From Containers to Code: Applying Post-9/11 Trade Security Lessons to the Software Supply Chain

Stephanie Lieggi
slieggi@ucsc.edu
University of California, Santa Cruz

Juanita Gomez
jgomez91@ucsc.edu
University of California, Santa Cruz

Robert Shaw
rashaw@middlebury.edu
Middlebury Institute of International Studies

Tekla Gagoshidze
tgagoshidze@middlebury.edu
Middlebury Institute of International Studies

Shayiq Shah
shayiqahmeds@middlebury.edu
Middlebury Institute of International Studies

1 Introduction

Over the past quarter century, two security challenges have arisen that, while distinct in nature, have comparative structures that could enable policymakers to leverage key lessons learned. The first, which was catalyzed by the attacks of September 11, 2001, saw a sudden and intense focus put on the security of the global trade network and the concern that belligerent state and non-state actors could exploit weak links and bring down the global flow of commerce. More recently, software security has taken center stage for policymakers due to incidents like SolarWinds, Log4j and XZ Utils, highlighting the extent to which the global software supply chain could be exploited for nefarious purposes [1–3]. While there are clear differences between the supply chains being challenged – one clearly dealing with tangible goods in shipping containers and the other with intangible code in digital artifacts – underlying policy questions and risk management needs have important overlaps. In both cases, policymakers, industry and civil society grapple with similar questions: how can we manage risk in a complex, multi-actor system where trust is essential but can be easily exploited? And how do we balance security imperatives without creating unnecessary barriers to efficiency and innovation?

This report takes an initial look at whether lessons learned from the experience of trade security can be leveraged to provide workable policies and practices for open source software security. In particular, the authors will provide an analysis of initiatives such as the US-based Customs-Trade Partnership Against Terrorism (C-TPAT) and the Container Security Initiative (CSI) – as well as their international counterparts – and the factors that enabled their relative success in meeting industry and policymakers goals, such as the incentive based structure that allowed effective public-private partnerships and provided a basis for international cooperation through harmonized standards [4]. This research also looks to move the conversation beyond the current analogies of

the domains – particularly use of Software Bill of Materials (SBOM) – and whether that directly parallels the trade security terminology related to shipping manifests.

This on-going research, funded through a grant from the Digital Infrastructure Insights Fund (DIIF), is continuing to build upon the preliminary analysis set forth in this paper. In the coming months, we expect to conduct a series of interviews aimed at validating our findings, better understanding how policy is currently evolving around open source software (OSS) security, and providing a detailed framework for actionable recommendations. In particular, we will explore in more detail the parallels between the policies that developed in our historic case study of trade controls and the current debate around regulatory frameworks like the EU's Cyber Resilience Act (CRA). We hope one output from our research is a framework that builds on the trusted actor programs from our historical case study and establishes a similar mechanism on the software development side. We also imagine a framework that – similar to strategic trade case study – focuses collaborative efforts and positive incentives for developing and maintaining software security. Policies that center investment in critical cyber-infrastructure, development of risk assessment tools, and establishing international standards are likely to be more effective than those that aim to control, dictate to and punish open source projects and their maintainers.

2 Open Source Supply Chain Security: Understanding the Weakest Links

Understanding the weakest links in open source ecosystems clarifies why purely technical solutions are insufficient and why policy interventions modeled on trade security may be effective. A number of major security incidents have brought the issue of software security into policy conversations globally. According to the European Union Agency for Cybersecurity (ENISA), supply chain attacks are expected to quadruple in frequency. Organizations need to develop comprehensive strategies to address this growing threat landscape [5]. The software supply chain encompasses all the materials, components, processes, people and channels that it takes to develop a software product [6]. This includes not only proprietary code but all the third party libraries used as dependencies which includes a vast amount of open source software.

Some of the most critical links in this supply chain are the package managers, integration systems, packaging and deployment tools and all the developers involved in the process. Therefore, when talking about supply chain security, we need to look at every layer of the software development and delivery process [7]. Protecting source code is not enough, since an attacker can compromise any weak link in the supply chain to get access to key systems. Magnifying the impact of compromises in security of tangible goods – where malicious intent can be more easily kept to a smaller area – software supply chain attacks inject malicious code into applications, allowing attackers to infect all their users rapidly [8]. For example, third party libraries have a lot of dependencies, and the impact of a security flaw can be inherited by a vast number of downstream applications [5]. Additionally many organizations struggle to maintain a complete and accurate inventory of all the software components in their environments, specifically the "transitive" dependencies that are pulled in by the libraries they use [9].

2.1 Examples of OSS Security Incidents

Recent security incidents help illustrate the diversity of supply chain attacks and the different vectors through which they can occur:

The **SolarWinds** attack targeted a critical network monitoring platform, and was the hallmark of a classic "trojan horse" approach where malware downloads onto a computer disguised as a legitimate program. Attributed to a Russia-backed group, the attackers compromised the Orion SolarWinds build pipeline, inserting malicious code directly during the software compilation process rather than exploiting a dependency or the final application [1]. In September 2019, they gained access to the platform's internal systems by compromising employee credentials, and between March and June 2020 distributed trojanized Orion's updates to roughly 18,000 customers, including government agencies, critical infrastructure providers, and private corporations [10]. The implanted malware, known as Sunburst, created backdoors in infected networks, allowing attackers to spy on Orion SolarWind customers and gain access to many critical IT systems in companies and government agencies using their software.

The **Log4j** "zero-day" vulnerability was a different kind of supply chain failure – and in many ways highlighted that security systems are only as strong as their weakest link. This vulnerability – also known as Log4Shell - was a critical flaw that allowed remote code execution in any system using this popular Java-based utility. Log4j is one of the most widely used logging libraries, built into consumer endpoints, web applications, and cloud services, which made the impact of this attack massive [11].Because it is often present as an indirect dependency (included automatically through another library rather than being directly installed by a developer), many users and organizations found it extremely challenging to figure out if they were affected by this vulnerability and how to fix it. The disclosure led to a worldwide effort to locate and patch vulnerable systems, a task the U.S. Department of Homeland Security estimated could take a decade to complete because of how embedded this vulnerability is in the software supply chain [12].

The most recent **XZ Utils** attack's impact on open source communities and project governance is still developing. This incident was a deliberate attack on the software supply chain of the open source compression library found in nearly all Linux and other Unix-like systems. The perpetrator of the attack was a supposedly trusted member of the community who planted a backdoor into the utility. Highlighting the impact of social engineering, the attacker gained the trust of project maintainers and was given administrative access to the repository. After that, the attacker made periodic, subtle changes that appeared harmless but introduced the backdoor functionality. On March 31, 2024, another member of the community discovered the malicious code and immediately reported it [13]. This developer's alert came just in time to avoid the changes being integrated into major distributions such as Debian and Red Hat, which could have led to widespread compromise across millions of computers [14].

2.2 Response to Incidents and Policy Changes

These three incidents highlight the breadth of software supply chain threats: SolarWinds illustrates the compromise of a trusted process; Log4j demonstrates the exploitation of a flawed component; and XZ Utils attack shows the compromise of trusted developers in a community. A comprehensive security strategy must address all of these attack vectors by combining process integrity, component transparency, and stronger safeguards around project governance and contributor trust. In response to these and other supply chain attacks of recent years, the cybersecurity community and policymakers have developed frameworks and tools aimed at improving software supply chain security:

1. **Databases** are used for vulnerability intelligence in supply chain security. They contain historically known vulnerabilities, and in component analysis, they are used as sources to identify risks in new software. The three commonly used databases are CVE [15], CWE [16],

and NVD [17].

2. **Static analyzers** play a crucial role within the software supply chain since they automatically identify vulnerabilities present in the artifact's source code. In this category, we find CodeQL [18], one of the most widely used tools integrated with GitHub that runs queries to find specific security issues in the codebase, and Semgrep [19], which analyzes code based on semantic rules that are easily customizable.

3. **Software Bills of Materials (SBOMs)** aim to enumerate all components in an artifact [20]. Tools such as Syft, Trivy [21], and Microsoft's SBOM Tool [22] generate SBOMs, typically exported in standard formats like SPDX or CycloneDX [23]. Complementary efforts include SWID tags for unique component identifiers and VEX documents (e.g., OpenVEX) that specify whether reported vulnerabilities are exploitable [24].

4. **Integrity tools** ensure software has not been tampered with. **Sigstore** provides signing and verification via cryptographic signatures [25], while **TUF** [26] and **in-toto** [27] add provenance and transparency, recording who performed each supply chain step.

5. **Software supply chain standards** help verify software components by measuring activities, controls, and practices in software development to reduce risks in the supply chain. The two primary standards are **SCVS**, created by OWASP, and **SLSA**, implemented by the Linux Foundation. SCVS provides a framework for identifying activities, controls, and best practices for reducing risk in the software supply chain, while SLSA has a narrower scope focused on integrity, ensuring that consumed code has not been tampered with [28].

Although these initiatives have advanced the security of open source projects, significant challenges remain. Adoption across the ecosystem is still limited, with many organizations lacking the resources, expertise, or incentives to fully integrate these practices. In addition, it is still not clear how to prioritize security actions, ensure integration among tools and standards, and measure the effectiveness of these actions.

The current regulatory effort gaining much attention from open source communities is the European Union's Cyber Resilience Act (CRA) which was announced in 2023 [29]. In the original draft of the CRA, software developers were required to ensure the security of their code throughout its life-cycle, a requirement that was seen as particularly problematic for open source software and projects. In an open letter to the EU, a number of leading open source foundations, including the Linux and Eclipse foundations and the Open Source Initiative (OSI), warned that the act would have a "chilling effect" on open source software development [30]. Since the initial announcement of the CRA, these OS leaders began active collaboration with the EU to revise the regulations to make it less onerous on open source projects; these discussions remain active and concerns remain that the CRA's regulation will take a punitive approach to ensuring software security.

The lack of open source community involvement in the drafting of rules like the CRA highlights the problem that can occur when key stakeholders are kept out of the development of policy around complex issues like software supply chains. Addressing the current gaps in OS security will require more consistent involvement of all stakeholders than what has been seen until now, as well as a recognition from policymakers and regulators that punitive efforts may be less effective than efforts that create incentives for promoting widespread adoption of supply chain security practices. Looking at the historic case study of trade security in the early 2000's may provide some guidance on the importance of public-private collaboration for software related policies moving forward.

3 The Post-9/11 Paradigm Shift in Tangible Supply Chain Security

As we turn our attention to the parallels and lessons learned from the historical case study around securing physical supply chains, we posit that the institutionalization of programs for securing trade and related supply chains demonstrate that voluntary, incentive-based partnerships can align public-sector security goals with private-sector efficiency priorities. This provides an analog to open source software governance, where community participation and market adoption hinge on positive incentives rather than compliance mandates.

The events of 9/11 highlighted the grave potential for terrorist exploitation of the vast network of global trade [31]. Governments feared that terrorist organizations would exploit the weakest link in cargo-centered supply chains to transport explosive devices or weapons of mass destruction – and even detonate one in a major port-of-entry, disrupting commerce on a global scale. Trade security therefore became an imperative for authorities charged with national security, who aimed to secure the entirety of global cargo-based supply chains, from point-of-origin to final delivery of goods. Ultimately the goal was to eliminate the weakest link in such supply chains, and specific to the U.S., strategies even include mandates to screen 100% of inbound cargo – an effort that proved impractical, given the sheer volume of cargo entering the country each day [32]. However, pragmatic, tailored and overlapping countermeasures quickly emerged after 9/11, at multilateral and national levels – and involving public and private sector actors. Engagement with the latter reflected recognition that securitization of trade needed to be achieved without adversely impacting global commerce. These initiatives included incentive-based public-private partnerships such as C-TPAT and Automated Economic Operator (AEO) programs, reinforced internationally through Mutual Recognition Agreement (MRAs) [33]. Additionally, government-to-government efforts such as the SAFE Framework and the Container Security Initiative (CSI) increased information-sharing and pushed screening of cargo further up the supply chain and closer to the point-of-origin [34]. Examining these initiatives and the drivers leading to their introduction and, ultimately, their institutionalization can yield useful insights when considering strategies for enhancing security across the OSS domain.

3.1 Incentive-based Public-Private Partnerships: C-TPAT and AEO Programs

Launched in November of 2001, the US-based Customs-Trade Partnership Against Terrorism (C-TPAT) initiative is aimed at navigating this balancing act between trade securitization and preserving global commerce [35]. As the lead agency in charge of introducing and implementing the C-TPAT initiative, US Customs and Border Protection Agency (CBP) recognized that the government alone could not secure every node in the immense global commercial supply chain. Hence, drawing upon input from the private sector, CBP envisioned the C-TPAT initiative as a government-industry collaborative venture [36]. Being a voluntary program, C-TPAT was founded on the principle of shared responsibility and features an incentive-based apparatus [37]. A company voluntarily applied to join C-TPAT and agreed to work with CBP to protect its own supply chain by identifying areas of risk or gaps in their security efforts and formulating action plans to mitigate such associated risks [38]. In exchange for their commitment to enhancing global supply chain security, companies are certified by CBP as C-TPAT partners. Such designations carry significant and tangible benefits related to streamlining the importation of goods into the U.S.. Partners enjoy a reduced number of examinations, front-of-the-line inspections, shorter wait times at the border, and access to Free and Secure Trade (FAST) lanes [35]. C-TPAT possesses a tiered structure, offering progressively greater benefits to partners who demonstrate a greater level of commitment to trade security [37]. Such a tiered structure creates a powerful market incentive for continuous

improvement, treating security not as an obligatory compliance checkbox but as an opportunity to attain competitive business advantage.

C-TPAT continues to operate and is now largely institutionalized within key import-focused stakeholders in the trade community [39]. References to C-TPAT can be found on logistics service providers' and other stakeholders' websites, as a signal to business partners that the company is well-positioned to facilitate smooth importation of goods into the U.S [40]. In this sense, C-TPAT is seen by the trade-focused sector as adding value to services offered or import-dependent retail operations. From a government perspective, the program still occupies a pivotal position in the US' strategy of global trade securitization.

In a parallel international effort to secure global supply chains, the World Customs Organization (WCO) introduced Authorized Economic Operator (AEO) programmes in 2005, equivalent to C-TPAT in its efforts to partner with the private sector and balance trade and security objectives [41]. After the European Union implemented its AEO system in 2008, it has expanded dramatically, from 512 certified traders to over 18,000 by 2022, with AEO-certified companies involved in 74% of total imports and 84% of total exports of EU [42]. Currently, AEO programs are operational in 90 countries, covering all of EU, 15 countries in the Americas, 12 in Asia-Pacific, 9 in East and South Asia, 8 in Middle East and North Africa, and 1 in West and Central Africa, while 7 additional programmes are under development [43].

Once a company is granted AEO status by one EU country, that status is recognized across all other EU Member States by their customs authorities. The EU scheme offers three types of authorisations: AEOC for customs simplification, AEOS for security and safety, or a combined AEOC/AEOS version. The programme is open to all supply-chain actors who meet the relevant criteria, including importers, exporters, manufacturers, freight forwarders, etc [44].

Depending on their AEO type, holders may enjoy benefits such as: easier qualification for simplified customs procedures; fewer physical and documentary checks under both security/safety and customs legislation; advance notice if selected for inspections; priority treatment when controls occur; the option to request inspections at preferred locations; and various indirect advantages such as enhanced reputation, fewer delays and losses, improved customer service, and lower inspection costs for supply-chain partners. The benefits also include Mutual Recognition Agreements (MRAs) with third countries, under which an AEO with the security/safety category (AEOS or AEOC/AEOS) can request that the third-party customs authorities grant them comparable perks in that foreign jurisdiction, extending the facilitation benefits beyond the EU's borders. The EU currently holds MRAs with 8 countries: Canada, China, Japan, Moldova, Norway, Switzerland, the US, and United Kingdom [45].

3.2 International, Government-to-Government Initiatives: The SAFE Framework and the Container Security Initiative (CSI)

The AEO program contributed to a wider framework linking various "pillars" of global supply chain security and focused on customs services worldwide. Established as the SAFE Framework of Standards to Secure and Facilitate Global Trade (SAFE Framework), this framework Included three pillars: Customs-to-Customs network arrangements (Pillar 1), Customs-to-Business partnerships (Pillar 2), and Customs-to-other Government Agencies cooperation (Pillar 3) [46]. The Framework was adopted explicitly as a counter-terrorism measure post 9/11, which promoted measures like advance cargo data, risk management, and the development of national AEO-type programs [47].

Reflective of Pillar 1, governments turned attention and resources to strengthening Customs-to-Customs partnerships. An example is the U.S. Container Security Initiative (CSI), which was launched by U.S. Customs and Border Protection (CBP) in January 2002 aimed at achieving global trade securitization through pre-screening of outbound cargo. While C-TPAT, another CBP initiative, focused on partnering with industry professionals and the private sector at large, CSI focused on partnering with willing foreign governments.

Such partnership was aimed at identifying and inspecting high-risk or suspicious maritime cargo containers at foreign ports, rather than screening them at a US port of entry [31]. Such a cooperative security arrangement, focused on extended US security far beyond its physical borders to, ideally, prevent any potential threats from even reaching US domestic soil. This pre-screening of cargo before being put on vessels traveling to the US, remains a voluntary undertaking that foreign governmental hosts, or their customs authorities, assist USCBP officers with. Containers identified as high-risk, under the CSI arrangement, are inspected by deploying non-intrusive screening technologies, such as large X-ray and radiation detection mechanisms to ensure all threats are identified.

Much like other CBP programs, CSI also carries a dual burden. The initiative is aimed at ensuring trade securitization while preventing an impediment to global flow of commerce. Therefore, there are built in arrangements within CSI that ensure that such duality in goals is achieved by marrying security measures with efficiency. A shipment that has passed through a foreign CSI port will undergo zero inspections upon arrival on US soil, except for an occasional random examination [48]. Even CSI's deployment of non-intrusive technologies for pre-screening purposes ensures rapid and effective screening with the least possible disruption in the flow of licit trade activity [49]. Having completed 23 years of its operations in January 2025, CSI has expanded its scope steadily since its introduction. Operational on six continents, CSI's 61 operational ports are located across Asia, Africa, Asia, the Americas, and the Middle East. It is estimated that over 80 percent of all maritime containerized cargo imported into the United States, is being successfully pre-screened at foreign CSI ports [49].

3.3 Success Factors

Multiple factors drove adoption of the above programs at both multilateral and national levels – and in the case of C-TPAT and AEO, both the private and public sector – and then their sustainment to continuing relevance today. These factors include:

1. **Sense of shared urgency** - The immediate impact of the 9/11 terrorist attacks was felt intensely world-wide, particularly in the transportation sector with the grounding of flights – inclusive of air cargo [50]. Fears of additional attacks drove action, and there was already an expectation among governments as well as in the private sector that security across the global supply chain would require more attention [51].

2. **Incentivization for key stakeholders** – Unlike security-focused government agencies, private sector actors' focus is directed primarily toward profit-focused goals and being responsive to investors' expectations. Among these actors, security-related compliance requirements may be seen as a sunk, unrecoverable cost. Notably, the C-TPAT island AEO initiatives addressed this concern by featuring an opportunity for participating companies to streamline importing processes. Context here is important, as this streamlining occurred at a time when supply chains were rapidly globalizing with an emphasis on speed of delivery – underscored by the popularity of just-in-time (JIT) logistical structures [52]. The incentives offered by the C-TPAT and AEO initiatives are directly connected with these priorities, resonating with importers

and related companies due to perceived advantages for the "bottom line". Incentivization also played a role in the widespread, international adoption of CSI, with partner governments seeing benefits in access to shared intelligence, visibility in contributing to international anti-terrorism and nonproliferation goals, and streamlined access to a key export market (the US via reduction of costly import holds associated post-arrival screening) [53].

3. **Clear guidance that aligned with existing operations** – While adoption of C-TPAT, AEO and more broadly, the SAFE Framework and CSI, involved implementation of new security-focused requirements, these did not require a full re-engineering of the importation process and related supply chain operations. New elements were added to import declarations designed to improve advanced screening of cargo for security purposes. However, the declaration process itself was not overhauled. Similarly, in the case of C-TPAT and AEO, freight forwarders were expected to improve physical security measures, such as fencing and lighting, but this did not translate into a need to build new warehouses altogether[1].

4. **Auditing / monitoring** – A central feature of C-TPAT and AEO specifically is the auditing and monitoring of security-specific records and operations [54]. Related to alignment with existing systems, auditing was already a core component of the importing sector, as Customs periodically assessed Harmonized Standard Tariff Codes recorded by importers, to verify that they applied appropriate duty rates[2]. In the same way that the importers' self-classification of goods has been subject to regular audits– thus keeping the community attuned to accurate classification practices – auditing / monitoring ensured that C-TPAT-driven security practices were not treated as 'one-off' actions, to be completed but then forgotten. Rather, they became part of continuous operations – analogous perhaps to quality control in product development (inclusive of software).

As a result of the above, C-TPAT enjoyed early success that was then sustained, and supply chain security became part of a common lexicon shared by the U.S. trade community and relevant government stakeholders. Finding parallels to the above within potential public-private collaboration on security of OSS may translate to similar success.

4 Bridging the Tangible and Digital Worlds: Analogies and Their Limits

The primary focus of this research has been to understand whether we can develop an effective framework for the open source software supply chains by looking at potential parallels in the more mature domain of trade security for tangible goods. The concepts and frameworks developed since 2001 to manage risk in the movement of tangible goods provide potential models for thinking about the development and flow of software – recognizing that there may be limits to analogies between the two domains.

[1] One of this report's authors observed these changes in real-time while working in the air export sector north of San Francisco International Airport.

[2] Also referred to as "HS Codes", goods declared for importation are classified using an internationally-accepted system of Harmonized Standard Tariff codes. National customs authorities publish duty rates in tabular format, indexed according to HS Codes.

4.1 (S)BOMs – Similar Objectives but Different Objects

A shipping manifest, and its corresponding bill of material, is a foundational document in global trade, providing a detailed inventory of all items within a shipment. Its primary purposes are to enable customs officials to verify contents for compliance and risk assessment, to allow carriers to track and manage cargo, and to provide a definitive record in case of dispute. These documents provide transparency for what is in a shipping container and give authorities and other trade stakeholders a place to begin risk assessments. These are also documents that can enable more effective auditing and tracking of tangible goods.

Many draw a direct analogy to the shipping manifest and the growing popular SBOM. The concept of the SBOM was of course borrowed directly from the trade security lingo and looked to the software inventories as a source of transparency and risk management. However, drawing too many parallels between these two divergent standards is problematic and ignores the fundamentally divergent nature of tangible and digital goods. Software cannot be inventoried and listed in the same way tangible goods can, and software that includes "third party package may incorporate it in its entirety, or select individual files, or functions, or even lines of code.... There is no atomic unit of software" [55]. While this does not mean that SBOMs do not have their place in the overall OSS security framework, they should be seen as just one of many tools to build a security standard around.

A shipping manifest typically describes a single layer of contents while an SBOM must capture a deeply nested set of dependencies. A chain of "transitive dependencies" can run dozens of layers deep, creating a level of complexity far beyond that of a bill of materials for a tangible item [5]. The accuracy of SBOM generation is therefore a significant challenge, as current tools can struggle to identify all dependencies [56].

4.2 Building Trust - From Trade to Code

One of the key C-TPAT components that could have a direct impact on the workings of open source projects and their communities is the Trusted Trader program. That program established a framework for certifying that a company's internal security processes meet a defined standard, thereby designating that company as a low-risk partner deserving of benefits like expedited processing. While not solving all issues related to social engineering and trust within a community, establishing a system to certify that open source projects are adhering to established best practices for secure software development could mitigate a number of trust issues currently impacting OSS security. Similar to the Trusted Trader program, this OSS security certification would be voluntary, with projects opting in to gain a competitive advantage. The incentives would shift from faster border crossings to benefits like preferential treatment in government and enterprise procurement and safe harbor from potential liabilities. A project certification program would likely be based on established industry standards such as the NIST Secure Software Development Framework (SSDF) [57], the SLSA framework [58], and/or OpenSSF Scorecard [59]. These frameworks provide a common baseline for what constitutes "good" security practice. In order to make this project security certification more effective, a mechanism could follow the C-TPAT process of self-assessment then external validation [36]. In the realm of software, a project core team would verify its compliance with the chosen framework, and then this self-check would be validated, perhaps by a certified third-party auditor or a government body. For the certification to be recognized internationally, multilateral agreements like those that underpinned the trade security landscape would need to be negotiated, including agreement from agencies such as the US government's CISA and European

Union Agency for Cybersecurity (ENISA), among others. As was evident in the C-TPAT process, those who want to maintain this certification would need to demonstrate an on-going process for managing security and vetting their contributors and contributions.

4.3 Limits of Analogies

Applying lessons learned must include recognising when the comparison between tangible and digital goods have their limits. As noted in the section above pointing out the limitations of SBOMs, tangible goods are static; software is dynamic. Software is not made in a single place like a factory, can be replicated and distributed globally nearly instantaneously. Whereas the "point of origin" for a physical good is a verifiable factory, the "origin" of software can be spread across a geographically diverse community of contributors, many working in a volunteer capacity. Inspecting these projects and their contributions is a different process than a system that relies on physical scanners and searches. Like in trade controls, 100% "scanning" for software security would be near impossible and not cost efficient. Even more than with cargo screening, creating systems of trust that incentivize self-assessments are essential in open source software development.

An SBOM is a list of ingredients; it does not, on its own, confirm that those ingredients are free of vulnerabilities. Its true value is realized only when used as a part of vulnerability scanning and risk analysis. Policy must therefore focus not just on generating SBOMs, but on building the ecosystem of tools and standards needed to act on them. Looking at the diverse nature of open source projects and their contributor communities also means that there cannot be a one-size-fits-all program for any cyber-version of the Trusted Trader program.

5 Actionable Lessons

While this project is continuing discussions with stakeholders to create a full framework and set of policy recommendations, we have identified a number of initial actionable lessons learned from the trade security case study that can help with creating a more resilient software supply chain. It is important to note that the lessons we see for open source software security are not meant to replicate specific regulations, but instead enable the adoption of strategic pathways that proved effective in a similarly complex, global, and high-stakes environment.

5.1 Incentivizing Security Minded Communities

The most enduring lesson from trade security policies – particularly C-TPAT – is that security adoption is most effective and sustainable when it is driven by incentives rather than by punitive regulation. C-TPAT was successful in large part because it reframed an actor's investment in security as a pathway to a competitive advantage like faster border crossings, fewer inspections, and lower operational costs [60]. While C-TPAT was voluntary, the disadvantages for non-participants, such as higher inspection rates and unpredictable delays, create a strong pull toward participation.

This principle has the potential to be applicable to the software world, where developers can often perceive security as a source of friction that slows down release schedules and overall adoption. A successful software supply chain security program could therefore be built around a value proposition which focuses on incentives such as liability safe harbors for projects and adopters adhering to best practices, which would fundamentally alter the risk calculus for software producers [61]. Other incentives could include preferential treatment in both public and private sector procurement, and reduced cyber insurance premiums. By creating a clear "market" signal that secure development

practices lead to greater adoption and user access, policymakers can look to these incentives to elevate the security posture of the entire ecosystem [61] – although this impact may have limitations in many open source projects where market forces are not as impactful.

The Container Security Initiative's strategy of "pushing the border out" by screening high-risk cargo at the port of origin is a powerful demonstration of proactive security [62]. It is vastly more efficient and effective to identify and neutralize a threat before it is loaded onto a vessel and sent across the ocean than it is to find it upon arrival amidst millions of other containers. Looking at this approach from the lens of software development, incentives for integrating security into the earliest stages of the software development lifecycle rather than treating it as a final, pre-release inspection step is also vital [37].The lesson from CSI is to apply this principle relentlessly to the software supply chain. Security should begin with the selection of components and vetting of libraries before they are incorporated. Scanning should also occur throughout the build and deployment pipeline to prevent the injection of malicious code, treating it as the critical "port of origin" for all software artifacts [63].

While some of this can be automated, a critical part of creating these effective pathways in software development is to ensure the community has a security mindset throughout the development process. Open source projects rely on their communities, and their communities are made of people – people who need to understand the importance security has on the long-term success and sustainability of the project they are working on. Documentation for projects should include instructions for security checks on contributions and stress the need to work only with trusted members of the community. Similar to how trade professionals saw their role in container security efforts, open source community members should also understand that they are the first line of defense for software security.

5.2 International Standards and Mutual Recognition

The global nature of trade necessitated global security standards over the last two decades. C-TPAT's success in the international arena is heavily reliant on its Mutual Recognition Arrangements (MRAs) with the Authorized Economic Operator (AEO) programs of other nations [64]. These agreements mean that those certified under one country's trusted trader program are recognized by another, allowing benefits to extend across borders and creating a harmonized international security framework. This lesson could have far reaching implications for OSS supply chains, which are arguably more globalized and interconnected than physical trade. Projects – particularly larger, widely adopted ones – are developed by global communities of contributors and are used worldwide. Security standards from a single jurisdiction are insufficient and reliance on standards not accepted more widely would be ineffective and potentially counterproductive, creating barriers to trade and innovation. Policymakers must therefore pursue a strategy of international cooperation to align on foundational security principles. Looking at the way this occurred in the trade security case provides a roadmap for OSS security efforts. Countries, regional groupings, and international organizations should work towards harmonizing standards for secure development lifecycles, agreeing on common formats and data fields for SBOMs, and establishing shared protocols for vulnerability disclosure. MRAs for software security would facilitate a secure global digital infrastructure and allow nations to pool their resources against common challenges.

5.3 The Mandate for Continuous Validation and Improvement

As with C-TPAT certification, certifications for OSS security can not be about one snap short in time but about maintaining and an ongoing commitment to security. In the C-TPAT case, regular

risk assessments and re-evaluations were required to maintain your status. This requirement for continuous review and validation creates a virtuous cycle of improvement and prevents security postures from becoming stale and ineffective. An effective assurance program must be built on a foundation of continuous validation. This means a project's security documentation would be living documents, and their SBOMs would need updating with every build or release [56]. Development and adoption of tools that allow for automated continuous validation are vital and investment into those tools would be a priority for policymakers and industry – who rely on a security supply chain. Leveraging these tools to generate, collect, and analyze security evidence for most of the project's activities, while still undertaking some "manual" audits for most critical systems would help ensure potential vulnerabilities are caught before they become a major security challenge to users.

6 Building a Set of Recommendations - Work in Progress

The ultimate goal of this project is to create a comprehensive framework for enhancing the resilience of global software supply chains. The recommendations coming from this project will look to translate the principles of incentivized partnership, proactive risk management, international cooperation, and continuous validation into a concrete set of initiatives into OSS security. At the writing of this paper, we are just getting into the stage validating our initial recommendations highlighted below. We expect our discussions with relevant experts will help us fine tune and improve upon this initial framework.

6.1 Trusted Source: Programs Recognizing Trusted Contributors

The cornerstone of a new national software security strategy should be the creation of voluntary "Trusted Source" programs, which would provide a series of clear incentives for open source projects and related software producers to demonstrate their security practices and be rewarded for their efforts. Similar to C-TPAT, the program should be tiered to encourage a ladder of continuous improvement. The standards set forth by these programs – which would be administered at the national level – would be overseen by an international collaboration that would include government agencies, industry groups and international organizations. Incentives for achieving and maintaining the security tiers would include the liability safe harbors, streamlined regulatory compliance, and enhanced information sharing between tiered projects and governments on relevant threat intelligence. To enable the broad participation in these programs of open source projects and their largely volunteer communities, governments and other funding organizations should refocus efforts on making international digital infrastructure more resilient and secure. This could involve creating a government fund to provide grants for development of open source tools specifically for security and continual validation as well as resources (financial and otherwise) for maintainers of projects deemed vital to national critical functions.

6.2 Building Multilateral Institutions for Global Standards

Unilateral standards from one nation or region are likely to create barriers to meeting global security needs. The current multilateral efforts and discussions should be institutionalized into a multilateral regime that is often used to develop common standards around global challenges. The goal would be to create a global framework based on shared principles culminating in Mutual Recognition Arrangements (MRAs) for national Trusted Source programs. This would create a secure and interoperable global platform for trusted software, mirroring the success of C-TPAT's AEO partnerships. Due to the community nature of open source, these efforts would also need to

include open source foundations and communities. These stakeholders, along with the government and international organizations, would create standardized documentation for open source projects to utilize that could increase transparency and accessibility of software security internationally.

6.3 Sharing the Burden

Open source is heavily dependent on "volunteer" efforts. While many larger companies who recognize the value of open source are increasingly contributing back to projects they rely on – typically through the in-kind efforts of employees – this support needs to be more standardized if it is to help improve OSS security. Stewardship from the larger open source consumers is needed. Foundations or consortia centered around critical open source projects and based on the relevant communities should be created and maintained in order to provide the resources for the necessary security standards. This operationalizes the C-TPAT principle of shared responsibility, acknowledging that all who benefit from open source projects have a role to play in maintaining its security and sustainability.

7 Conclusion: Securing the Future of Code

The complex challenge of tracking the software dependencies and potential vulnerabilities that underpins our global digital infrastructure is a source of profound risk to international security. The cybersecurity incidents mentioned in this paper were just the most notorious cases highlighting the extent of current structural vulnerability, and showing the truth to the "weakest link" axiom. In order to mitigate the impact of software vulnerabilities, strengthening the full development cycle must be a priority to governments, industry and open source communities.

When the world faced a similar challenge in securing the physical supply chain, stakeholders – led by the US and European governments – opted to create a public-private partnership that focused on proactive risk management and incentivized security. The core principles that guided the trade security efforts – including trust through verification, transparency through documentation, and shared responsibility, all while understanding the need to balance security with efficiency—are also important in any complex, distributed system. The efforts in the early 2000s offers both a historic parallel and provides a strategic pathway for taking on our current digital infrastructure challenge and creating a more resilient system moving forward.

The path forward requires a practical understanding of both the notable analogies and the critical differences between the tangible and digital good domains. It demands a nuanced approach that leverages automation, differentiates between commercial vendors and open source communities, and invests in the foundational components of our digital infrastructure. Securing the future of code is not a task for government or industry alone. It is a shared responsibility, and by drawing on the hard-won wisdom of the past, we can build a supply chain that is not only more secure but also more trustworthy and innovative.

References

[1] Fortinet. Solarwinds supply chain attack. Fortinet Cyber Glossary. [Online]. Available: https://www.fortinet.com/resources/cyberglossary/solarwinds-cyber-attack

[2] IronNet. Log4j: New software supply chain vulnerability unfolding as this holiday's cyber nightmare. [Online]. Available: https://www.ironnet.com/blog/log4j-new-software-supply-chain-vulnerability-unfolding-as-this-holidays-cyber-nightmare

[3] L. Ravid. The xz utils backdoor: Everything you need to know. Wired. [Online]. Available: https://www.wired.com/story/xz-backdoor-everything-you-need-to-know/

[4] United Nations Conference on Trade and Development, "Container security: Major initiatives and related international developments," UNCTAD, Tech. Rep., 2004. [Online]. Available: https://unctad.org/system/files/official-document/sdtetlb20041_en.pdf

[5] FOSSA. The complete guide to software supply chain security. [Online]. Available: https://fossa.com/learn/software-supply-chain-security/

[6] L. Security. Software supply chain security 101. [Online]. Available: https://www.legitsecurity.com/software-supply-chain-security-101

[7] Splunk. What is software supply chain security and how does it work? [Online]. Available: https://www.splunk.com/en_us/blog/learn/software-supply-chain-security.html

[8] CrowdStrike. What is a supply chain attack? [Online]. Available: https://www.crowdstrike.com/en-us/cybersecurity-101/cyberattacks/supply-chain-attack/

[9] GitLab. Software supply chain security guide: Why organizations struggle. [Online]. Available: https://about.gitlab.com/blog/software-supply-chain-security-guide-why-organizations-struggle/

[10] Zscaler. What is the solarwinds cyberattack? [Online]. Available: https://www.zscaler.com/resources/security-terms-glossary/what-is-the-solarwinds-cyberattack

[11] N. C. S. Centre. Log4j vulnerability: What everyone needs to know. [Online]. Available: https://www.ncsc.gov.uk/information/log4j-vulnerability-what-everyone-needs-to-know

[12] IBM. What is the log4j vulnerability? [Online]. Available: https://www.ibm.com/think/topics/log4j

[13] A. Freund. (2024) Backdoor in upstream xz/liblzma leading to ssh server compromise. oss-security mailing list. [Online]. Available: https://www.openwall.com/lists/oss-security/2024/03/29/4

[14] Red Hat. (2024) Urgent security alert for fedora 41 and fedora rawhide users. Red Hat Blog. [Online]. Available: https://www.redhat.com/en/blog/urgent-security-alert-fedora-40-and-rawhide-users

[15] MITRE Corporation. Common vulnerabilities and exposures (cve). [Online]. Available: https://www.cve.org/

[16] ——. Common weakness enumeration (cwe). [Online]. Available: https://cwe.mitre.org/

[17] National Institute of Standards and Technology. National vulnerability database (nvd). [Online]. Available: https://nvd.nist.gov/

[18] GitHub. Codeql. [Online]. Available: https://codeql.github.com

[19] Semgrep. Semgrep: Lightweight static analysis for many languages. [Online]. Available: https://semgrep.dev/

[20] Cybersecurity and Infrastructure Security Agency. Software bill of materials (sbom). [Online]. Available: https://www.cisa.gov/sbom

[21] Aqua Security. Trivy: Find vulnerabilities, misconfigurations, secrets, sbom in containers, kubernetes, code repositories, clouds and more. [Online]. Available: https://trivy.dev/latest/

[22] Microsoft. Sbom tool. [Online]. Available: https://github.com/microsoft/sbom-tool

[23] National Telecommunications and Information Administration, "Survey of existing sbom formats and standards," Tech. Rep., 2019. [Online]. Available: https://www.ntia.gov/files/ntia/publications/ntia_sbom_formats_and_standards_whitepaper_-_version_20191025.pdf

[24] Chainguard. What is openvex? [Online]. Available: https://edu.chainguard.dev/open-source/sbom/what-is-openvex/

[25] Z. Newman *et al.*, "Sigstore: Software signing for everybody," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, 2022. [Online]. Available: https://dl.acm.org/doi/10.1145/3548606.3560596

[26] The Update Framework. Overview. [Online]. Available: https://theupdateframework.io/overview/

[27] S. Torres-Arias *et al.*, "in-toto: Providing farm-to-table guarantees for bits and bytes," in *Proceedings of the 28th USENIX Security Symposium*, 2019. [Online]. Available: https://www.usenix.org/system/files/sec19-torres-arias.pdf

[28] P. Ladisa, H. Plate, M. Martinez, and O. Barais, "Sok: Taxonomy of attacks on open-source software supply chains," in *2023 IEEE Symposium on Security and Privacy (SP)*, 2023, pp. 1509–1526.

[29] A. Ramaswami and M. Boehm. (2023, September) Understanding the cyber resilience act: What everyone involved in open source development should know. [Online]. Available: https://www.linuxfoundation.org/blog/understanding-the-cyber-resilience-act

[30] E. Foundation *et al.* (2023, April) Open letter to the european commission on the cyber resilience act. [Online]. Available: https://newsroom.eclipse.org/news/announcements/open-letter-european-commission-cyber-resilience-act

[31] U.S. Government Accountability Office, "Container security: Expansion of key customs programs will require greater attention to critical success factors," Tech. Rep. GAO-03-770, July 2003.

[32] C. Nolan, "The rubik's cube of cargo screening: Is 100% screening of all u.s.-bound cargo containers prudent?" *American Bar Association Brief*, vol. 42, no. 3, 2013, spring 2013.

[33] U.S. Department of Homeland Security, "Dhs/cbp/pia-013 customs-trade partnership against terrorism (c-tpat)," Tech. Rep., August 2021. [Online]. Available: https://www.dhs.gov/sites/default/files/publications/privacy-pia-cbp013-ctpat-august2021.pdf

[34] U.S. Government Accountability Office, "Maritime security: One year later: A progress report on the safe port act," Tech. Rep. GAO-08-171T, October 2007. [Online]. Available: https://www.gao.gov/assets/a118059.html

[35] ——, "Partnership program grants importers reduced scrutiny with limited assurance of improved security," Tech. Rep. GAO-05-404, 2005.

[36] M. D. Laden, "The genesis of the us c-tpat program: Lessons learned and earned by the government and trade," *World Customs Journal*, vol. 1, no. 2, p. 76, 2007.

[37] U. G. A. Office, "Supply chain security: U.s. customs and border protection has enhanced its partnership with import trade sectors, but challenges remain in verifying security practices," Tech. Rep. GAO-08-240, 2008.

[38] M. Ojah, "Securing and facilitating trade through u.s. land borders: Critical analysis of c-tpat and fast programs," *Transportation Research Record*, vol. 1938, no. 1, p. 32, 2005.

[39] U.S. Customs and Border Protection. (2025) Customs trade partnership against terrorism (ctpat). Accessed October 5, 2025. [Online]. Available: https://www.cbp.gov/border-security/ports-entry/cargo-security/CTPAT

[40] OceanSpray. Ctpat statement of support. [Online]. Available: https://www.oceanspray.com/ctpat-statement-of-support

[41] U.S. Customs and Border Protection. (2025) Securing and facilitating trade in north america. [Online]. Available: https://www.cbp.gov/border-security/ports-entry/cargo-security/c-tpat-customs-trade-partnership-against-terrorism/mutual-recognition/aeo-programs

[42] European Court of Auditors, "Special report: Authorised economic operators," Tech. Rep., 2023. [Online]. Available: https://www.eca.europa.eu/ECAPublications/SR-2023-13/SR-2023-13_EN.pdf

[43] World Customs Organization. Aeo — home. [Online]. Available: https://aeo.wcoomd.org/

[44] European Commission Taxation and Customs Union. Authorised economic operator (aeo) programme. [Online]. Available: https://taxation-customs.ec.europa.eu/customs/authorised-economic-operator/programme_en

[45] ——. Mutual recognition of aeos. [Online]. Available: https://taxation-customs.ec.europa.eu/customs/authorised-economic-operator/mutual-recognition_en

[46] International Chamber of Commerce. (2024, May) Icc releases updated recommendations on authorised economic operator programmes. [Online]. Available: https://iccwbo.org/news-publications/policies-reports/icc-sets-out-recommendations-for-successful-authorized-economic-operators-programmes

[47] World Customs Organization. (2011, September) Wco remembers september 11. [Online]. Available: https://www.wcoomd.org/zh-cn/media/newsroom/2011/september/wco-remembers-september-11.aspx

[48] X. Zhang, "The united states container security initiative and european union container seaport competition," 2018. [Online]. Available: http://dx.doi.org/10.24382/838

[49] U.S. Customs and Border Protection, U.S. Department of Homeland Security, "Csi: Container security initiative cbp seal," https://www.cbp.gov/border-security/ports-entry/cargo-security/csi/csi-brief, 2025, last modified: September 30, 2025. Accessed: October 16, 2025.

[50] P. J. Crowley and B. Butterworth, "Keeping bombs off planes: Securing air cargo, aviation's soft underbelly," Center for American Progress, May 2007.

[51] S. E. Flynn and D. B. Prieto, "Neglected defense: Mobilizing the private sector to support homeland security," Council on Foreign Relations, CSR No. 13, March 2006.

[52] C. Brezing, "Just in time delivery – success stories and practical examples," https://www.lcmd.io/en/blog/just-in-time-delivery-success-stories-and-practical-examples, 2025.

[53] N. Monacelli, "Improving maritime transportation security in response to industry consolidation," Homeland Security Affairs 14, Article 2, page 4, January 2018.

[54] U.S. Customs and Border Protection, "Accountability and transparency features in the united states customs trade partnership against terrorism (ctpat) programme," WCO News 104, Issue 2 / 2024, June 25 2024, https://mag.wcoomd.org/magazine/wco-news-104-issue-2-2024/accountability-and-transparency-features-ctpat/.

[55] A. Gantman. (n.d.) Sbom: Good intentions, bad analogies, ugly outcomes. https://www.linkedin.com/pulse/sbom-good-intentions-bad-analogies-uglyoutcomes-alex-gantman/. Accessed: 2025-10-16.

[56] SANS Institute, "Strengthen your supply chain security with effective sbom management," https://www.sans.org/blog/strengthen-your-supply-chain-security-with-effective-sbom-management.

[57] National Institute of Standards and Technology, "Secure software development framework (ssdf)," https://csrc.nist.gov/projects/ssdf, 2025, last updated: February 27, 2025; Accessed: 2025-10-16.

[58] SLSA, "Supply-chain levels for software artifacts (slsa)," https://slsa.dev/, 2025, accessed: 2025-10-16.

[59] OpenSSF, "Openssf scorecard," https://scorecard.dev/, 2025, assess open source projects for security risks via automated checks.

[60] S. Sullivan and D. Garza, "C-tpat and supply chain effectiveness," in *Proceedings of the Scientia Moralitas Conference, Research Association for Interdisciplinary Studies*, 2021.

[61] George Mason University Costello College of Business, "Creating market incentives to improve cybersecurity," https://business.gmu.edu/news/2025-01/creating-market-incentives-improve-cybersecurity, January 2025.

[62] U.S. Government Accountability Office, "Container security: A flexible staffing model and minimum equipment requirements would improve overseas targeting and inspection efforts," Tech. Rep. GAO-05-557, 2005. [Online]. Available: https://www.gao.gov/assets/a246126.html

[63] C. Okafor, T. R. Schorlemmer, S. Torres-Arias, and J. C. Davis, "Sok: Analysis of software supply chain security by establishing secure design properties," in *Proc. 2022 ACM Workshop on Software Supply Chain Offensive Research Ecosystem Defenses (SCORED)*, Nov. 2022, pp. 15–24.

[64] U.S. Customs and Border Protection, "Mutual recognition, c-tpat program," https://www.cbp.gov/border-security/ports-entry/cargo-security/c-tpat-customs-trade-partnership-against-terrorism/mutual-recognition.