/MDS [16], it has become clear that modern hardware presents a vast attack surface. Attackers can exploit this surface to mount attacks compromising real-world systems' integrity and confidentiality. Sadly, our understanding of these hardware-level vulnerabilities remains limited. Researchers continuously discover new vulnerabilities, and vendors respond with long embargo periods to develop solutions that minimize disruption. These solutions often include patches for CPU microcode, hypervisors, operating system kernels, compilers, and browsers.

This cycle of vulnerability disclosure and patch deployment is ongoing, with newly discovered issues sometimes refining or complementing existing vulnerabilities. As a result, new *"spot"* mitigations are frequently applied to real-world systems. However, the fragmented landscape of mitigations, especially in the last five years since the disclosure of Spectre and Meltdown, has led to a chaotic state of security for systems. Practitioners aiming to protect their systems against known (N-day) vulnerabilities face many mitigation techniques, complex dependencies, and unclear applicability and security guarantees.

More concerning is that even when deploying all vendor-recommended mitigations for a specific hardware/software stack, experts find it difficult, if not impossible, to quantify the residual attack surface. This challenge exists even when ignoring the complex interactions between N-day vulnerabilities or any yet-to-be-disclosed zero-day vulnerabilities. The current state of the art reflects this fragmented status quo [17]. While individual efforts often focus on specific attack variants [18], there remains a critical lack of comprehensive techniques to systematically assess the security of a hardware/software stack. Furthermore, developing strategies that can offer robust security guarantees against zero-day vulnerabilities is still an unmet need.

11.3 Methodology

RESCALE considers supply chain security a holistic issue that must be addressed by integrating hardware and software components within the software application supply chain. This approach establishes a chain of trust among the various elements within the supply chain. Unlike recent efforts focusing solely on the software aspects of supply chain security, RESCALE incorporates hardware and firmware components, which serve as the backbone of software application DevSecOps and the overall SDLC. Recognizing this, RESCALE adopts the *Bill of Materials (BOM)* as a conceptual vehicle for conveying supply chain information, leveraging established BOM standards (e.g., SPDX, CycloneDX, and SWID) while enhancing this construct with supply chain trust for each link within the chain.

On top of that, RESCALE introduces the *Trusted Bill of Materials (TBOM)*, which lists the components of an application and their dependencies and includes information about the security tests performed, their outcomes, and the guidelines followed in conducting these tests. To ensure trust, the TBOM is linked to a public Blockchain 2.0 (e.g., Ethereum, Cardano), where the results of the security tests are recorded and secured, with the TBOM itself integrated as a digital asset. The security of each TBOM component is guaranteed through cryptographic measures. In contrast, the non-repudiation of the security tester/evaluator and the test results are ensured by storing a smart contract on the blockchain. Acting as proof of responsibility, this contract uses the TBOM hash as its asset. As a result, the RESCALE trusted and secure TBOM-based supply chain ensures high levels of traceability, immutability, and authenticity, all inherited from its blockchain association, while supporting trusted updates.

The abovementioned concept is built around an advanced static and dynamic security testing mechanism, activated whenever a software or hardware asset is generated and utilized. This testing mechanism is part of a broader security assessment and assurance process, resulting in a TBOM. The process consists of two modules: the static code analysis module and the dynamic testing module. The RESCALE static code analysis module, after evaluating a given software component (referred to as an asset) within the supply chain (e.g., third-party proprietary code or open-source software library), produces a trusted report, known as the *Static Supply Chain Component Guarantee (SSCG)*. The SSCG includes details such as *(a)* the tests performed, *(b)* the test results (whether vulnerabilities were discovered or not), *(c)* the tester's/evaluator's ID, and *(d)* the asset's version. The SSCG is linked to the component's entry within the TBOM of the final software application (the endpoint of the supply chain).

Similarly, after evaluating a given hardware or software component within the supply chain, the RESCALE dynamic testing module produces a trusted report known as the *Dynamic Supply Chain Component Guarantee (DSCG)*. This report is linked to the component's TBOM entry within the final software application or hardware implementation.

The RESCALE security testing modules establish a trusted, secure supply chain via the TBOM. Following the latest NIST guidelines, RESCALE introduces two key entities in the supply chain and BOM establishment: the software/hardware component producer and the software/hardware component consumer. The producer integrates existing components with newly developed code to create a new product, whether a final application or a component (e.g., a library) for other products. In addition to existing components, the producer controls their code and executable code (e.g., binaries or hardware netlists) from prior components. To contribute to the RESCALE trusted secure supply chain, the producer uses the static

code analysis module, generates the SSCG for the new component, and securely includes it in the TBOM. However, other applications cannot use the component directly in the RESCALE model. It must first be further assessed and marked as a candidate for the supply chain.

To make a candidate component a full TBOM entry in the RESCALE supply chain, it must be adopted or used by a consumer (another component of the supply chain or the final application). The consumer characterizes the candidate component as a third-party asset (proprietary or open-source) and performs a security audit. Initially, the SSCG guidelines are validated for compliance with existing standards and the consumer's security policy. After passing this initial check, the component undergoes dynamic testing within the RESCALE sandbox environment, tailored to the component type (e.g., hardware, processor, firmware, OS, driver, etc.). If the component passes all security tests, its DSCG outcome is linked to its SSCG, and the component becomes part of the consumer's TBOM. The dynamic testing process only occurs when a consumer first uses the component as the evaluator. The evaluator records the DSCGs on a public blockchain and secures them with a smart contract, with the software/hardware TBOM as its asset. This blockchain transaction, concatenated with the SSCG, legitimizes the component for integration into other applications, and future consumers can use the component without re-evaluating it unless the evaluation period has expired or the component has been updated.

We acknowledge that many open-source and proprietary components may lack the producer's security guarantees or static code analysis. In the absence of an SSCG, and if the consumer's security policy permits, the consumer can perform static code analysis using the RESCALE module to generate an SSCG for inclusion in the supply chain. In this case, the component will still undergo dynamic testing to receive the DSCG, and both the SSCG and DSCG will be stored on the blockchain. Figure 11.1 depicts RESCALE's security assessment process.

11.3.1 Security and Trust in the TBOM-based Supply Chain

The RESCALE trusted supply chain is structured around secure TBOM constructs, combining the *Hardware Bill of Materials (HBOM)* and SBOM. Each recorded component (as a BOM entry) is associated with the related SSCG and DSCG, along with proper authenticity information about every component created or used in the supply chain.

To achieve a high level of trust, the information in the TBOM must maintain integrity and be genuine, unforgeable, and authentic. Existing BOM standards, particularly for SBOM components, often rely on unverified attestations with security assumptions related to the *Uniform Resource Identifier (URI)*, the primary identifier

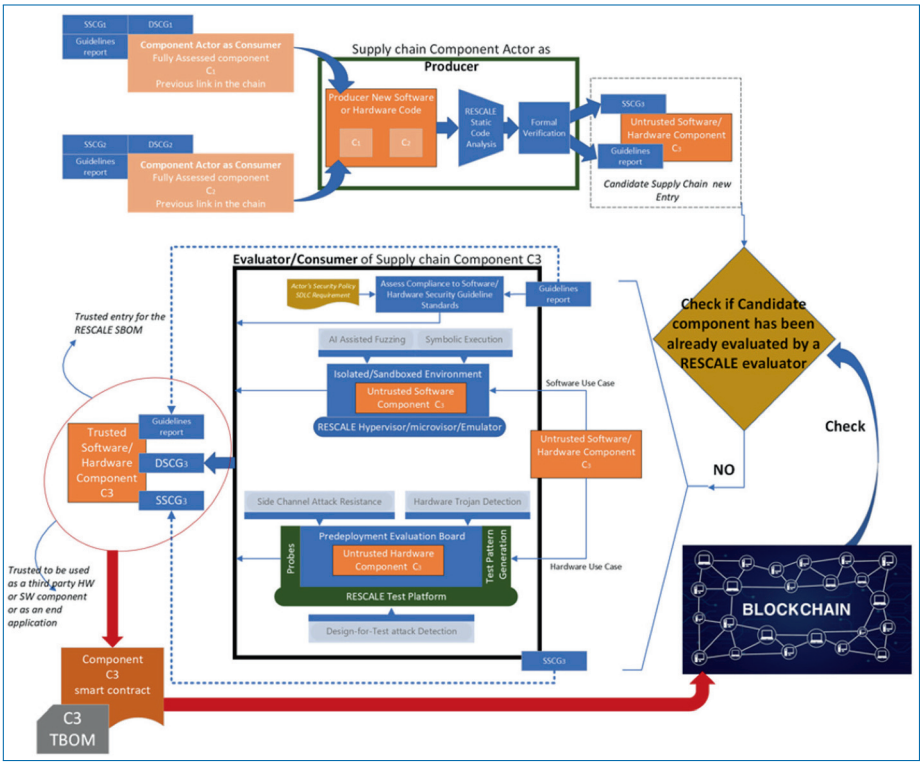

Figure 11.1. RESCALE security assessment process.

for a software component within an SBOM. Within the BOM, these URIs are presumed to act as contact points or identifiers for the manufacturer. The assumption is that the manufacturer's attestation and trust are established through conventional security protocols like TLS/SSL, with associated public key certificates validated when the URI is accessed.

However, this reliance on domain names poses challenges, as domain names can be hacked or mismanaged, especially by entities outside the control of the BOM producer or consumer. This weakness implies that the current BOM authentication process lacks the robustness required for a truly trusted system. This is even more critical in RESCALE, as the TBOMs carry vital information regarding the security testing performed on each component, including the SSCG, DSCG, guidelines, and associated vulnerabilities.

RESCALE aims to provide a trust framework capable of acting as a trust attestation for the hardware and software component stack in any supply chain or end software/hardware application through the TBOM. To succeed, the BOM information must be authentic, accountable, attributable to the component's producer and evaluator, and non-repudiable. In RESCALE, we extend the functionality of

BOM validation tools, such as the OWASP Dependency Graph, by adding trust capabilities, including formal verification processes, reputation systems, and secure building blocks like centralized or decentralized registries/ledgers. A key activity for ensuring authenticity in BOM entries is linking them to public blockchains (Blockchain 2.0 and beyond) to complement URIs. When a TBOM is created, the software or hardware solution producer associates the TBOM with a smart contract on the blockchain. This contract uses the TBOM as a blockchain asset; it binds the TBOM with the producer, product version, TBOM validity period, component assessment periods, TBOM hash, TBOM ownership, and component IDs.

Each producer in RESCALE acts as a consumer for third-party components used in their products. If any component lacks a DSCG (or SSCG), meaning it has not been evaluated yet, the producer, acting as a consumer, takes on the role of evaluator. For these components, the producer retrieves the SSCG from the component's TBOM blockchain entry, generates the DSCG, and stores it in their own TBOM. The evaluated component's SSCG, DSCG, and guidelines are then added to the product's TBOM. Additionally, the producer/evaluator creates a smart contract on the blockchain that includes the TBOM as an asset and records the components they have evaluated. This blockchain contract ID is forwarded to the evaluated component's creator. Thus, whenever this component is used in the future, it will provide the SSCG along with the blockchain smart contract ID of the evaluator. The consumer can review all the security testing details by accessing this smart contract. The security testing can be trusted since blockchain entries are unforgeable and immutable. RESCALE will also implement a reputation-based trust system for evaluators to enhance trust further.

A crucial aspect of the overall trusted supply chain is the unique, unforgeable identification of a software or hardware component and its associated TBOM. In RESCALE, the software TBOM includes a digital signature of the software (using its hash value) and a digital certificate of the software author. This process complements the RESCALE blockchain mechanism and is tied to the blockchain's unique ID, as described earlier.

For hardware TBOMs, a fundamental challenge is establishing an indisputable physical-to-digital association between a piece of hardware and its related certifications. This issue is addressed in various ways, depending on the hardware characteristics and the desired security level. Solutions may range from traditional Public Key Infrastructure (PKI) approaches, which resemble the software TBOM process, to QR-code stamping on the device. In RESCALE, we will explore existing techniques to establish a low-requirement hardware "identity" through hardware fingerprinting (e.g., via Physically Unclonable Functions, etc.), identifying appropriate fingerprinting methods for certain hardware classes, and formalizing

processes and formats to create this association. Ultimately, a unique hardware ID will be generated and added to the RESCALE blockchain, digitally signed, and incorporated into the TBOM.

11.3.2 Management of Trustworthy Updates

Building on the trust mechanism, blockchain technology ensures the traceability of hardware and software components used in the supply chain of a given application. TBOM smart contracts contain information about the assessed components, their version, and the validity period of the assessment. In RESCALE, we provide a trust validator mechanism that evaluates a software or hardware application's overall correctness, security, trustworthiness, and TBOM. A crucial aspect of this process is determining, via blockchain entries in the TBOM, whether a component has been updated. If an update is identified, the TBOM entry for that component must be updated with a new blockchain entry ID (linked to the new smart contract). Once the updated component has successfully passed the static and dynamic security tests and has new SSCG and DSCG reports, the associated product will be updated with the new component version.

When a software or hardware solution—whether the end product of the supply chain or an individual component—undergoes an update, it must be statically and dynamically retested following the RESCALE testing approach. Until the updated solution passes these tests, it cannot be considered a fully trusted component of the RESCALE trusted supply chain. Instead, it reverts to a candidate component status after the static code analysis. The existing blockchain smart contract for the product has been updated to indicate that an update exists, and the TBOM has been revised to a new version. A new blockchain smart contract is linked to the updated solution version. Since the component is now in candidate status, consumers of the solution will act as evaluators, and they will link the new blockchain smart contract ID (of the updated solution) within their own TBOM.

11.4 Security Testing

The complexity of modern software and hardware supply chains demands rigorous security testing to ensure the integrity and resilience of systems. As software and hardware components are sourced from multiple vendors and integrated into the SDLC, vulnerabilities at any stage of the supply chain can compromise the entire system. Security testing is critical in identifying these vulnerabilities, mitigating risks, and strengthening the overall security posture of an application.

11.4.1 Static Code Security Testing

The RESCALE static code security testing provides a comprehensive static code analysis module, incorporating innovative static code analyzers for both source code and binary files. This module combines traditional code analysis engines with *Deep Learning (DL)*-based techniques to offer accurate vulnerability assessments for the most popular programming languages (e.g., C, C++, Java, Python). Traditional analysis engine results enhance the DL analyzer's training process, improving its ability to detect code vulnerabilities.

Static code analyzers generate alarm reports based on rule-based techniques and graph-based control flow analysis. While effective, these approaches often produce a high volume of false alarms, undermining user trust in the analyzer and its results. This issue is addressed in RESCALE by integrating *Machine Learning (ML)* and DL techniques into the static code analysis flow, following the latest research trends. Additionally, reinforcement learning algorithms are deployed to improve the accuracy of the ML/DL models. If component distribution during assessment is necessary, RESCALE adopts federated learning techniques to ensure privacy for both the components and transmitted data. These models are then applied to post-classify generated alerts, identifying false positives and allowing the static analyzers to filter out or assign low risk to these alerts, thus enhancing the accuracy of the analysis through a multiparametric fusion approach.

Beyond improving static code analysis, RESCALE utilizes DL-based classification to identify whether data processed in transit or storage is sensitive or non-sensitive. Based on this classification, appropriate security and cryptography operations are recommended to protect the data. The RESCALE static code analyzers are linked to developed security and cryptography primitive blocks (software or hardware libraries), which, based on the data classification, guide the developer in achieving high data security during design time. Sensitive data insecurity is another type of vulnerability that must be addressed.

RESCALE also introduces an intelligent vulnerability exposure engine based on symbolic execution. Although various static analysis tools, such as Coverity, Parasoft, and Klocwork, exist, most suffer from high false positive rates due to their reliance on simpler syntactical and semantic analysis rather than symbolic execution. While symbolic execution tools like JPF (for Java) and Otter (for C) are available, they struggle to scale to real-world applications. RESCALE overcomes these limitations by offering a formal verification solution for software.

Given a software component, RESCALE's intelligent vulnerabilities exposure engine verifies whether the program satisfies specific properties (i.e., assertions embedded in the code). A property is satisfied if no feasible execution path leads to violating the corresponding assertion. RESCALE either proves the property holds

or provides a counterexample demonstrating a violation. This verification process involves symbolic interpretation, which combines the explicit exploration of all feasible execution paths with a symbolic representation of input variables. Through this approach, RESCALE delivers an intelligent code quality assessment framework that remains active throughout the entire software development and maintenance lifecycle.

11.4.2 Dynamic Testing

Apart from the above static code analysis, the RESCALE security testing toolbox includes the RESCALE Dynamic Testing Module, which features specialized ML/DL-based fuzzers capable of performing black-box or gray-box dynamic testing. These fuzzers use ML/DL techniques to generate and update the appropriate test vector inputs through a reinforcement learning process, which acts as the corpus for the fuzzer's mutation or evolution mechanism. These ML/DL techniques are complemented by symbolic execution analyzers to generate proper constraints on the input corpus data as aligned by the ML/DL analyzers. In the second stage of the RESCALE dynamic testing approach, we provide tools and mechanisms to discover security vulnerabilities in the supply chain's hardware (or firmware) and software components. Different dynamic testing approaches address these issues per component type within the RESCALE supply chain.

11.4.2.1 Hardware and Software Side Channel Security Testing

For side-channel vulnerabilities, RESCALE offers dedicated platforms capable of performing hardware power and electromagnetic emission side-channel trace collection. These traces are assessed using in-house RESCALE assessment libraries focusing on *(a)* generic non-specific leakage assessment through high-order t-tests and *(b)* dedicated profile-based side-channel attack assessment, similar to hardware penetration testing. A dedicated side-channel trace collection and analysis platform for hardware IP cores, combined with COTS side-channel trace collection tools, is enhanced with specialized DL trace analysis software. In addition to power consumption and electromagnetic emission side-channel leakage analysis, RESCALE explores timing leakages, particularly in software components handling sensitive information (e.g., cryptographic keys). Finally, the platform also identifies potential hardware Trojans by detecting abnormalities during computation.

11.4.2.2 Chip-Level Security/Vulnerability Testing

Given the diverse and often untrusted hardware supply chain, the risk of hardware Trojan injection at various stages (designers, testing facilities, foundries, etc.)

is a significant threat. RESCALE addresses this using ML/DL test vector generation techniques, such as *Automatic Test Pattern Generation (ATPG)* and ML/DL side-channel analysis from the Hardware Side Channel Assessment toolbox. This multiparametric approach increases the likelihood of triggering hardware Trojans and detecting timing or power consumption anomalies that reflect such triggers. Additionally, RESCALE assesses vulnerabilities in hardware chips that could be exploited post-design, such as those stemming from the on-chip Design-for-Test infrastructure, which may expose sensitive information through scan-chain-based attacks.

11.4.2.3 Firmware Dynamic Testing

To address growing concerns around the security of embedded systems, RESCALE provides an accurate analysis of firmware binaries, even when source code or hardware documentation is unavailable. A dynamic analysis framework is developed that orchestrates the execution of an emulator together with real hardware. A special software proxy allows firmware instructions to be executed in the emulator while I/O operations are channeled to the physical hardware. This approach facilitates large-scale firmware analysis and uncovers new security insights into embedded devices and their firmware.

11.4.2.4 Operating System and Processor Security Testing

To identify vulnerabilities at the processor level (e.g., transient execution errors) and operating system vulnerabilities specific to certain processors, RESCALE offers dedicated lightweight Virtual Machine-like environments based on QEMU/Gem5 hypervisors/emulators equipped with appropriate sensors/detectors. These environments capture vulnerabilities in a sandbox, focusing on microarchitectural attacks, such as cache attacks, processor architecture-based side-channel attacks, and fault injection attacks like Rowhammer. Furthermore, security sensors will be integrated into the QEMU-based processor kernel emulation to identify embedded OS kernel vulnerabilities, which could impact the end software solution using these kernel modules in its supply chain.

11.4.2.5 Cloud/Container/Microservice Security Testing

RESCALE also considers container-based security aspects. From a RESCALE perspective, microservices introduced in a container are treated as supply chain components and handled accordingly, following the RESCALE trusted supply chain establishment process. A cloud service, which relies on multiple microservices, is treated like software and hardware component-based supply chains and is associated with a TBOM. Dynamic testing within a container (i.e., a provided microservice) is conducted by placing the container in a sandbox environment (e.g., mine sandbox

environments) and monitoring system call behavior through system call interpolation analysis. This technique is enhanced with anomalous behavior analyzers using Deep Learning models to detect unknown, anomalous interactions between the tested container and its external environment. Fuzzing techniques developed for software executables will be adapted for container security testing when applicable.

11.4.3 Supply Chain Trust Orchestrator and Continuous Security Assurance

The RESCALE *Trust Orchestrator (TrustOR)* acts as the primary handler of the TBOM for a given hardware or software solution. TrustOR has a dual role in the RESCALE project. First, it generates the RESCALE TBOM by utilizing the security assessment results from static and dynamic testing (producing the relevant SSCG, DSCG, and guidelines as previously described). Second, it validates existing TBOMs. TrustOR automatically processes software and hardware, orchestrating the necessary RESCALE components to generate the appropriate TBOM. This includes utilizing RESCALE's asset modeling features and sharing mechanisms with relevant initiatives such as the DBoM consortium,[5] which focuses on developing an infrastructure for supply chain attestation and TBOM sharing. As a TBOM generator, TrustOR orchestrates the security testing of software or hardware solutions. As a consumer of software or hardware components, TrustOR identifies whether a component is a candidate or a fully integrated RESCALE supply chain entry. If it is still a candidate, TrustOR manages all necessary evaluator operations, including dynamic testing, recording information on the RESCALE public blockchain, and ultimately incorporating the component into the end solution's TBOM.

Beyond the above capabilities, TrustOR addresses the need for continuous security assessment at runtime, going beyond the static TBOM information of a given software or hardware component. Since the vulnerability landscape evolves rapidly and zero-day vulnerabilities are constantly discovered, relying solely on the TBOM for security may be insufficient. From the time of security testing to when a TBOM may need to be updated or replaced, new vulnerabilities could emerge that are not reflected in the existing TBOM. To mitigate this, RESCALE integrates TrustOR with a *continuous security assurance platform*, which monitors the security status of software in real time. If new vulnerabilities are detected, the platform triggers a TBOM update process.

The RESCALE TrustOR, in conjunction with the continuous, evidence-based security assurance platform, provides comprehensive safety, security, and privacy

5. https://dbom.io/

assessments for all software or hardware solutions' assets (i.e., components) (e.g., network, compute, data, processes). The RESCALE model-based assessments go beyond the security testing platform and incorporate continuous, evidence-based evaluations of safety, security (confidentiality, integrity, availability), and privacy for software and hardware components. This process supports the following modalities: *(a)* dynamic testing based on the RESCALE security testing platform, *(b)* penetration testing, *(c)* runtime monitoring, *(d)* certificate-based assessments that evaluate the relevance and impact of existing certifications (and the evidence underpinning them) on the overall risk posture of the software or hardware solution, and *(e)* hybrid analysis that combines assessments from different modalities.

Finally, the proposed solution programmatically integrates with various architectural components of a system via appropriate probes (e.g., event captors, test tools), enabling RESCALE's risk identification capabilities and orchestrating its continuous risk assessment capabilities.

11.5 Field of Applications

The strategies and innovations presented in RESCALE provide a comprehensive solution to securing the software supply chain across various industries. RESCALE offers significant advancements for protecting complex supply chains in critical infrastructure, aerospace, automotive, healthcare, financial services, and cloud computing by addressing hardware and software vulnerabilities.

Critical infrastructure sectors, such as energy, telecommunications, transportation, and water management, rely heavily on intricate software and hardware systems to manage essential services. These systems are increasingly integrated with third-party software components, open-source libraries, and connected devices, all contributing to a significantly expanded attack surface. The RESCALE framework addresses these vulnerabilities by providing enhanced traceability through the TBOM, which allows granular tracking of hardware and software components, ensuring that all third-party components are rigorously tested and verified. In addition, by utilizing blockchain for immutable security records, RESCALE ensures that once a component has been certified, its security assessment remains tamper-proof. This feature is particularly valuable in protecting critical infrastructure systems, where any compromise could have severe consequences. Furthermore, RESCALE incorporates dynamic risk mitigation tools, which continuously monitor critical system components in real-time, allowing for proactive identification and resolution of emerging vulnerabilities. In this way, critical infrastructure systems benefit from increased resilience against known and zero-day supply chain attacks.

The ***aerospace and defense industries***, characterized by highly sensitive and complex systems, demand the most rigorous security measures. The reliance on diverse hardware and software components integrated across various platforms introduces significant risks to the supply chain, where compromised components could lead to catastrophic failures. RESCALE offers a holistic security approach by encompassing both hardware and software supply chains, ensuring that each component—from processors to software drivers—undergoes comprehensive security assessment. Using blockchain-backed TBOMs, enables the aerospace and defense sectors to maintain an auditable and trustworthy record of each component's security status. This capability mitigates the risk of introducing malicious or compromised components into mission-critical systems, providing assurance and traceability. The security-by-design methodology embedded in RESCALE allows these industries to meet stringent regulatory requirements and standards while ensuring their systems remain resilient in increasingly sophisticated cyberattacks.

The transition toward autonomous and connected vehicles has increased the complexity of software and hardware integrations in the ***automotive industry***. Relying on multiple suppliers for components such as sensors, communication modules, and onboard control systems introduces new cybersecurity challenges. RESCALE provides a robust solution by offering supply chain transparency through its TBOM mechanism, enabling automakers to track each component used within vehicle systems. This traceability ensures that all hardware and software components are rigorously assessed for security vulnerabilities, safeguarding against potential cyberattacks, such as remote vehicle hijacking or disabling critical safety features. Furthermore, RESCALE's dynamic testing tools, tailored for automotive systems, ensure the integrity of firmware, a crucial aspect of connected vehicle security. By enabling automakers to meet functional safety and cybersecurity standards, RESCALE helps reduce non-compliance risk, protecting manufacturers from costly regulatory penalties while enhancing overall vehicle security.

Industrial Control Systems (ICS), which play a crucial role in automating manufacturing, energy production, and logistics processes, face unique cybersecurity challenges due to their reliance on interconnected devices, including *Programmable Logic Controllers (PLCs)* and SCADA systems. While highly efficient, these systems are highly vulnerable to cyberattacks, especially when embedded firmware and hardware components are sourced from various suppliers. RESCALE enhances the security of ICS environments by providing comprehensive firmware and hardware testing, which helps to identify and mitigate vulnerabilities that could disrupt operations. The framework's continuous security compliance monitoring tools enhance system security by providing real-time assessments, ensuring that ICS operators

can promptly detect and address new vulnerabilities before they can be exploited. Additionally, RESCALE supports IoT device security within industrial settings by securing the integration of connected devices into the ICS supply chain, thereby preventing common attack vectors, such as botnet infections and data breaches, from compromising critical industrial processes.

The rapid adoption of connected medical devices, electronic health records (EHRs), and telemedicine platforms has introduced new cybersecurity risks in the ***healthcare industry***. The sensitive nature of healthcare data and the potential consequences of compromised medical devices make supply chain security a priority for healthcare providers. RESCALE addresses these concerns by offering comprehensive security assessments for medical devices and healthcare systems. The framework's static and dynamic testing modules ensure that each medical device, whether implantable or used within a hospital setting, undergoes rigorous security assessments to prevent unauthorized access and potential tampering. Additionally, RESCALE ensures the confidentiality and integrity of healthcare data through cryptographic measures, providing compliance with regulatory standards such as GDPR and HIPAA. By continuously monitoring system components for compliance and updating the TBOM with new vulnerability assessments, RESCALE ensures that hospitals and healthcare providers maintain high levels of security and trust in their systems, safeguarding patient data and medical treatments' safety.

Financial institutions face some of the highest cybersecurity risks, given the sensitive nature of the data they handle and the value of the transactions they process. The financial sector's reliance on third-party software, cloud services, and hardware for tasks such as transaction processing, fraud detection, and data analytics introduces significant risks to the security of these systems. RESCALE strengthens cybersecurity for financial services by providing a comprehensive supply chain risk management solution. Through the TBOM, financial institutions can track every component in their software stack, ensuring that third-party software, open-source tools, and hardware are thoroughly assessed before being integrated into critical systems. RESCALE's dynamic testing and AI-based predictive security services enable financial institutions to detect abnormal patterns and potential fraud indicators in real time, enhancing the security of banking operations. Additionally, the framework's automation of compliance audits ensures that financial systems meet the strict requirements of regulations such as PCI-DSS and PSD2, while the immutable blockchain-backed records provide a trusted foundation for regulatory assurance.

Finally, ***cloud providers and data centers*** form the backbone of modern IT infrastructure, handling vast amounts of sensitive data and facilitating critical operations for organizations across the globe. However, the complexity of cloud environments, which often involve virtual machines, containers, and microservices,

creates significant security challenges. RESCALE addresses these challenges by providing comprehensive security testing and continuous assessment tools for cloud and data center environments. The framework's container security testing ensures that microservices and containers, frequently used in cloud operations, are thoroughly vetted for vulnerabilities, reducing the risk of attacks such as container escapes or privilege escalation. RESCALE's side-channel attack detection tools add another layer of security by identifying vulnerabilities in software and hardware components, such as timing attacks or cache-based side-channel leaks, which are becoming increasingly prevalent in cloud environments. By automating the security testing of these components and continuously updating the TBOM with new vulnerability information, RESCALE ensures that cloud and data center services remain secure, even in the face of emerging threats.

11.6 Conclusion

RESCALE offers a comprehensive solution to the growing security challenges in software and hardware supply chains. By integrating static and dynamic security testing with blockchain technology, RESCALE ensures that every component, whether software, hardware, or firmware, undergoes rigorous assessment, resulting in a resilient and trustworthy supply chain. The innovative use of a Trusted Bill of Materials tracks each component's security status and secures the integrity of this information through blockchain, providing an immutable record that enhances transparency and accountability across the supply chain. RESCALE's holistic approach addresses vulnerabilities that traditional methods often overlook, such as hardware flaws and supply chain complexity. Its capability to continuously monitor and update components in real time positions RESCALE as a forward-thinking framework capable of mitigating existing and emerging threats. By adopting RESCALE, organizations can ensure their systems' integrity, security, and compliance, significantly reducing the risk of supply chain attacks and enhancing the overall cybersecurity posture.

Acknowledgments

Funded by the European Union (Grant Agreement Nr. 101120962, RESCALE Project). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Health and Digital Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

References

[1] R. Hiesgen, M. Nawrocki, T. C. Schmidt and M. Wählisch, "The Race to the Vulnerable: Measuring the Log4j Shell Incident," in *Network Traffic Measurement and Analysis Conference*, 2022.

[2] R. Alkhadra, J. Abuzaid, M. AlShammari and N. Mohammad, "Solar Winds Hack: In-depth Analysis and Countermeasures," in *International Conference on Computing Communication and Networking Technologies*, 2021.

[3] P. Ladisa, H. Plate, M. Martinez and O. Barais, "SoK: Taxonomy of Attacks on Open-Source Software Supply Chains," in *IEEE Symposium on Security and Privacy*, 2023.

[4] F.-X. Standaert, "Introduction to Side-Channel Attacks," *Secure Integrated Circuits and Systems*, 2010.

[5] W. Xiong and J. Szefer, "Survey of Transient Execution Attacks and their Mitigations," *ACM Computing Surveys*, 2021.

[6] K. Xiao, D. Forte, Y. Jin, R. Karri, S. Bhunia and M. Tehranipoor, "Hardware Trojans: Lessons Learned after One Decade of Research," *ACM Transactions on Design Automation of Electronic Systems*, 2016.

[7] B. Xia, T. Bi, Z. Xing, Q. Lu and L. Zhu, "An Empirical Study on Software Bill of Materials: Where we Stand and the Road Ahead," in *International Conference on Software Engineering*, 2023.

[8] J. Zaddach, L. Bruno, A. Francillon and D. Balzarotti, "AVATAR: A Framework to Support Dynamic Security Analysis of Embedded Systems' Firmwares," in *Network and Distributed System Security Symposium*, 2014.

[9] S. M. S. Talebi, H. Tavakoli, H. Zhang, Z. Zhang, A. A. Sani and Z. Qian, "Charm: Facilitating Dynamic Analysis of Device Drivers of Mobile Systems," in *USENIX Security Symposium*, 2018.

[10] M. Kammerstetter, C. Platzer and W. Kastner, "PROSPECT: Peripheral Proxying Supported Embedded Code Testing," in *ACM symposium on Information, computer and communications security*, 2014.

[11] F. Copty, A. Irron, O. Weissberg, N. Kropp and G. Kamhi, "Efficient Debugging in a Formal Verification Environment," *International Journal on Software Tools for Technology Transfer*, vol. 4, 2003.

[12] O. Mutlu and J. S. Kim, "Rowhammer: A Retrospective," *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2019.

[13] F. Liu, Y. Yarom, Q. Ge, G. Heiser and R. B. Lee, "Last-Level Cache Side-Channel Attacks are Practical," in *IEEE Symposium on Security and Privacy*, 2015.

[14] P. Kocher, J. Horn, A. Fogh, D. Genkin, D. Gruss, W. Haas, M. Hamburg, M. Lipp, S. Mangard, T. Prescher, M. Schwarz and Y. Yarom, "Spectre Attacks: Exploiting Speculative Execution," *Communications of the ACM*, 2020.

[15] M. Lipp, M. Schwarz, D. Gruss, T. Prescher, W. Haas, J. Horn, S. Mangard, P. Kocher, D. Genkin, Y. Yarom, M. Hamburg and R. Strackx, "Meltdown: Reading Kernel Memory from User Space," *Communications of the ACM*, 2020.

[16] S. Van Schaik, A. Milburn, S. Österlund, P. Frigo, G. Maisuradze, K. Razavi, H. Bos and C. Giuffrida, "RIDL: Rogue In-Flight Data Load," in *IEEE Symposium on Security and Privacy*, 2019.

[17] P. Jattke, V. Van Der Veen, P. Frigo, S. Gunter and K. Razavi, "Blacksmith: Scalable Rowhammering in the Frequency Domain," in *IEEE Symposium on Security and Privacy*, 2022.

[18] E. Barberis, P. Frigo, M. Muench, H. Bos and C. Giuffrida, "Branch History Injection: On the Effectiveness of Hardware Mitigations Against Cross-Privilege Spectre-v2 Attacks," in *USENIX Security Symposium* , 2022.