Securing Open Source:
Applying the Logic of Post-9/11 Trade Security to the Software Supply Chain

November 2026

Team Members

Stephanie Lieggi	University of California, Santa Cruz
Robert Shaw	Middlebury Institute of International Studies
K. Jarrod Millman	University of California, Berkeley
Juanita Gomez	University of California, Santa Cruz
David M Torres-Mendoza	University of California, Santa Cruz
Tekla Gagoshidze	Middlebury Institute of International Studies
Shayiq Shah	Middlebury Institute of International Studies

UC Santa Cruz Open Source Program Office

UC Berkeley Open Source Program Office

Middlebury Institute Center for Nonproliferation Studies

Supported by a grant from the Digital Infrastructure Insights Fund, a fiscally sponsored project of Aspiration.

Executive Summary

The security of open source software (OSS) supply chains has become a critical concern for governments, industry, and civil society worldwide. Despite growing regulatory attention and an expanding ecosystem of technical tools, significant gaps remain in how to effectively incentivize security-minded behavior across the diverse and largely volunteer communities that maintain critical digital infrastructure. This report argues that examining the efforts of the global trade security community in the early 2000s—which developed incentive-based frameworks for securing complex, multi-actor supply chains—could be key in addressing these gaps.

Through a combination of literature review and interviews with practitioners from both the trade security and OSS security communities, this research examines the factors that drove adoption and long-term institutionalization of programs such as the US Customs-Trade Partnership Against Terrorism (C-TPAT), the Container Security Initiative (CSI), and multilateral Authorized Economic Operator (AEO) programs, and assesses where their underlying logic translates meaningfully to the software domain. The research also identifies where the analogy breaks down, recognizing that software's dynamic nature, lack of a single point-of-origin, and complex transitive dependencies require adaptation rather than direct replication of trade security models.

The findings suggest a number of strategic transferable lessons: the value of voluntary, tiered certification programs that create genuine value for participants, from competitive advantage for commercial actors to funded security capacity for volunteer communities; the importance of aligning security requirements with existing workflows; the power of international mutual recognition arrangements in preventing fragmentation across jurisdictions; and the necessity of continuous validation over one-time compliance events. The expert interviews also surface traceability—the ability to verify who performed each step in a software artifact's creation—as an important unsolved technical problem on which the effectiveness of any broader framework ultimately depends. Most notably, the research finds that the incentive calculus differs significantly across actor types—commercial vendors, open source foundations and ecosystems, and individual volunteer maintainers each require different entry points, baseline requirements, and benefit structures.

Based on these findings, the report proposes a framework centered on voluntary "Trusted Source" certification programs, supported by expert attestations, proportionate burden-sharing between commercial actors and open source communities, and internationally harmonized standards developed with genuine open source community collaboration. The evidence points toward an approach that treats security investment as a pathway to competitive advantage and community trust, rather than a compliance burden, as the most promising model for securing the open source software supply chain at scale. Realizing that potential, however, depends on policymakers first building strong relationships with open source communities and investing in security awareness as foundational preconditions, before any certification or regulatory machinery is introduced.

Contents

Key Terms and Acronyms **3**

1 Introduction **6**

2 Open Source Supply Chain Security: Understanding the Weakest Links **6**
2.1 Examples of OSS Security Incidents 7
2.2 Response to Incidents and Policy Changes 8

3 The Post-9/11 Paradigm Shift in Tangible Supply Chain Security **9**
3.1 Incentive-based Public-Private Partnerships: C-TPAT and AEO Programs 10
3.2 Government-to-Government Initiatives: The SAFE Framework and the Container Security Initiative (CSI) 11
3.3 Success Factors 11

4 Bridging the Tangible and Digital Worlds: Analogies and Their Limits **12**
4.1 (S)BOMs—Similar Objectives but Different Objects 13
4.2 Building Trust—From Trade to Code 13
4.3 Limits of Analogies 13

5 Research Methodology **14**
5.1 Bridging Two Expert Communities 14
5.2 Reaching Across Domains: Expert Interviews 14
5.3 Integrating Research and Interview Evidence 14

6 Framework for Action: Recommendations Grounded in Expert Evidence **15**
6.1 Preconditions for Success: Building Trust, Awareness, and Cooperation 15
6.2 Incentivizing Security-Minded Communities 17
6.3 Trusted Source: Tiered Programs for Recognizing Trusted Contributors 18
6.4 International Standards and Mutual Recognition 19
6.5 The Mandate for Continuous Validation and Improvement 20
6.6 Traceability as the Technical Frontier 21
6.7 Sharing the Burden: Commercial Responsibility for Open Source Security 22

7 Conclusion: Securing the Future of Code **24**

References **26**

A Interview Expert Descriptions **32**

B Interview Instruments **34**
B.1 Trade Security Practitioners 34
B.2 OSS and Software Supply Chain Security Practitioners 35

Key Terms and Acronyms

The following terms and acronyms appear throughout this report, organized into two sections reflecting the two domains examined.

Trade Security Terms	
Acronym / Term	**Definition**
AEO (Authorized Economic Operator)	A multilateral voluntary certification program through which businesses that meet defined security and compliance standards are recognized as trusted trading partners by customs authorities. Established under the World Customs Organization's SAFE Framework, adopted in 2005, and now implemented through more than 90 operational programs worldwide.
BOM (Bill of Materials)	A document used in physical trade that provides a detailed inventory of all items, components, and materials within a shipment or product. Used by customs officials to verify contents and assess risk.
C-TPAT (Customs-Trade Partnership Against Terrorism)	A voluntary US government program launched in 2001 and administered by Customs and Border Protection (CBP). Companies that meet defined supply chain security standards receive benefits such as fewer inspections, faster border processing, and access to dedicated trade lanes.
CBP (Customs and Border Protection)	The US federal agency responsible for regulating and facilitating international trade, collecting import duties, and enforcing trade laws.
CSI (Container Security Initiative)	A US government program launched in 2002 that stations CBP officers at foreign ports to pre-screen high-risk cargo containers before they are loaded onto vessels bound for the United States.
MRA (Mutual Recognition Arrangement)	A bilateral or multilateral agreement between governments that recognizes each other's trusted trader certifications as equivalent, allowing certified companies to receive benefits in multiple jurisdictions without requiring separate certification in each.
SAFE Framework	The World Customs Organization's Framework of Standards to Secure and Facilitate Global Trade, adopted after 9/11. It promotes international cooperation through three pillars: customs-to-customs cooperation, customs-to-business partnerships, and customs-to-other government agency cooperation.
WCO (World Customs Organization)	An intergovernmental organization representing customs administrations from over 180 countries. The WCO developed the SAFE Framework and AEO program guidelines, providing the international standards backbone for post-9/11 trade security cooperation.

OSS Security Terms	
Acronym / Term	**Definition**
CISA (Cybersecurity and Infrastructure Security Agency)	An agency within the US Department of Homeland Security responsible for protecting critical infrastructure from cyber and physical threats, and for coordinating national efforts on software security, including guidance on SBOMs and secure software development.
CRA (Cyber Resilience Act)	A European Union regulation, proposed in 2022 and adopted in 2024, establishing mandatory cybersecurity requirements for products with digital elements sold in the EU. The adopted text creates a lighter regime for open source software stewards and provides for voluntary security attestation of open source software.
CVE (Common Vulnerabilities and Exposures)	A publicly maintained database of known security vulnerabilities in software, each assigned a unique identifier. CVEs allow the security community to reference and track specific vulnerabilities consistently across tools and platforms.
European Union Agency for Cybersecurity (ENISA)	The European Union agency dedicated to cybersecurity, which developed the European cybersecurity certification scheme, published threat assessments, and supports implementation of EU cybersecurity legislation, including the Cyber Resilience Act.
in-toto	An open source security framework that records and verifies who performed each step in the software development and delivery process, providing guarantees about supply chain integrity.
NIST (National Institute of Standards and Technology)	An agency within the US Department of Commerce that develops technical standards and guidelines, including widely used cybersecurity frameworks such as the Secure Software Development Framework (SSDF). NIST also maintains the National Vulnerability Database (NVD).
OpenSSF (Open Source Security Foundation)	An initiative hosted by the Linux Foundation that brings together technology companies, government agencies, and open source communities to improve the security of open source software.
OSS (Open Source Software)	Software whose source code is made publicly available under a license that allows anyone to inspect, modify, and distribute it. Much of the world's critical digital infrastructure relies on open source software maintained by communities of volunteer and paid contributors.
Reproducible Builds	A set of software development practices ensuring that a given source code input always produces an identical binary output, allowing anyone to independently verify that a distributed artifact was built from its claimed source.
SBOM (Software Bill of Materials)	A formal, machine-readable inventory of the components, libraries, and dependencies that make up a software product. While the term was borrowed from trade security terminology, the two instruments function quite differently in practice.

SCA (Software Composition Analysis)	A category of security tooling that automatically identifies open source components in a software project and checks them against vulnerability databases to detect known security issues.
SLSA (Supply-chain Levels for Software Artifacts)	A security framework developed by Google and now hosted by the OpenSSF that defines a tiered set of requirements for improving software supply chain integrity, focusing on preventing tampering during the build and release process.
Software Supply Chain	The full set of components, tools, processes, people, and systems involved in developing and delivering a software product—including all third-party libraries, dependencies, build systems, and distribution channels.
SSDF (Secure Software Development Framework)	A set of guidelines published by NIST providing a common language and practices for integrating security into the software development lifecycle.

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 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?

The research presented in this report, funded through a grant from the Digital Infrastructure Insights Fund (DIIF), examined in detail whether lessons learned from the experience of trade security could be leveraged to provide workable policies and practices for open source software security. In particular, the project team analyzed initiatives such as the US-based Customs-Trade Partnership Against Terrorism (C-TPAT) and the Container Security Initiative (CSI) [4]—as well as relevant international counterparts—and the factors that enabled their broad adoption and institutionalization, such as the incentive-based structure that allowed effective public-private partnerships and provided a basis for international cooperation through harmonized standards. The direct security effects of these programs are unclear: a 2005 US government report indicated that C-TPAT offered limited assurance of improved security [5], while a more recent review identified issues with inconsistent enforcement [6]. The lessons drawn here therefore concern how these programs achieved participation and durability at scale, and not evidence that they measurably reduced risk. This research aims to move the conversation beyond surface-level analogies between the two domains—in particular the comparison between Software Bills of Materials (SBOMs) and their physical trade counterparts. While the term SBOM was borrowed directly from trade terminology, the two instruments function quite differently in practice and the analogy breaks down under scrutiny.

Through a combination of literature review and select interviews with trade security and OSS security practitioners, this research enabled the creation of a detailed framework for actionable recommendations. This framework focuses on collaborative efforts and positive incentives for developing and maintaining software security, recognizing that policies which 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

It is commonly acknowledged that supply chains are only as strong as their weakest link; understanding how this axiom plays out across OSS ecosystems helps clarify why policy interventions modeled on trade security may prove effective. A number of major security incidents have brought the issue of software

security into policy conversations globally. The European Union Agency for Cybersecurity's 2026 Threat Landscape report found that supply chain attacks continued to cause major security incidents, and concluded that growing digital dependencies are expanding the overall attack surface [7]. Artificial intelligence is expected to add to this burden, accelerating both the discovery of vulnerabilities and new forms of attack against open source maintainers (see Section 6.6). The software supply chain encompasses all the materials, components, processes, people and channels that it takes to develop a software product [8]. This includes not only code but all the third-party libraries used as dependencies, which include 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. Securing the software supply chain therefore requires examining every layer of the development and delivery process [9]. Protecting source code is not sufficient, since an attacker can compromise any weak link in the supply chain to get access to key systems. Unlike compromises in physical supply chains—where malicious intent can more easily be contained to a specific area—software supply chain attacks inject malicious code into applications, allowing attackers to rapidly infect all downstream users [10]. For example, third-party libraries often carry numerous dependencies, and the impact of a security flaw can be inherited by a vast number of downstream applications [11]. 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 [12].

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, in which malware is disguised as a legitimate program to gain access to a system. Attributed to a Russia-backed group, the attackers compromised the SolarWinds Orion build pipeline, inserting malicious code directly during the software compilation process rather than exploiting a dependency or the final application [1]. In September 2019, this group gained access to the platform's internal systems by compromising employee credentials, and between March and June 2020 distributed trojanized Orion updates to roughly 18,000 customers, including government agencies, critical infrastructure providers, and private corporations [13]. The implanted malware, known as Sunburst, created backdoors in infected networks, allowing attackers to spy on SolarWinds Orion customers and gain access to many critical IT systems in companies and government agencies using this platform.

The **Log4j** zero-day vulnerability was a different kind of supply chain failure—highlighting the concept 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 [14]. 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 determine whether they were affected by this vulnerability and how to fix it. The disclosure led to a worldwide effort to locate and patch vulnerable systems. The US Department of Homeland Security's Cyber Safety Review Board later described Log4j as an "endemic vulnerability," concluding that vulnerable instances would remain in systems for perhaps a decade or longer because of how deeply the library is embedded in the software supply chain [15].

The impact of **the XZ Utils** attack 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 ostensibly a 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. Subsequently, the attacker made periodic, subtle changes that appeared harmless but introduced the backdoor functionality. On March 29, 2024, another member of the community discovered the malicious code and immediately reported it [16]. 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 [17].

2.2 Response to Incidents and Policy Changes

These three incidents highlighted the breadth of software supply chain threats: SolarWinds illustrated the compromise of a trusted process; Log4j demonstrated the exploitation of a flawed component; and the XZ Utils attack showed 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. 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 known vulnerabilities, and, in component analysis, are used as sources to identify risks in new software. The three commonly used databases are the Common Vulnerabilities and Exposures database (CVE) [18], the Common Weakness Enumeration (CWE) [19], and the National Vulnerability Database (NVD) [20].

2. **Static analyzers** are automated tools that scan source code to identify potential security vulnerabilities before software is deployed. Two widely used examples are CodeQL [21], which integrates with GitHub - the world's largest code hosting platform - to run queries that detect specific security issues in a codebase, and Semgrep [22], which analyzes code based on customizable rules that can be tailored to a project's specific security requirements.

3. **Software Bills of Materials (SBOMs)** provide a formal inventory of all the components, libraries, and dependencies that make up a software product. Among the most widely used tools for generating these inventories are Syft, Trivy [23], and Microsoft's SBOM Tool [24], which typically produce output in standard machine-readable formats that allow the information to be shared and analyzed across different systems [25]. Complementary efforts include Software Identification (SWID) tags, which provide unique identifiers for individual components, and Vulnerability Exploitability eXchange (VEX) documents, which help clarify whether a reported vulnerability in a component actually poses a risk in a specific software product [26].

4. **Integrity tools** verify that software has not been altered or tampered with between the moment it was created and when it reaches the end user. **Sigstore** [27] works like a digital notary, providing cryptographic signatures that confirm a piece of software is authentic and unchanged. **TUF** (The Update Framework) [28] protects software update systems from attacks, ensuring that when users download an update it is legitimate and has not been intercepted or modified. **in-toto** [29] takes a broader approach, recording who performed each step in the software development and delivery process — creating a verifiable chain of custody from the developer's keyboard to the end user, similar in concept to the provenance documentation used in physical supply chains. Reproducible builds complement these tools by allowing anyone to rebuild a released artifact from its source code and confirm that it matches what was shipped, enabling independent verification of how the software was built [30].

5. **Software supply chain standards** provide common frameworks for measuring and improving security practices across the software development process. Two prominent examples are the Software Component Verification Standard (SCVS), created by OWASP, a widely respected open source security community, and SLSA (Supply-chain Levels for Software Artifacts), developed by Google and now maintained by the Linux Foundation. SCVS offers a broad framework for identifying security activities, controls, and best practices across the full software supply chain, while SLSA has a narrower focus on integrity—specifically, providing verifiable guarantees that software has not been tampered with between the moment it was written and when it reaches the end user [31].

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 unclear how to prioritize security actions, ensure integration among tools and standards, and measure the effectiveness of these actions.

The European Union's Cyber Resilience Act (CRA), proposed by the European Commission in September 2022 and adopted in October 2024, is the most significant regulatory effort to date affecting software security, and has drawn considerable attention from open source communities [32]. In the original draft, software developers were required to ensure the security of their code throughout its life-cycle, a requirement that was particularly problematic for open source 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 [33]. Sustained engagement by open source leaders during the legislative process produced substantial changes. The adopted text excludes open source software developed outside a commercial activity, and creates a new category of "open-source software steward" (Article 24) for organizations such as foundations that systematically support open source projects intended for commercial use. Stewards are subject to a lighter regime than manufacturers, centered on a documented cybersecurity policy and cooperation on vulnerability handling. Article 25 further empowers the Commission to establish voluntary security attestation programs for open source software, intended to support manufacturers' due diligence. Reporting obligations under the CRA have applied since September 2026, and its main obligations take effect in December 2027. In July 2026, the Commission published non-binding guidance on applying the regulation, including when open source software falls within its scope and when the lighter steward regime applies [34]. Concerns now center on implementation, including how attestation programs and supporting standards will be designed, and whether they will shift responsibility back onto open source developers [35].

The lack of open source community involvement in the original drafting of rules like the CRA highlighted the problem that can occur when key stakeholders are kept out of the development of policy around complex issues like software supply chains. In the case of the CRA, the open source community's concerns were largely addressed only after the initial draft had provoked widespread alarm. Addressing the current gaps in OSS security will require consistent involvement of all stakeholders from the outset, as well as a recognition from policymakers and regulators that punitive efforts may be less effective than efforts that create incentives for widespread adoption of supply chain security practices. The historical case study of trade security may provide guidance on the importance of enabling public-private collaboration for software-related policies moving forward.

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

The attacks of September 11, 2001 forced a rapid transformation in how governments and industry approached the security of global trade. The programs that emerged from that moment demonstrate that public-sector security goals and private-sector efficiency priorities can be aligned without resorting to punitive mandates.

Governments feared that terrorists would exploit vulnerabilities in cargo-based supply chains to smuggle weapons of mass destruction in the millions of ocean containers arriving at seaports each year [36], and even detonate one at a major port of entry [37]. 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 weak links in supply chains; early US strategies even included mandates to screen 100% of inbound cargo—an effort that proved impractical, given the sheer volume of cargo entering the country each day [38]. However, pragmatic, tailored and overlapping countermeasures quickly emerged, at multilateral and national levels. Engagement with all stakeholders reflected recognition that securing trade needed to be achieved without adversely impacting global commerce. These initiatives included incentive-based public-private partnerships such as Customs-Trade Partnership Against Terrorism (C-TPAT) and Authorized Economic Operator (AEO) programs, reinforced internationally through Mutual Recognition Agreements (MRAs) [39]. 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 [40].

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

Launched in November 2001, the US-based C-TPAT initiative aimed to maintain a balance between securing trade and preserving global commerce [5, 36]. 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. Drawing upon input from the private sector, CBP envisioned the C-TPAT initiative as a government-industry collaborative venture [41]. A voluntary program, C-TPAT was founded on the principle of shared responsibility and featured an incentive-based approach [42]. 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 [43]. In exchange for their commitment to enhancing global supply chain security, companies were certified as C-TPAT partners. Such designations carried tangible benefits. Partners enjoyed a reduced number of examinations, shorter wait times at the border, and access to Free and Secure Trade (FAST) lanes [5]. C-TPAT has a tiered structure, offering progressively greater benefits to partners who demonstrate a greater level of commitment to trade security [42]. This structure created 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 more than two decades after its launch. Congress gave the program a statutory basis in the SAFE Port Act of 2006 [44], and CBP reports more than 11,400 certified partners across the trade community [45]. Companies also advertise their C-TPAT status to business partners as a signal that they are well-positioned to facilitate smooth importation of goods into the US [46]. Member surveys commissioned by CBP found that participants valued C-TPAT beyond its direct cost savings, citing risk reduction, eligibility for contracts requiring membership, and the credibility it confers; in a 2020 assessment, about half of members reported that the benefits outweighed the costs, and another three in ten that they were roughly equal [47]. From a government perspective, the program remains an important tool in US strategy for securing global trade [48].

In a parallel international effort, the World Customs Organization (WCO) adopted the SAFE Framework of Standards in 2005, which established the concept of the Authorized Economic Operator (AEO); the WCO's AEO guidelines followed in 2006 and were fully integrated into the framework in 2007 [49]. AEO programs are equivalent to C-TPAT in their efforts to partner with the private sector and balance trade and security objectives. After the European Union implemented its AEO system in 2008, it 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 [50]. As of 2026, the WCO reports 92 operational AEO programs worldwide, with seven more under development [51]. 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 authorizations: AEOC for customs simplification, AEOS for security and safety, or a combined AEOC/AEOS version. The program is open to all supply-chain actors who meet the relevant criteria, including importers, exporters, manufacturers, and freight forwarders [52]. The AEO benefits include Mutual Recognition Agreements (MRAs), under which an AEO holder can request that the foreign customs authorities grant them comparable perks in the relevant jurisdiction, extending the facilitation benefits beyond the EU's borders. The EU currently holds MRAs with eight countries: Canada, China, Japan, Moldova, Norway, Switzerland, the US, and United Kingdom [53].

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

The AEO program contributed to a wider framework linking various elements 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) [54].

Reflective of Pillar 1, governments turned attention and resources to strengthening Customs-to-Customs partnerships. An example is the US Container Security Initiative (CSI), which was launched by US Customs and Border Protection (CBP) in January 2002, aimed at achieving global trade securitization through pre-screening of outbound cargo. These collaborative arrangements were 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 [36]. They sought to extend US security far beyond its physical borders to prevent threats from reaching US domestic soil. This pre-screening remains a voluntary undertaking, with the foreign government hosts and their customs authorities assisting CBP officers on-site. Like the other programs examined, CSI is designed to balance security with trade facilitation. Shipments cleared through a foreign CSI port undergo no further inspection upon arrival in the US, except for occasional random examination, and the program's reliance on non-intrusive screening technologies ensures minimal disruption to legitimate commerce [55]. Now in its third decade of operations, CSI has expanded steadily since its introduction and is now active at 61 ports across six continents. It is estimated that over 80% of all maritime containerized cargo imported into the United States is successfully pre-screened at foreign CSI ports [55].

3.3 Success Factors

The adoption and long-term institutionalization of these trade security programs—spanning multilateral, national, public, and private sector actors—was driven by several key factors. These factors include:

1. **Sense of shared urgency**: The immediate impact of the 9/11 terrorist attacks was felt world-wide, particularly in the transportation sector with the grounding of flights, including air cargo [56]. Fears of additional attacks drove action, and there was an expectation among governments as well as in the private sector that security across the global supply chain would require more attention [57].

2. **Incentivization for key stakeholders**: Unlike government agencies with dedicated security mandates, private sector actors are oriented primarily toward profitability and investor returns. Among these private sector actors, security-related compliance requirements may be seen as a sunk cost. Notably, the C-TPAT and AEO initiatives addressed this concern by featuring an opportunity for participating

companies to streamline importing processes; for C-TPAT, the financial incentive of quicker customs processing was the primary driver of participation [58]. 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 [59], and observers at the time warned that new security requirements could disrupt JIT delivery systems [4]. The incentives offered by the C-TPAT and AEO initiatives are directly connected with these priorities, resonating with importers and related companies due to tangible advantages for their bottom line. Incentivization also played a role in the international adoption of CSI. Host countries gained shared intelligence and a closer working relationship with US officials [58], as well as the reciprocal right to station their own officers at US ports to screen cargo bound for their countries, and participating ports gained a competitive advantage, since cargo shipped from CSI ports faced fewer delays entering the United States [4].

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 approaches 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 advance 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.

4. **Auditing and monitoring**: A central feature of C-TPAT is the validation of members' security practices: under the SAFE Port Act, CBP must validate certified members within one year of certification and revalidate them at least once every four years [47]. Related to alignment with existing systems, auditing was already a core component of the importing sector, as Customs periodically assessed Harmonized System Tariff Codes recorded by importers to verify that they applied appropriate duty rates.[1] In the same way that the importers' self-classification of goods has been subject to periodic reviews—thus keeping the community attuned to accurate classification practices – auditing and monitoring ensured that C-TPAT security practices were not treated as 'one-off' actions, to be completed but then forgotten. Rather, they became part of continuous operations—analogous to quality control in product development.

As a result of the above factors, C-TPAT and other programs achieved broad early adoption and have been sustained for more than two decades. Supply chain security has become part of a common lexicon shared by the global trade community and relevant government stakeholders. The sections that follow draw on these programs as models of adoption and institutionalization in developing a framework for open source software supply chain security.

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

The concepts and frameworks developed since 2001 to manage risk in the movement of tangible goods can provide useful models for thinking about the development and flow of software—with important limits. This section examines where the analogy holds, where it breaks down, and what those boundaries mean for developing an effective framework for open source software supply chain security.

[1]The Harmonized System classification used for tariff purposes is commonly referred to as "HS Codes." Goods declared for importation are classified using this internationally accepted system, and national customs authorities publish duty rates in tabular format indexed by HS Codes.

4.1 (S)BOMs—Similar Objectives but Different Objects

A shipping manifest, and its corresponding bill of materials, 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. In doing so, these documents give authorities and trade stakeholders the transparency needed to begin evaluating risk and conducting effective audits of tangible goods. Many draw a direct analogy between the shipping manifest and the increasingly popular reliance on SBOMs, looking to software inventories as a source of transparency and risk management [60]. However, drawing too many parallels between these two instruments is problematic, as it ignores the fundamentally divergent nature of tangible and digital goods. As one analyst notes, a software package that includes a third-party component "may incorporate it in its entirety, or select individual files, or functions, or even lines of code.... There is no atomic unit of software" [60]. This does not mean SBOMs have no place in the overall OSS security framework, but they should be seen as just one of many tools rather than a foundation for security standards. A shipping manifest typically describes a single layer of contents, while an SBOM must capture a deeply nested set of dependencies: a study of the npm ecosystem found that installing an average package introduces implicit trust in 79 third-party packages and 39 maintainers [61], creating a level of complexity far beyond that of a physical bill of materials. The accuracy of SBOM generation is therefore a significant challenge, as current tools can struggle to identify all dependencies [62].

4.2 Building Trust—From Trade to Code

One of the key C-TPAT components with direct implications for open source communities is the Trusted Trader program, which established a framework for certifying that a company's internal security processes meet a defined standard, qualifying it for benefits such as expedited processing. While not resolving all issues related to social engineering and community trust, an analogous voluntary certification system that rewards OSS projects which adhere to secure development best practices would mitigate several of the trust vulnerabilities currently affecting OSS security. Such a program would be grounded in established industry standards such as the NIST Secure Software Development Framework (SSDF) [63], the SLSA framework [64], or the OpenSSF Scorecard [65], and would follow the C-TPAT model of self-assessment followed by external validation [41]. The framework for such a program is developed in detail in Section 6.

4.3 Limits of Analogies

As the SBOM discussion above illustrates, tangible goods are static; software is dynamic. Software is not made in a single place like a factory; it 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. Even more than with cargo screening, creating systems of trust that incentivize self-assessments is essential in open source software development. The diverse nature of open source projects and their contributor communities further means that security frameworks cannot follow a one-size-fits-all approach, and must allow for differentiation across project types and community structures.

Two further limits deserve attention. First, the shared shock of 9/11 created rare alignment across government, industry, and the public that made rapid coalition-building possible. Any framework for OSS security must account for this absent forcing condition. Second, C-TPAT's logic, in particular, rests on the concept of a supplier with commercial obligations, a framing much of the OSS community actively rejects. Many open source developers object to being labeled suppliers at all, arguing that they have no contractual relationship with downstream users and cannot be held to obligations that were never agreed to. A Trusted Source

program must therefore be designed with this tension in mind. Section 6 addresses these limits directly —proposing preconditions for building the political and institutional foundations that C-TPAT benefited from, and designing a framework around voluntary, community-driven participation rather than supplier obligations.

5 Research Methodology

5.1 Bridging Two Expert Communities

This research was conducted in three phases. The first phase comprised parallel literature reviews of post-9/11 trade security programs and the current OSS supply chain security landscape, the findings of which were synthesized into a preliminary paper presented in November 2025 [66, 67]. The second phase extended that foundation through a series of expert interviews designed to validate the preliminary findings and refine the resulting recommendations. The third phase brought the two streams together into the comparative analysis and framework presented in this report. The research design reflects the interdisciplinary nature of the project's central question: because the analogy between trade security and OSS supply chain security crosses two practitioner communities with limited overlap, neither literature review alone nor expert consultation within a single domain could adequately test whether the lessons identified are both analytically sound and practically actionable. The interview phase was therefore designed to stress-test the preliminary framework against the lived experience of those who built and operated the trade security programs under study, and those working on the front lines of OSS security today.

5.2 Reaching Across Domains: Expert Interviews

The project team conducted interviews as well as informal discussions among two distinct expert groups. The first group comprised trade security practitioners, including current and former government officials, trade compliance professionals, and legal practitioners with direct experience advising on C-TPAT and related programs. The second group comprised OSS and software supply chain security practitioners, including OSS foundation and community leaders, corporate security practitioners, security tooling experts, and academic researchers working at the intersection of cybersecurity policy and open source communities. Candidates in both groups were identified through the project team's professional networks, which span both domains. All participants were informed of the research context and the non-attribution policy for published outputs prior to participation.

Two formal interview instruments were developed, one for each practitioner group, with both converging on the project's central research question and the key findings identified in the first phase. For the trade security track, this meant probing the factors behind the adoption and long-term institutionalization of programs such as C-TPAT, AEO, and CSI—particularly the role of incentive structures, public-private partnership dynamics, and auditing mechanisms—and asking practitioners directly what they saw as the most transferable lessons for analogous initiatives in other sectors. For the OSS security track, the instrument moved from practitioners' assessment of current challenges and existing tools to their reactions to the trade security analogy, including whether incentive-based "trusted actor" models could be meaningfully adapted for the OSS context. Trade security interviewees were not expected to have any background in software security; OSS security interviewees were provided brief contextual framing on the trade security programs before being asked to engage with the analogy directly.

5.3 Integrating Research and Interview Evidence

Most formal interviews were recorded and transcribed. Qualitative analysis was conducted using a structured codebook organized around seven analytical categories: (1) OSS challenges; (2) current tools and practices;

(3) role of government and regulation; (4) reactions to the trade security analogy; (5) incentive-based models; (6) recommendations and future directions; and (7) emergent themes. The final phase of the project brought together the interview findings with the literature review data to develop the comparative analysis and framework of recommendations presented in this report. The analytical approach centered on identifying structural parallels and meaningful differences between the trade security and OSS security domains.

The interview sample, while expert and deliberately cross-disciplinary, is not exhaustive, and the views of participants may not represent the full range of perspectives within either community. The OSS security landscape is also rapidly evolving, and some developments in tooling, regulation, and community practice may not be fully reflected in the analysis. The interdisciplinary scope of this research required working across substantial domain boundaries; specialists in any single area may identify nuances that a comparative approach necessarily compresses. A total of nine formal expert interviews were conducted for this research: six with software supply chain security practitioners (Experts 1–6) and three with trade security professionals with direct experience implementing C-TPAT and equivalent programs (Experts 7–9). To protect the anonymity of participants, all experts are referenced throughout the text by number rather than name. A description of each expert's professional background and area of expertise—without identifying information—is provided in the appendix.

6 Framework for Action: Recommendations Grounded in Expert Evidence

The following presents actionable lessons and final recommendations informed by our expert interviews and grounded in the historical analysis of C-TPAT and related programs presented earlier in this report. It is important to note that the lessons we draw 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.

Three design principles from C-TPAT and other secure trade approaches run through every recommendation that follows. First, security adoption is most durable when participation is *voluntary and incentive-driven*: companies joined C-TPAT because the operational benefits outweighed the compliance costs, not because they were compelled. Second, *graduated, tiered standards*—rather than a binary pass/fail—allow a diverse population of actors to enter at an appropriate level and improve continuously. Third, *burden must be proportionate to benefit*: the program asked the companies profiting from the supply chain to invest in securing it, not the individual workers who kept it running. Crucially, this last principle establishes the targeting logic that governs this entire framework: incentive-based approaches are designed for volunteer-based open source communities, while proportionate regulatory obligations fall on commercial actors who profit from those communities. For the communities and contributors, the relevant incentive is not market advantage but increased capacity such as funded security work, tooling, and validation, recognized at the level of the project. These two approaches are meant to be complementary, not contradictory.

6.1 Preconditions for Success: Building Trust, Awareness, and Cooperation

A key lesson from the trade security case study is that the successful approaches did not emerge from a blank institutional slate. Before 2001, the Customs Modernization Act of 1993 set the stage for cooperation between Customs and industry ("the trade community"), shifting the relationship from adversarial to cooperative, built around the concept of "reasonable care." Expert 7 captures the program's spirit: "we can't search every container, so we give [industry] more responsibility." Expert 8 reinforces this: the shift from a "gotcha" enforcement model to a client-based approach—with dedicated client managers assigned to major importers—meant that industry had already experienced government as a partner before it was asked to act like one.

The same logic applies to OSS policy: regulatory discussions like the EU's Cyber Resilience Act proceeded with limited consultation of open source communities, and were perceived as punitive policies as opposed to collaborative endeavors. This does not mean punitive regulation and incentive-based approaches are inherently incompatible; the problem with the CRA was targeting, not mechanism. Liability on commercial actors who profit from open source can redirect private resources toward volunteer-maintained infrastructure, making the two approaches complementary tools operating on different parts of the ecosystem.

Expert 3 identifies the deeper process failure: the role of government should be to "become a meaningful stakeholder in open source software communities rather than try to transliterate policy frameworks that exist in closed-source environments." Expert 1 agrees: "the most important thing any government can do before writing laws and regulations is making sure they understand the circumstance, and they're talking to the folks who are directly involved." Expert 9, who was involved in the original design of C-TPAT, illustrates why partnership had to precede requirements: in the earliest design discussions, a government agent demanded that one major retail company guarantee 100% security across 15,000 vendors in 84 countries. The lesson Expert 9 drew was clear: "if we leave them to themselves and we don't have input into the process, they're gonna make a program, but it's gonna be onerous and costly."

A Trusted Source program, or any certification-based framework, is unlikely to achieve C-TPAT-scale adoption unless government has first built genuine relationships with the open source ecosystem. As Expert 2 notes, engagement must begin as contribution and support, not compliance burden; the communities that need to be secured are already operating at the edge of their capacity. As Expert 3 observes, "you need a lot more meaningful understanding of the reputational and trust dynamics that lend credibility to work in the open source space"—a condition that cannot be satisfied by regulatory mandate alone.

A particular challenge arises in decentralized, volunteer-maintained scientific open source ecosystems—a category that shares characteristics with both individual maintainers and open source foundations, but whose distinct coordination structures may warrant separate attention. Projects such as NumPy [68], SciPy [69], NetworkX [70], and scikit-image [71] are coordinated through lightweight, consensus-based processes rather than formal foundations, and together underpin mission-critical infrastructure at national laboratories, observatories, and biomedical research institutions worldwide. In these ecosystems, there is often no single institutional relationship for policymakers to build; engagement must be earned project by project, and any certification or compliance framework must be designed to work *with* existing ecosystem-coordination mechanisms—such as the Scientific Python Ecosystem Coordination (SPEC) process [72], through which shared technical standards are developed and endorsed across dozens of independent projects—rather than alongside or over them. Moreover, the layered, federated structure of these supply chains means that a compromise at the foundation level propagates through hundreds of downstream packages, none of which controls what happens upstream. Certification schemes that address only individual projects leave this cross-project vulnerability unaddressed; policymakers should therefore engage with ecosystem-level coordination bodies before any compliance machinery is introduced.

One precondition from trade security is harder to manufacture: the political urgency that made C-TPAT and other programs possible. Expert 3 observed: "post 9/11, you have like a galvanizing moment. . . . I don't think this is happening right now. . . not in a way that would allow. . . this very closely knit collaboration between the industry and governments." This view is consistent with a widespread organizational tendency to underestimate exposure. Expert 4 describes it: "there's too many organizations that are just saying 'well, we're not the slowest gazelle in the pack. The cheetah is going to catch and eat someone else more likely than not. And anything we spend to be any faster than other people, it's wasted.'" Policymakers cannot manufacture a crisis, but they can make structural risk more prominent—through public reporting on supply chain incident costs, coordinated risk assessments, and transparency about commercial dependence on volunteer-maintained infrastructure.

Alongside relationship-building, education and awareness emerged as one of the most consistently cited priorities across the interviews. Expert 1 identified it as the single most important shift the field needs: "start with the basics; in all the computer science curricula right now, there is a big lack of security awareness." Expert 2 echoed this, arguing for "minimum basic security awareness, uniformly taught to all computer science engineers." Multiple experts framed this as a "socio-technical problem": the tools to secure the software supply chain largely exist, but awareness of why and how to use them does not. Expert 8, from the trade security side, pointed to Congressional hearings on TikTok [73] as evidence of what happens when decision-makers lack basic literacy about the industries they regulate. Expert 7's account of the Customs Modernization Act tells the same story from the other direction; Customs had to learn how industry operated before it could design requirements that industry would accept. Investment in security awareness—for developers, enterprise procurement teams, policymakers, and open source communities—is therefore a precondition for any other element of this framework to work.

6.2 Incentivizing Security-Minded Communities

Security adoption is most effective and sustainable when driven by incentives rather than punitive regulation. Approaches like C-TPAT reframed security investment as a pathway to competitive advantage—faster border crossings, fewer inspections, lower operational costs [74]. Voluntary participation was reinforced by the disadvantages of non-participation (higher inspection rates, unpredictable delays), creating a strong pull without mandating it. Expert 9 noted that one large retail company found enough internal efficiencies during C-TPAT compliance that "the program was almost self-funded."

Expert 4 makes the parallel explicit for software: "SBOMs are being adopted not because the government says 'you must have an SBOM' but because 'you have to have an SBOM to sell to the government'." Access-based incentives create value rather than mere compliance obligations. A successful software supply chain security program should therefore be built around a value proposition centered on positive incentives:

- **Liability safe harbors** for projects and adopters adhering to best practices, an approach the 2023 US National Cybersecurity Strategy proposed for software producers that securely develop and maintain their products, drawing on established practices such as the NIST SSDF [75].
- **Preferential treatment in public and private sector procurement**: governments and large enterprises should explicitly favor certified projects in their procurement decisions, creating a direct market incentive that extends beyond the public sector—already working in practice as companies adopt SBOMs to qualify for US government contracts.
- **Reduced cyber insurance premiums** for certified projects and their adopters: as Expert 4 noted, "insurance companies are actually much better at dealing with risk and understanding risk" than government bodies, and cyber insurance rewards genuine security improvement rather than checkbox compliance.

A clear market signal linking secure development practices to greater adoption can elevate the security posture of the entire ecosystem [76], though its impact will be uneven across projects where market forces play a limited role.

The incentive calculus is not uniform across actor types. Expert 8 found that the key question companies asked was: "can you tell us with confidence that, if we spend $1M on a C-TPAT compliance program, our competitors won't have an advantage by not participating?" For large-volume importers the answer was yes; for mid-size companies ROI calculations were less certain. The same differentiation applies to software: large technology companies and major open source foundations are well-positioned to engage with a certification program; individual volunteer maintainers and smaller projects are not. Expert 3 recommends targeting at the

project level, "where an open source product can be tagged in such a way that it receives favorable treatment," rather than applying uniform expectations across individuals and organizations. For volunteer-based open source projects, the incentives listed above offer little direct benefit; what these contributors and maintainers need is capacity and tools to increase project security and external validation that is not fully based on unpaid effort (see Section 6.7). Many maintainers see their contributions as a service to their community; their commitment is therefore closely tied to the health of the project and the people who depend on it. A project-level Trusted Source designation gives that collective commitment a visible form, one a community can take pride in much as fans rally around a home team. Therefore recognition still matters, but needs to be in a form that reflects how open source communities work.

The CSI's strategy of "pushing the border out" by screening high-risk cargo at origin offers a further lesson: security interventions are vastly more efficient when applied early than discovered at the point-of-delivery [77]. Applied to software, incentives should encourage integrating security into the earliest stages of the development life-cycle, treating the build and deployment pipeline as the critical port-of-origin for all software artifacts [42, 78].

6.3 Trusted Source: Tiered Programs for Recognizing Trusted Contributors

The cornerstone of a new software security strategy should be the creation of voluntary "Trusted Source" programs, providing clear incentives for open source projects and software producers to demonstrate their security practices. Applied to software, the same logic that underlies C-TPAT's Trusted Trader designation—voluntary adherence to defined security standards earns tangible operational benefits and a reputational market signal—would allow a certified project or enterprise to signal trustworthiness to procurement officers, security auditors, and downstream consumers.

Expert interviews converge on several principles for the program's design:

- **Tiered Structure**: Like C-TPAT, the program should offer graduated certification levels corresponding to increasing security maturity, with progressively greater benefits at each tier. This creates a continuous improvement incentive rather than a binary pass/fail dynamic. Existing frameworks such as SLSA levels, the OpenSSF Scorecard, and the NIST SSDF provide starting points for defining these tiers. More research would be needed to fully define the content of each tier, but they could plausibly run from automated baseline practices such as code review and branch protection, through independently attested build integrity, to full traceability, including reproducible builds and validated contributors. Two principles should govern any tier design. First, progression should depend on funded capacity, not unpaid effort: baseline practices that can be automated at little cost belong in the lowest tier, while the work required at higher tiers should be resourced through the mechanism described in Section 6.7. Because certification is continuous, funding must be sustained and not tied to one-time grants. Second, tiers should signal levels of verified, resourced assurance, and be careful to not be seen as a rating of project quality. A critical project at a lower tier may simply be under-resourced, and procurement preferences that steer users away from such projects would undermine the infrastructure the program aims to secure.
- **Differentiation by actor type**: The program should explicitly distinguish the primary categories of participant: commercial software vendors and large enterprises; open source foundations and ecosystems managing widely-used projects; and individual maintainers. Each category requires different entry points, baseline requirements, and incentive structures. Compliance burdens must fall predominantly on commercial actors. For volunteer-maintained projects, external validation could be carried out through community peer attestation and funded by those who depend on the project, through the mechanism described in Section 6.7. As Expert 1 notes, many OSS developers "really strongly

object to the word 'supplier'... 'I gave this to you freely. Why do you think that you can demand things from me? For a gift I gave you with no contract?' " The program's language, governance, and entry points must reflect the voluntary, community-driven nature of OSS contribution rather than importing supply chain assumptions.

- **Trusted contributors**: Alongside project-level tiers, the program should recognize validated contributors as a security control against the kind of social engineering seen in the XZ Utils attack. Validation would rely on the contributor having a sustained, verifiable track record, key-based identity, and some level of direct human connection – contributors who have met others in the community, whether in person or by video, and are known to them as people. Open source communities place high value on in-person gatherings in part because it allows for the building of these relationships. Support for community events is therefore a security investment, not only a community-building one.
- **Build on existing signals**: Expert 5 observed that vendors "would really like to just be able to have this stamp and say, look, you can trust me because I'm SLSA level three." A formal framework should build on these organically emerging signals rather than replace them.
- **Open source community co-development**: The program should be recognized and supported by designated agencies (CISA in the US, ENISA in the EU) but overseen by an international collaboration including government agencies, open source foundations, industry groups, and academic institutions. Open source foundations and communities should be structural participants in program governance, and the process genuinely co-developed with all relevant stakeholders.

This proposed approach recognizes concerns about voluntary attestation under the CRA's Article 25, raised by OpenSSF in January 2026, namely that attestation should not shift legal or financial responsibility upstream to open source developers or project stewards [35]. The Trusted Source approach would address this by treating attestation as a community activity, closer to a peer review process than to a commercial certification service. Attesters would be experienced members of the open source community, independent of the project under review and of its funders, who assess a project's practices as a form of service and potentially receive a modest stipend from the Trusted Source fund for doing so. The resulting attestation represents an expert judgment, but not a guarantee. It would not be a replacement for the due diligence the CRA places on manufacturers. Open source governance structures would be the basis for setting relevant standards, including managing conflicts of interest and overseeing the process. In this way the process avoids duplicating official certification schemes, such as those potentially prepared by ENISA. The Trusted Source program would instead be recognized as one of the voluntary attestation programs CRA Article 25 points to, adding community-governed review alongside the regulatory route. Internationally, it could serve as a common layer across jurisdictions, an option Section 6.4 takes up through mutual recognition.

6.4 International Standards and Mutual Recognition

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 [79]. Worldwide, customs administrations had concluded 126 bilateral and six plurilateral MRAs between AEO programs as of 2026 [51]. 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 has far-reaching implications for OSS supply chains, which are even 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. Expert 3 identifies fragmentation

and balkanization of norms across jurisdictions as one of the two primary challenges in OSS security today: diverging regulatory environments reduce the community engagement and resource pooling that are the backbone of open source's security model. Expert 8's practical experience guiding companies through both C-TPAT and AEO compliance programs across the US, Europe, and Asia provides firsthand confirmation that cross-jurisdictional certification is achievable at scale—and that when it works, it creates significant efficiency gains for all parties. Conversely, the parallel and non-harmonized demands of multiple certification regimes create compliance costs that fall hardest on the smallest actors.

The trade security case provides a clear roadmap for policymakers and the open source community: countries, regional groupings, and international organizations should work toward harmonizing standards for secure development lifecycles, agreeing on common formats and data fields for SBOMs, and establishing shared protocols for vulnerability disclosure. The goal should be to create a global framework of mutual recognition for national Trusted Source programs, mirroring the success of the MRA framework that linked C-TPAT and AEO programs internationally. Due to the community nature of open source, these efforts must include open source foundations and communities as active participants; their co-development of standards is as essential here as it is in the Trusted Source program design itself.

Based on Article 34 in the CRA, the EU can create mutual recognition agreements with third countries, taking into account their level of technical development and approach to conformity assessment [32]. Similar to customs-related MRAs, these agreements rest on mutual acceptance of the certificates and test reports issued by each party's conformity assessment entities. It remains unclear whether this mechanism could extend to the voluntary attestations envisaged under Article 25. If it could, a Trusted Source program recognized under Article 25 in the EU could be linked with counterpart programs elsewhere through the same kind of arrangement that connected C-TPAT and AEO, allowing attested open source projects to be recognized across jurisdictions without separate review in each. Clarifying this question should be a priority for all international partners.

6.5 The Mandate for Continuous Validation and Improvement

Any certification framework for OSS security must be a continuous process, and not just a one-time snapshot. Expert 5 notes that "there are still people every week that say that they've never heard about supply chain security before." The framework needs to recognize that continuous engagement is needed to shift this baseline. In C-TPAT, regular risk assessments and re-evaluations were required to maintain certified status, creating a virtuous cycle of improvement and preventing security postures from becoming stale. The January 2026 GAO report on C-TPAT found that 4% of participants were involved in security incidents over a five-year period, and that CBP had not consistently investigated those incidents or taken enforcement action—illustrating that participation alone does not guarantee compliance, and that ongoing auditing with clear enforcement criteria is essential even in a mature program [6].

Priorities for automated continuous validation include:

- **Pipeline attestations** that provide, in Expert 4's words, "an accurate representation of what actually happened," generating verifiable records at every build stage as a continuous process output rather than a retrospective audit; Expert 6 describes these as records that "link all of the actions and attribute them to who did it and how;"
- **Automated dependency management**, such as Dependabot and the built-in update mechanisms of Go and Rust, so that projects are alerted when a dependency becomes outdated or compromised rather than discovering it after the fact (Expert 5); and
- **Human review of contributor activity** for high-tier certified projects: Expert 4 emphasizes that basic

practices like code review before merge and branch protection rules remain far too rarely applied, and Expert 6's account of XZ Utils shows that automated tools alone would not have caught an attacker who built trust gradually over about two years of legitimate-looking commits.

Compliance burdens must not be prohibitive for volunteer-driven projects. Automated tooling that minimizes manual overhead is vital for equitable participation across project types.

6.6 Traceability as the Technical Frontier

Traceability emerged across both expert groups as an unsolved technical problem. Expert 8 observed that "the tech really hasn't evolved enough to get to true traceability" even in physical cargo; in software, where contributors are geographically dispersed and dependency chains run deep, the challenge is greater still. Yet traceability is the condition that would allow all other elements of this framework—Trusted Source certifications, continuous validation, burden-sharing, and international MRAs—to function.

The most promising current approach is provenance-based: frameworks like SLSA and in-toto record not just what is in a software artifact but who performed each step in its creation and how, linking all build-pipeline actions to identifiable, verifiable actors. This approach captures process integrity, not just component lists. As Expert 6 observed, "most of the major supply chain compromises are about lack of transparency." The three incidents discussed in Section 2.2 illustrate different forms of that gap: a compromised build process (SolarWinds), a flawed component that many organizations could not locate in their dependency trees (Log4j), and a compromised contributor (XZ Utils). Provenance and contributor validation address the first and third by recording who performed each step in an artifact's creation; component transparency, including accurate SBOMs, addresses the second. No single tool closes all three gaps, which is why traceability must combine these approaches.

Policymakers, funders, and industry practitioners should treat investment in provenance tooling and traceability infrastructure as a priority, including:

- Funding development of automated provenance generation and verification tools that integrate into common build systems with minimal friction;
- Establishing provenance documentation as a requirement within the Trusted Source framework, with higher tiers corresponding to more complete traceability coverage, including reproducible builds; and
- Supporting cross-ecosystem standards for provenance data formats, allowing provenance chains to be followed across language ecosystems, package managers, and organizational boundaries.

A forward-looking note on emerging threats is warranted, as artificial intelligence reshapes software supply chain risk. In February 2026, an autonomous AI agent published a personalized attack on a volunteer maintainer of a widely used Python library after he rejected its code contribution, an incident the maintainer described as an "autonomous influence operation against a supply chain gatekeeper" [80]. In the same period, an attacker used a prompt injection in a GitHub issue title to hijack an AI-powered triage bot in the Cline project, poisoned the project's CI cache, stole its publishing credentials, and released a compromised package that was downloaded roughly 4,000 times [81]. Provenance attestations for publishing, which the project adopted afterward, would have prevented that compromise [81]. AI is also accelerating vulnerability discovery, increasing the pace at which flaws are found in long-trusted dependencies [82].

These emerging threats are particularly dangerous in decentralized, volunteer-maintained ecosystems where maintainer capacity is thin and security governance is informal: an AI-enabled social engineering campaign targeting a volunteer maintainer is qualitatively harder to detect and respond to than the same campaign against a commercial vendor with dedicated security staff. The traceability and provenance infrastructure

described above, together with contributor validation (Section 6.3) and the ecosystem-level coordination bodies discussed in Section 6.1, provide the technical and governance foundation needed to detect and respond to these threats collectively. Provenance frameworks and continuous monitoring are the baseline defenses against the AI-accelerated attack classes now emerging. Progress on traceability is therefore a technical foundation on which this framework's long-term effectiveness depends.

6.7 Sharing the Burden: Commercial Responsibility for Open Source Security

Open source security is structurally under-resourced because those who bear the burden – volunteer maintainers – are not those who capture most of the value. Expert 6, who observed the XZ Utils incident firsthand, describes the failure directly: when a major technology company's employee found the backdoor, the company published a blog post and walked away, leaving one volunteer maintainer alone to spend weeks inspecting every commit the attacker had made, with no sustained support from the companies that depended on the library. As Expert 6 argues, those companies should either be required to help fix it or "give money to this poor guy who's maintaining like half of the compression libraries in the world so that he doesn't do this on the side and doesn't do it alone."

The C-TPAT model offers a direct analog: the program asked the companies profiting from the supply chain to invest in securing it. Expert 8 observed that C-TPAT-compliant companies gained not just regulatory benefits but supply chain visibility—"spillover effects" that created private value, not just public goods. Adapted to software, commercial actors who depend on open source must bear a proportionate share of security investment. Governments occupy a symmetric position: as one of the world's largest OSS consumers, they are subject to the same free-rider critique as commercial actors. Expert 2 put it plainly: "I wouldn't make a difference between public and private sector... you're all consuming open source. You should give back to it... money, expertise, support."

Proportionality must be a governing principle: obligation should scale with the volume and criticality of OSS dependence, just as C-TPAT compliance requirements scaled with import volume and supply chain complexity. Companies whose core products depend heavily on specific OSS components bear the greatest responsibility for securing them.

Table 1 summarizes the framework mechanisms, the actors they target, and their expected effects:

Actor targeted	Mechanism	Example	Expected effect
Large commercial dependents	Enterprise Stewardship Obligations	Require companies above a dependency-use threshold to contribute resources to security of relied-upon OSS	Closes the gap between value extraction and security investment
Foundations and community projects	Funded Foundations or Consortia	Government and industry co-fund a foundation-style entity for security around critical OSS; a pooled base allocated based on criticality that also allows direct contributions to specific projects.	Provides resources for security tooling, attestation, and governance without displacing volunteer-led contributions, including for critical but low-visibility projects.
Commercial actors earning revenue from OSS	Regulatory Leverage	Liability proportional to revenue derived from OSS [2]	Creates financial incentive for proactive commercial investment in OSS security
Entire ecosystem (commercial and public sector)	Reputational Accountability	Public scorecards rating companies on OSS security contributions relative to their consumption	Applies social and market pressure where direct regulation is infeasible

Table 1: Burden-sharing mechanisms: actor targeted, mechanism, example, and expected effect.

The funded foundation model in Table 1 also offers a way to resource the Trusted Source validation of volunteer-maintained projects. We propose a foundation-style fund, financed by industry and public funders, operating through two channels. A pooled base is allocated according to project criticality rather than visibility, so that deeply embedded transitive dependencies receive support even when few downstream users know they rely on them. Alongside the pool, a directed channel allows funders to support specific projects, letting the organizations most exposed to a given component signal where they see the greatest risk. Both channels feed a single attestation process carried out by expert attesters independent of the project (Section 6.3). Funders can choose where resources go, but not how projects are assessed, so maintainers do not face a separate review for each funder. Funding could support incentives like attester stipends as well as the higher-tier security work that unpaid effort cannot sustain, such as reproducible builds, end-to-end provenance, and contributor review. This approach is consistent with Article 25 of the CRA, which empowers the Commission to establish voluntary security attestation programs for open source software, in which developers, users, and other third parties can assess conformity with all or some of the regulation's essential cybersecurity requirements, and the CRA's non-binding preamble (Recital 21) anticipates that manufacturers, users, and public administrations may initiate or finance such attestation. A Trusted Source fund could serve as a vehicle for this third-party route, with independent peer attestation as a deliberate design choice beyond what the regulation requires.

The evidence on funding supports directing the fund toward attestation and specified security work rather than general grants. An early study of Germany's Sovereign Tech Fund found that funding increased development velocity, measured in commits, pull requests, and new issues, but had no significant effect on the number of contributors, releases, or closed issues; funding mobilized the work of existing maintainers rather than

[2]Expert 6: "you're going to get sued for 1% of your last year's profit. Well, why don't you spend 0.5% of that money making sure that open source developers are actually complying?"

expanding the community [83]. Similarly, an analysis of more than 4,600 repositories funded through Open Collective found that general funding did not appear to improve security posture as measured by the OpenSSF Scorecard, despite a modest correlation between the two. In that study, the authors conclude that targeted support such as audits, tooling, training, and dedicated developer time is more effective [84].

7 Conclusion: Securing the Future of Code

The complex challenge of tracking the software dependencies and potential vulnerabilities that underpin our global digital infrastructure is a source of profound risk to international security. The cybersecurity incidents mentioned in this report (SolarWinds, Log4j, XZ Utils) were just the most notorious cases highlighting the extent of current structural vulnerability, and demonstrating the real-world consequences of the weakest link problem. To mitigate the impact of software vulnerabilities, strengthening the full development cycle must be a priority for 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 rather than punitive enforcement. The core principles that guided the trade security efforts—trust through verification, transparency through documentation, shared responsibility, and the need to balance security with efficiency—are also important in any complex, distributed system. The expert interviews conducted for this research, spanning both software security practitioners and trade security veterans, suggest that these principles can translate: the strongest consensus was that incentives work better than punishment, that cooperative relationships must be built before compliance demands are made, and that the burden of security cannot fall on the volunteers who build the infrastructure that others profit from.

The framework developed in this report translates these principles into concrete recommendations. Building the cooperative and educational foundations first—before any certification or regulatory machinery is introduced—is the precondition that the trade security case study most clearly demonstrates, and the one most consistently identified by practitioners as missing from current OSS policy discussions. Trusted Source programs, tiered by actor type and grounded in existing community tools, offer an actionable path toward recognizing and rewarding good security practice without imposing impossible burdens on volunteer maintainers. Traceability, as both expert groups acknowledged, remains the unsolved technical problem that limits how far any framework can go—and investment in provenance infrastructure is therefore the highest-leverage technical priority. The threat landscape is also actively evolving, making this investment urgent rather than merely forward-looking. International harmonization, continuous validation rather than one-time compliance, and a rebalancing of commercial responsibility toward the companies that benefit most from open source are the remaining structural requirements for a framework that can work at scale.

Any pilot of these recommendations should build in evaluation from the outset. The impacts of public funding for open source remain poorly understood, with little consensus on how to measure them [85], and the early causal evidence is limited and mixed [83, 84]. A Trusted Source pilot should therefore define its intended outcomes, such as tier progression, attestation coverage, and changes in security practice, and establish baselines and comparison groups before funding begins, so that its effects can be distinguished from those of project maturity or existing resources.

The path forward requires a practical understanding of both the notable analogies and the critical differences between the tangible and digital 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 including awareness and education as necessary preconditions. 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 sustainable for the communities that make it work. The framework proposed here cannot manufacture the political urgency that made C-TPAT possible, but it can build the institutional and relational foundations that would allow rapid scaling when that moment arrives.

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] D. Goodin. (2024, Mar.) The XZ 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., Feb. 2004. [Online]. Available: https://unctad.org/system/files/official-document/sdtetlb20041_en.pdf

[5] U.S. Government Accountability Office, "Cargo security: Partnership program grants importers reduced scrutiny with limited assurance of improved security," GAO, Tech. Rep. GAO-05-404, 2005. [Online]. Available: https://www.gao.gov/products/gao-05-404

[6] ——, "Supply chain security: Actions needed to improve CBP management of the customs trade partnership against terrorism program," GAO, Tech. Rep. GAO-26-107893, 2026. [Online]. Available: https://files.gao.gov/reports/GAO-26-107893/index.html

[7] European Union Agency for Cybersecurity, "ENISA threat landscape 2026," ENISA, Tech. Rep., Sep. 2026. [Online]. Available: https://www.enisa.europa.eu/news/exploring-the-evolution-of-the-cyber-threat-landscape-how-dependencies-weaken-our-digital-resilience

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

[9] 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

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

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

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

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

[14] National Cyber Security Centre. Log4j vulnerability: What everyone needs to know. [Online]. Available: https://www.ncsc.gov.uk/information/log4j-vulnerability-what-everyone-needs-to-know

[15] Cyber Safety Review Board, "Review of the December 2021 Log4j Event," U.S. Department of Homeland Security, Tech. Rep., Jul. 2022. [Online]. Available: https://www.cisa.gov/sites/default/files/publications/CSRB-Report-on-Log4-July-11-2022_508.pdf

[16] A. Freund. (2024, Mar.) 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

[17] Red Hat. (2024) Urgent Security Alert for Fedora 40 and Fedora Rawhide Users. Red Hat Blog. [Online]. Available: https://www.redhat.com/en/blog/urgent-security-alert-fedora-40-and-rawhide-users

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

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

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

[21] GitHub. CodeQL. [Online]. Available: https://codeql.github.com

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

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

[24] Microsoft. SBOM tool. [Online]. Available: https://github.com/microsoft/sbom-tool

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

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

[27] 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

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

[29] 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

[30] Reproducible Builds. Reproducible builds. Accessed: 2026-10-27. [Online]. Available: https://reproducible-builds.org/

[31] 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.

[32] European Parliament and Council of the European Union, "Regulation (EU) 2024/2847 of the European Parliament and of the Council of 23 October 2024 on Horizontal Cybersecurity Requirements for Products with Digital Elements (Cyber Resilience Act)," Official Journal of the European Union, L series, Nov. 2024. [Online]. Available: https://eur-lex.europa.eu/eli/reg/2024/2847/oj

[33] Eclipse Foundation *et al.* (2023, Apr.) 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

[34] European Commission, "Commission guidance on the application of the Cyber Resilience Act, C(2026) 5252," Jul. 2026, published July 27, 2026. [Online]. Available: https://digital-strategy.ec.europa.eu/en/library/commission-publishes-new-guidance-support-timely-cyber-resilience-act-implementation

[35] M. Neag, "Preserving open source sustainability while advancing cybersecurity compliance: Reflections on voluntary attestation models under the Cyber Resilience Act," OpenSSF Blog, Jan. 2026. [Online]. Available: https://openssf.org/blog/2026/01/21/preserving-open-source-sustainability-while-advancing-cybersecurity-compliance/

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

[37] ——, "Homeland security: Summary of challenges faced in targeting oceangoing cargo containers," GAO, Tech. Rep. GAO-04-557T, Mar. 2004. [Online]. Available: https://www.gao.gov/assets/gao-04-557t.pdf

[38] C. Nolan, "The rubik's cube of cargo security: Is 100% screening of all U.S.-bound cargo containers prudent?" *The Brief*, vol. 42, no. 3, 2013, ABA Tort Trial & Insurance Practice Section.

[39] U.S. Department of Homeland Security, "DHS/CBP/PIA-013 customs-trade partnership against terrorism (C-TPAT)," U.S. Customs and Border Protection, Tech. Rep., 2013, appendix updated August 2021. [Online]. Available: https://www.dhs.gov/sites/default/files/publications/privacy-pia-cbp013-ctpat-august2021.pdf

[40] U.S. Government Accountability Office, "Maritime Security: One Year Later: A Progress Report on the SAFE Port Act," GAO, Tech. Rep. GAO-08-171T, Oct. 2007. [Online]. Available: https://www.gao.gov/assets/a118059.html

[41] 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, pp. 75–80, 2007.

[42] U.S. Government Accountability 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," GAO, Tech. Rep. GAO-08-240, 2008.

[43] 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, pp. 30–37, 2005.

[44] U.S. Congress, "Security and Accountability for Every Port Act of 2006 (SAFE Port Act)," Public Law 109-347, 2006.

[45] U.S. Customs and Border Protection. Customs trade partnership against terrorism (CTPAT). Accessed: 2026-10-27. [Online]. Available: https://www.cbp.gov/border-security/ports-entry/cargo-security/CTPAT

[46] Ocean Spray. CTPAT statement of support. [Online]. Available: https://www.oceanspray.com/ctpat-statement-of-support

[47] U.S. Customs and Border Protection, "Accountability and Transparency Features in the United States' Customs Trade Partnership Against Terrorism (C-TPAT) Program," *WCO News*, no. 104, Jun. 2024. [Online]. Available: https://mag.wcoomd.org/magazine/wco-news-104-issue-2-2024/accountability-and-transparency-features-ctpat/

[48] ——, "Customs trade partnership against terrorism: 2023 program impact report," CBP, Tech. Rep., 2024. [Online]. Available: https://cbp.gov/sites/default/files/2024-06/ctpat_2023_impact_report_508.pdf

[49] World Customs Organization. About: SAFE framework of standards and the AEO programme. Accessed: 2026-10-27. [Online]. Available: https://aeo.wcoomd.org/about

[50] European Court of Auditors, "Special report 13/2023: Authorised economic operators - solid customs programme with untapped potential and uneven implementation," European Court of Auditors, Tech. Rep., 2023. [Online]. Available: https://www.eca.europa.eu/ECAPublications/SR-2023-13/SR-2023-13_EN.pdf

[51] World Customs Organization. Authorized economic operator programme. Accessed: 2026-10-27. [Online]. Available: https://aeo.wcoomd.org/

[52] 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

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

[54] 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

[55] U.S. Customs and Border Protection. (2025) CSI: Container security initiative. Last modified September 30, 2025. Accessed: 2025-10-16. [Online]. Available: https://www.cbp.gov/border-security/ports-entry/cargo-security/csi/csi-brief

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

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

[58] N. Monacelli, "Improving maritime transportation security in response to industry consolidation," *Homeland Security Affairs*, vol. 14, no. 2, Jan. 2018. [Online]. Available: https://www.hsaj.org/articles/14257

[59] M. Holweg, "The genealogy of lean production," *Journal of Operations Management*, vol. 25, no. 2, 2007.

[60] A. Gantman. (2021, Sep.) SBOM: Good intentions, bad analogies, ugly outcomes. LinkedIn. Published September 30, 2021. [Online]. Available: https://www.linkedin.com/pulse/sbom-good-intentions-bad-analogies-uglyoutcomes-alex-gantman/

[61] M. Zimmermann, C.-A. Staicu, C. Tenny, and M. Pradel, "Small world with high risks: A study of security threats in the npm ecosystem," in *Proceedings of the 28th USENIX Security Symposium*, 2019, pp. 995–1010. [Online]. Available: https://www.usenix.org/conference/usenixsecurity19/presentation/zimmerman

[62] 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.

[63] 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.

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

[65] OpenSSF, "OpenSSF scorecard," https://scorecard.dev/, 2025, accessed: 2025-10-16.

[66] UCSC Open Source Program Office. UCSC OSPO participation in the open forum academy symposium. [Online]. Available: https://ucsc-ospo.github.io/post/2025ofapresentation/

[67] S. Lieggi *et al.*, "From containers to code: Applying post-9/11 trade security lessons to the software supply chain," 2025, available at https://ucsc-ospo.github.io/post/2025ofapresentation/.

[68] C. R. Harris, K. J. Millman, S. J. van der Walt *et al.*, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, 2020.

[69] P. Virtanen, R. Gommers, T. E. Oliphant *et al.*, "SciPy 1.0: Fundamental algorithms for scientific computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[70] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using NetworkX," in *Proceedings of the 7th Python in Science Conference (SciPy 2008)*, 2008, pp. 11–15.

[71] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias *et al.*, "scikit-image: Image processing in Python," *PeerJ*, vol. 2, p. e453, 2014.

[72] Scientific Python. Scientific python ecosystem coordination (SPEC) documents. Accessed: 2026-11-06. [Online]. Available: https://scientific-python.org/specs/

[73] House Energy and Commerce Committee. (2023, Mar.) TikTok: How Congress Can Safeguard American Data Privacy and Protect Children from Online Harms. [Online]. Available: https://www.congress.gov/event/118th-congress/house-event/115519

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

[75] The White House, "National cybersecurity strategy," Executive Office of the President, Tech. Rep., Mar. 2023. [Online]. Available: https://bidenwhitehouse.archives.gov/wp-content/uploads/2023/03/National-Cybersecurity-Strategy-2023.pdf

[76] Baroni Center for Government Contracting, "Creating market incentives to improve cybersecurity," George Mason University, Baroni Center Report 11, Jan. 2025. [Online]. Available: https://business.gmu.edu/news/2025-01/creating-market-incentives-improve-cybersecurity

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

[78] 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 *Proceedings of the 2022 ACM Workshop on Software Supply Chain Offensive Research and Ecosystem Defenses (SCORED)*, Los Angeles, CA, USA, Nov. 2022, pp. 15–24.

[79] U.S. Customs and Border Protection. Customs-trade partnership against terrorism mutual recognition. Accessed: 2026-10-27. [Online]. Available: https://www.cbp.gov/border-security/ports-entry/cargo-security/c-tpat-customs-trade-partnership-against-terrorism/mutual-recognition

[80] S. Shambaugh, "An AI agent published a hit piece on me," The Shamblog, Feb. 2026. [Online]. Available: https://theshamblog.com/an-ai-agent-published-a-hit-piece-on-me/

[81] grith, "A GitHub issue title compromised 4,000 developer machines," Mar. 2026. [Online]. Available: https://grith.ai/blog/clinejection-when-your-ai-tool-installs-another

[82] W. Woolwine, "AI is changing vulnerability discovery and your software supply chain strategy has to change with it," Rapid7 Blog, Apr. 2026. [Online]. Available: https://www.rapid7.com/blog/post/ai-changing-vulnerability-discovery-software-supply-chain-strategy/

[83] L. Domenech Burin, "Measuring the invisible: Evaluating the impact of public funding on open source software," arXiv preprint arXiv:2607.05413, 2026. [Online]. Available: https://arxiv.org/abs/2607.05413

[84] S. A. Brackett, S. Scott, and C. Cheng, "Buying security: Open source software funding and security posture," Center for Security, Innovation, and New Technology, American University, CSINT Policy Paper Series, Fall 2025. [Online]. Available: https://infrastructureinsights.fund/wp-content/uploads/2025/12/CSINTPolicyPaperNo22025BrackettScottandCheng.pdf

[85] C. Osborne, P. Sharratt, D. Foster, and M. Boehm, "A toolkit for measuring the impacts of public funding on open source software development," arXiv preprint arXiv:2411.06027, 2024. [Online]. Available: https://arxiv.org/abs/2411.06027

A Interview Expert Descriptions

The expert findings cited throughout this report draw on nine interviews conducted between September 2025 and April 2026. All experts are referenced anonymously as Experts 1 through 9 in accordance with the study's Institutional Review Board (IRB) protocol. The following descriptions summarize the professional background and expertise of each interviewee. No names or institutional affiliations are included, in accordance with the study's IRB anonymization protocol. Experts 1–6 are software supply chain security practitioners; Experts 7–9 are trade security practitioners with direct experience implementing C-TPAT and AEO programs.

Expert 1—Software Supply Chain Security Practitioner and Educator
A director-level practitioner leading open source supply chain security at a major international open source foundation, with nearly fifty years of experience in software development and security. Author of a widely-used secure development course. Combines deep technical expertise with direct engagement with government bodies and standards organizations including the EU Cyber Resilience Act process.

Expert 2—Open Source Security Lab Leader
A senior leader running security research operations at a major software development platform, responsible for securing the open source supply chain at scale. Brings approximately thirty years of software engineering experience, including a decade focused on supply chain security, with a platform-level view of how security practices are adopted across millions of repositories.

Expert 3—Technology Policy Researcher, AI and Open Source Governance
A researcher at an international policy think tank working on AI governance, open source software standards, and technology policy, with a focus on the Global South and voluntary governance mechanisms. Examines how OSS governance frameworks function across different regulatory environments and how fragmentation of international norms affects open source security and sustainability.

Expert 4—Academic Researcher and Supply Chain Security Tool Builder
An academic whose research focuses on building widely-used software supply chain security tools, all freely available under major open source foundations. Combines academic expertise with deep practical knowledge of how security tools are adopted in production, and holds a strong view that government liability frameworks are a necessary complement to voluntary security practices.

Expert 5—Research Scientist in Data Center and Cloud Security
A research scientist specializing in cloud and data center security, applying hardware and software security technologies to supply chain integrity for approximately six years. Expertise in key management, cryptographic signing, and dependency management tooling. Also an active open source contributor with insight into both enterprise and individual maintainer perspectives.

Expert 6—Open Source Security Researcher and Long-Tenured Package Maintainer
A security researcher with over a decade of experience maintaining packages in a major Linux distribution, combined with research focused on supply chain security. Brings a socio-technical framing to the field—the problem is not a lack of tools but a lack of unified frameworks for applying them— and advocates for provenance-based approaches and greater commercial responsibility for the open source components companies depend on.

Expert 7—Retired US Customs Special Agent and Trade Security Consultant
A retired agent from the US Customs Service who was serving on a multi-agency counter-terrorism task force on September 11, 2001, and observed the development and early implementation of C-TPAT. Implemented Customs Mutual Assistance Agreements with a foreign partner country and observed C-TPAT compliance

operationalized in a major US port district. Subsequently worked as a private sector trade security consultant.

Expert 8—Retired US Customs Analyst, Trade Compliance Consultant, and Author
A former US Customs analyst and investigator with over a decade of government service, followed by a career as a trade compliance consultant at a major professional services firm. Guided large companies through C-TPAT compliance in the US and AEO compliance programs in Europe and Asia, providing a cross-jurisdictional perspective on trusted trader frameworks. Also a published author on technology, trade, and geopolitics.

Expert 9—Corporate Trade Security Leader and C-TPAT Charter Member
A senior trade compliance professional with decades of experience in international trade, including roles as a licensed customs house broker and as a trade compliance leader at a major US retail corporation. One of the original charter members of C-TPAT, with direct involvement in the program's foundational design discussions with US Customs leadership in the weeks following September 11, 2001. Brings a private-sector perspective on the incentive structures and industry co-authorship that shaped C-TPAT's early architecture and drove its adoption among large-volume importers.

B Interview Instruments

The following interview instruments were developed to guide interviews with two distinct expert groups. Each instrument was designed around the project's central research question: what can open source and cybersecurity communities, as well as policymakers, learn about developing supply chain security and management guidelines from the experience of post-9/11 trade security efforts?

B.1 Trade Security Practitioners

Candidate Criteria

Interviewees were selected based on professional expertise in supply chain security or customs security programs, ideally with direct involvement in programs such as C-TPAT, AEO, CSI, or related national customs risk management initiatives. Candidates included former government officials from agencies such as DHS; trade compliance and supply chain risk management professionals from multinational corporations; academics specializing in customs security or trade facilitation; and legal practitioners and consultants with experience advising on trade security programs.

Interviewer Introduction

Participants were provided the following introduction prior to the interview:

> Thank you for taking the time to speak with me. This interview is part of a research collaboration between the University of California and the Middlebury Institute of International Studies. Our project examines what lessons from post-9/11 global trade security—including initiatives such as C-TPAT, AEO, and CSI—might help inform emerging approaches to software supply chain security. Although our project focuses on software supply chain security, you are not expected to have any background in software or cybersecurity for this interview. We are here to learn from your professional insights based on your experience with trade security systems. Before we begin, would it be all right if I record this interview? This will help me avoid taking detailed notes and allow me to stay fully engaged. Everything is confidential; your name, affiliation, and any identifying information will not appear in anything we publish.

Interview Questions

1. *Background and Experience*
   (a) Could you start by telling me about your professional experience with trade security programs?
2. *Successes and Challenges of C-TPAT, AEO, and CSI*
   (a) In your professional opinion, what have been the biggest successes of programs like C-TPAT, AEO, or CSI in enhancing trade security?
   (b) What are the biggest challenges these programs have faced in terms of implementation and industry participation?
   (c) Have these programs evolved over time in meaningful ways? What drove those changes?
3. *Incentive-Based Models*
   (a) C-TPAT and AEO rely heavily on voluntary participation and incentive-driven compliance. From your perspective as a trade security practitioner, what has made this model work, or not work?
   (b) How important have the incentives been to keeping participating companies engaged?

(c) Were there incentives you felt were missing or insufficient?

(d) What kinds of incentives mattered most to participants?

4. *Public–Private Partnerships*

   (a) Trade security programs after 9/11 were often structured as public-private partnerships. How would you describe the quality of collaboration between government and industry in conjunction with these programs?

   (b) What made those partnerships effective when they worked well?

   (c) In cases where they broke down, what typically caused the friction?

   (d) How did companies perceive the compliance requirements? Were they manageable, excessive, or varied across sectors and size?

5. *Future Outlook*

   (a) If you had to boil it down, what is the single most important lesson that the established trade security programs can offer similar public-private partnership-based initiatives in other sectors, such as open source software?

   (b) Looking ahead, what changes or reforms do you think would most strengthen trade security programs today?

   (c) What would it take to implement those changes—political will, resources, new technologies, or stronger industry buy-in?

B.2 OSS and Software Supply Chain Security Practitioners

Candidate Criteria

Interviewees were selected based on demonstrable, hands-on experience in software supply chain security or open source software (OSS) security. Candidates included corporate security and Open Source Program Office (OSPO) leaders; core maintainers and staff at major OSS foundations such as the OpenSSF, the Linux Foundation, and Apache; security tooling and vendor experts from companies building SCA, SBOM, and signing tools; and academic researchers working at the intersection of cybersecurity policy and open source communities.

Interviewer Introduction

Participants were provided the following introduction prior to the interview:

> Thank you for offering your time to meet with me. This is part of a research collaboration between the UCSC Open Source Program Office (OSPO) and the Middlebury Institute of International Studies. We are studying how lessons from post-9/11 global trade security might inform software supply chain security. You do not need to know anything about trade or those past efforts; we will explain that part as we go. What we are really interested in is your experience and perspective from the software supply chain side. Before we start, would it be okay if I record this interview? It will save me from needing to take notes and allow me to be fully present in this conversation. Everything will be kept completely confidential; your name, organization, or any other identifying details will not appear in anything we publish.

Interview Questions

1. Before we dive in, could you share a bit about your current role and how your work relates to software supply chain security?
2. From your perspective, what are the one or two biggest challenges in securing the software supply chain today, especially concerning open source software?
   (a) Could you share an example where one of those challenges became visible in your own work?
   (b) Where do you think the next big weakness is most likely to appear?
3. When you look at current practices for OSS security—such as SBOMs, SLSA, and Sigstore—which ones do you find most useful in practice?
   (a) How widely are these tools and practices being used where you work?
   (b) What do you see as the biggest gaps towards broader adoption?
4. What do you see as the appropriate role for the government in managing OSS security?
   (a) The EU's Cyber Resilience Act (CRA) is a mandatory regulation making manufacturers legally liable for a product's security for its entire lifecycle. What do you think is the right balance between direct regulation like the CRA, government standards, and market-driven approaches?
   (b) Do you think policies like these are actually helping improve security in practice? What is working, and what is not?
5. *Context provided to interviewee at this point:* After 9/11, governments were trying to make international trade more secure without slowing it down. A key initiative was the Customs-Trade Partnership Against Terrorism (C-TPAT), a voluntary public-private partnership in which companies that voluntarily audit their security practices and meet a high standard are certified as 'Trusted Traders.' In return, they receive significant business incentives such as expedited customs processing and far fewer inspections.

   Do you see any parallels for this 'Trusted Trader' model in the software world?
   (a) Do you think public-private partnerships where companies or projects are rewarded for good security practices can work in this space?
   (b) What kinds of incentives would be powerful enough?
   (c) What do you see as the biggest reasons it might fail, and are those reasons unique to the open source community?
6. When you think about where this field needs to go next, what is the single biggest shift that would improve software supply chain security?
   (a) What would it take to get there?