March 2023 · Dr. Alexandra Paulus & Christina Rupp

Government's Role in Increasing Software Supply Chain Security

A Toolbox for Policy Makers

Stiftung Neue Verantwortung

Think Tank at the Intersection of Technology and Society

Executive Summary

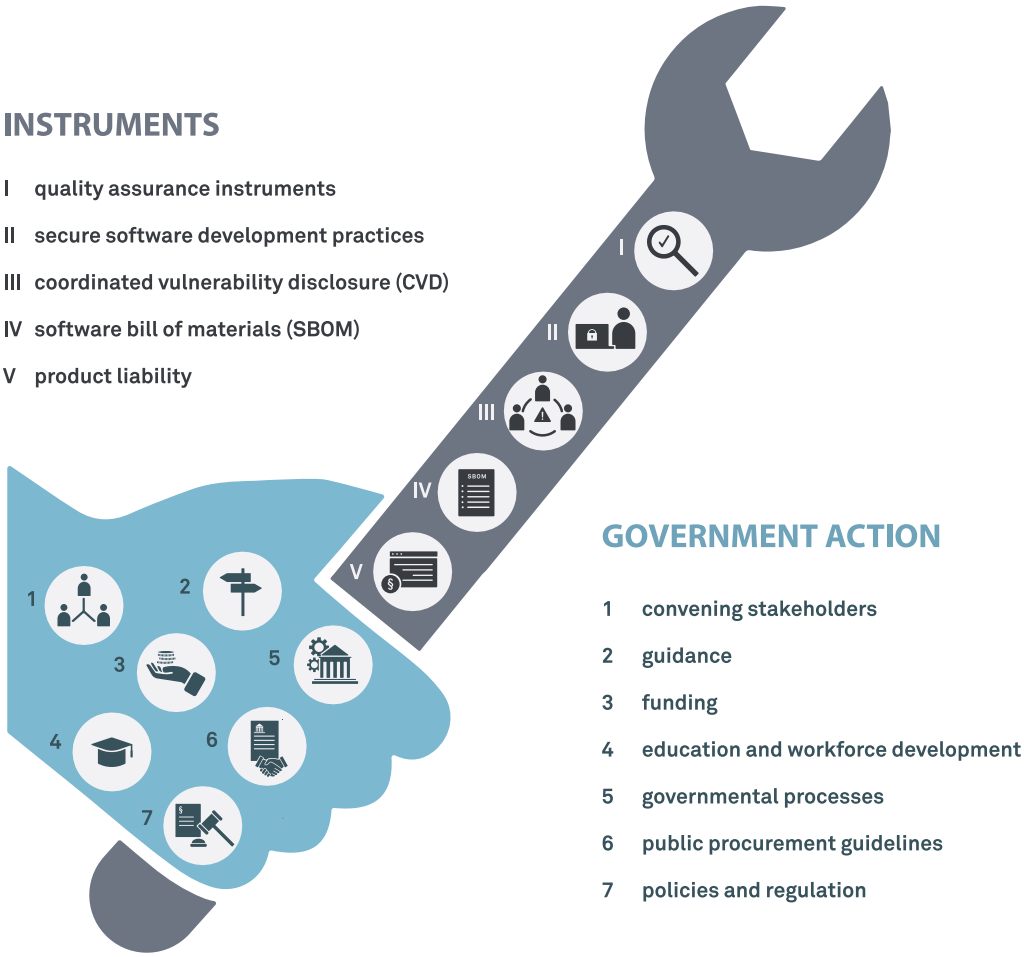
Software has become a cornerstone of the systems that are essential to modern society. Like many other products, software is the result of complex international supply chains. There are unique characteristics of the software-developing ecosystem that make software supply chains particularly vulnerable: First, software-developing entities rely significantly on software components developed, delivered, and maintained by others. Besides this, software-developing entities often do not prioritize security when developing their products, which is why software-using entities often find it difficult to assess the security of a given software based solely on the information provided to them. In addition, the supply chain can be compromised at different stages of the software development life cycle, which makes securing software supply chains a tough challenge.

Software supply chain compromises such as SolarWinds or Log4Shell can have large-scale impact: an initial compromise of one entity in the chain violates the confidentiality, integrity, and availability of data further down the software supply chain, often affecting multiple organizations and sectors across national borders. Software supply chain compromises have led to, inter alia, ransomware operations on software-using entities and to the unauthorized access to sensitive customer data and proprietary source code. The perpetrators of such compromises are malicious actors with criminal, political and economic espionage, and sabotage objectives.

Given these threats, software supply chain security poses an urgent problem to policy makers. For too long, the issue has been seen mainly as a problem for vendors to resolve. But recently, policy makers have started to recognize that this field is also ripe for policy interventions, as shown by the US president's 2021 Executive Order 14028 on Improving the Nation's Cybersecurity and by the European Commission's 2022 draft of the Cyber Resilience Act. Still, the possibility for government action for increasing software supply chain security extends further than the elements of these initiatives.

In this analysis, we develop a toolbox that combines diverse instruments with targeted government action to be practical guidance for policy makers. This toolbox approach has the advantage that policy makers can choose instruments suited to their respective positions, considering, for example, available resources and capabilities. After reviewing the instruments and the possibilities for government action, we have compiled three sets of policy priorities that policy makers should

focus on, providing three levels of ambition that cater to different national venture points.



Level 1 – Basics First: Any government interested in increasing software supply chain security should make use of three instruments. First, they should include secure software development practices in software developer education and in workforce development efforts. Second, they should issue guidance for organizations on how to set up organizational policies for coordinated vulnerability disclosures (CVDs). In such policies, software-developing entities clarify the process that occurs between the receipt of information about a vulnerability that others (often security researchers) have found in their software and the provision of a remediation (often a software patch). Third, governments should issue guidance specifying data formats for software bills of materials (SBOMs) and identifying technical tools building on SBOM

data (SBOMs list the components and supply chain relationships of a given piece of software.) These three actions can draw on broad and existing best practices and can be implemented with limited resources and capabilities and in a short timeframe.

Level 2 – Ambitious but Tried and Tested: Governments who want to take additional steps should focus on four actions. First, they should convene the national and international stakeholders involved in quality assurance instruments – that is, technical standards, quality assurance schemes, and product security labeling schemes – to allow for the coordination and exchange of good practices. Second, governments should issue guidance, tailored to the needs of different types of organizations including small and medium enterprises (SMEs), on how to implement secure software development practices. Third, governments should adapt processes to require software-developing government agencies to develop and publish organizational CVD policies as well as require such policies from organizations supplying the public sector through public procurement guidelines. Fourth, governments should convene stakeholders to discuss challenges and solutions regarding SBOM use. These four actions can be implemented with limited resources and capabilities and within short to medium timeframes; they have also been implemented in different jurisdictions already.

Level 3 – Breaking New Ground: Governments who want to lead the way in increasing software supply chain security could explore five further actions. First, they could fund assessments of the effects of quality assurance tools on SSC security. Given positive results, they could fund the development of new technical standards or the adaptation of existing and relevant ones and establish a national – and ideally internationally harmonized – conformity assessment scheme and product security labeling scheme for software. Second, governments could develop regulation mandating software-developing entities to implement secure software development practices. Third, governments could develop a national legal framework for CVD, which would require software-developing entities to put in place organizational CVD policies. Fourth, governments could fund the development and refinement of SBOM data formats and technical tools that build on

SBOM data and develop regulation mandating SBOM use; for instance, starting with suppliers of critical infrastructure providers. Fifth, governments could develop a product liability regime that covers software (or amend an existing one to include software). When developing any regulation in this field, policy makers need to consider how such regulation will affect SMEs and individual developers, who typically have fewer resources and may therefore be disproportionately affected by regulatory burdens. These five ambitious actions would require significant resources and high capabilities and would take time to implement. In some cases, since good practices are not yet available, those actions would break new ground.

Whichever level policy makers choose, policies on software supply chain security will often be most effective when international coordination and cooperation are considered from the start. Emblematic examples of this include the harmonization of regulation on CVD to facilitate cross-border vulnerability disclosure or the international coordination of public procurement guidelines. In many cases, like-minded coalitions will provide the most promising starting point for international dialogue on these issues. Progress on this issue can also contribute to advancing multilateral cyber diplomacy. As all UN member states have already agreed back in 2015 that governments should take steps to increase the security of software and hardware supply chains, it is high time that policy makers act on this commitment by developing concrete national policies.

This project was made possible by the generous support of the *German Federal Foreign Office*. The views expressed in this paper do not necessarily represent the official position of the ministry.

Acknowledgements

This analysis was supported by the cyber diplomacy working group *Government's Role in Increasing Software Supply Chain Security* through online collaboration and a joint virtual workshop.

The views and opinions expressed in this paper are those of the authors and do not necessarily reflect the official policy or position of the working group members and of their respective employers.

We acknowledge the essential contributions of the following, in alphabetical order:

Mary Brooks, The Wilson Center

Sonia Compans, ETSI

Philip Engelmartin, SAP SE

Sven Herpig, Stiftung Neue Verantwortung

Trey Herr, Atlantic Council

Stefan Hessel, reuschlaw

Bart Hogeveen, Australian Strategic Policy Institute

So Jeong Kim, Institute for National Security Strategy

Andreas Kuehn, Observer Research Foundation America

Lim May-Ann, Fair Tech Institute, Access Partnership

Jiro Minier, DCSO

Nadine Nagel, Stiftung Neue Verantwortung

Stefan Saatmann, Siemens AG

Tara Tarakiyee, Sovereign Tech Fund

Kaylin Trychon, Chainguard

Chris Wysopal, Veracode

Table of Contents

Abbreviations and Acronyms

BSI	Bundesamt für Sicherheit in der Informationstechnik (German Federal Office for Information Security)
CISA	US Cybersecurity and Infrastructure Security Agency
CNCF	Cloud Native Computing Foundation
CSAF	Common Security Advisory Framework
CSIRT	computer security incident response team
CVD	coordinated vulnerability disclosure
DIN	Deutsches Institut für Normung (German Institute for Standardization)
DOC	US Department of Commerce
ENISA	European Union Agency for Cybersecurity
ETSI	European Telecommunications Standards Institute
EU	European Union
FIRST	Forum of Incident Response and Security Teams
GGE	UN Group of Governmental Experts on Developments in the Field of Information and Telecommunications in the Context of International Security
ICTs	information and communications technologies
IETF	Internet Engineering Task Force
IoT	Internet of Things
ISO	International Organization for Standardization
JPCERT/CC	Japan Computer Emergency Response Team Coordination Center
MSP	managed service provider
NGO	non-governmental organization
NIS2 Directive	EU Directive on measures for a high common level of cybersecurity across the Union
NIST	US National Institute of Standards and Technology
NTIA	US National Telecommunications and Information Administration
OECD	Organisation for Economic Co-operation and Development
OpenSSF	Open Source Security Foundation
OWASP	Open Web Application Security Project
SAFECode	Software Assurance Forum for Excellence in Code
SBOM	software bill of materials

SDLC	software development life cycle
SDO	standards development organization
SLSA	Supply-chain Levels for Software Artifacts
SMEs	small and medium enterprises
SSC	software supply chain
SSDF	Secure Software Development Framework
UN	United Nations
UNGA	United Nations General Assembly
UNIDIR	United Nations Institute for Disarmament Research
UNIDO	United Nations Industrial Development Organization
US	United States
VEX	Vulnerabilities Exploitability eXchange

1 Introduction

Software has become a cornerstone of systems that are essential to modern societies, such as critical infrastructure[1] and food production.[2] Similar to many other products, software is the result of complex international supply chains. However, software supply chains (SSCs) differ from supply chains of many other products in that in the former, compromises of one entity can lead to potentially large-scale violations of the confidentiality, integrity, and availability of data further down the chain. Software-using entities have had their systems targeted with ransomware,[3] have had sensitive customer data accessed,[4] or have witnessed unauthorized access to their proprietary source code[5] following SSC compromises.

Similar to compromises of non-software supply chains, compromises of SSCs can potentially impact different organizations and sectors across national borders. For instance, the 2021 SSC compromise of the company Kaseya affected, inter alia, supermarkets in Sweden, schools in New Zealand, municipalities in the United States (US), and hospitals in Romania.[6]

There are three root causes of SSC compromises in the software-developing ecosystem: Software-developing entities rely significantly on software components developed, delivered, and maintained by others; they often do not prioritize security when developing their products; and software-using entities often find it difficult to assess the security of a given software.

Foresight experts at the European Union Agency for Cybersecurity (ENISA) recently projected that the "supply chain compromise of software dependencies"[7] would be the top cybersecurity threat by 2030. The prevalence of such compromise is likely related to the fact that diverse malicious actors, from criminals to military intelligence, can exploit weaknesses in the SSC to achieve their objectives, as shown in the following examples:

1 Elena Jharko (2021): Ensuring the Software Quality for Critical Infrastructure Objects, in: IFAC PapersOnLine 54 (13), pp. 499-504.
2 BSA Foundation (2019): Every Sector Is a Software Sector: Agriculture. Agricultural Opportunity Is Growing With Software.
3 Kellen Browning (20.07.2021): Hundreds of Businesses, From Sweden to U.S., Affected by Cyberattack, The New York Times.
4 Ax Sharma (04.05.2021): Twilio discloses impact from Codecov supply-chain attack, Bleeping Computer.
5 Confluent (2021): Confluent Update Regarding Codecov Incident; Rapid7 (2021): Rapid7's Response to Codecov Incident.
6 Ellen Nakashima and Rachel Lerman (21.09.2021): FBI held back ransomware decryption key from businesses to run operation targeting hackers, The Washington Post; Kellen Browning (20.07.2021): Hundreds of Businesses, From Sweden to U.S., Affected by Cyberattack, The New York Times. See Section 2.2 for more details on this incident.
7 European Union Agency for Cybersecurity (ENISA) (2022): Cybersecurity Threats Fast-Forward 2030: Fasten your Security-Belt Before the Ride!

1. SSC compromises can be the financially motivated work of criminals, as in the Kaseya incident, when a criminal hijacked the company's update process to spread ransomware.[8]
2. They can serve political espionage purposes: the SolarWinds/SUNBURST incident, in which a software program used by many US government entities was compromised, was later attributed to a Russian intelligence service.[9]
3. Compromises can also serve economic espionage purposes, as in the case of a Chinese tax software program that contained malware.[10]
4. Such operations can satisfy sabotage objectives, such as the NotPetya incident: by leveraging the software program of a Ukrainian accounting software provider, the malware spread worldwide and permanently encrypted devices, so the data stored on them could not be recovered.[11]

In short, SSCs are vulnerable, there are many bad actors who want to exploit them, and their exploitation can have large-scale impact. Therefore, SSC security poses a particularly urgent problem to policy makers. Way back in 2015, all United Nations (UN) member states agreed that "[s]tates should take reasonable steps to ensure the integrity of the supply chain"[12] of information and communications technology (ICT) products in a key cyber norms document. However, the issue has long been seen mainly as a problem for vendors to solve, as the software ecosystem has historically been subject to only modest government action.[13] As a result, for several years, there was little progress in norm implementation, that is, in translating the aforementioned abstract commitment into policy.

Yet, since 2021 – with SSC compromises on the rise –, several policy initiatives have addressed the issue. For instance, the European Union (EU) issued the Directive on measures for a high common level of cybersecurity across the Union (NIS2 Directive);[14] proposed the EU Cyber Resilience Act draft legislation, which strives to provide a "uniform legal framework for essential cybersecurity requirements for placing products with digital elements on

8 Charlie Osborne (23.07.2021): Updated Kaseya ransomware attack FAQ: What we know now, ZDNet. See Section 2.2 for more details on this incident.
9 FireEye (2020): Highly Evasive Attacker Leverages SolarWinds Supply Chain to Compromise Multiple Global Victims With SUNBURST Backdoor, Mandiant.
10 Brian Hussey (2020): The Golden Tax Department and the Emergence of GoldenSpy Malware, Trustwave; Federal Bureau of Investigation (2020): FBI Flash Alert Number AC-000129-TT.
11 Andy Greenberg (22.08.2018): The Untold Story of NotPetya, the Most Devastating Cyberattack in History, Wired.
12 United Nations General Assembly (UNGA) (2015): Group of Governmental Experts on Developments in the Field of Information and Telecommunications in the Context of International Security (A/70/174).
13 Paris Call for Trust and Security in Cyberspace, Cigref, Kaspersky, and GEODE (2021): Securing ICT supply chains.
14 Directive (EU) 2022/2555 of the European Parliament and of the Council of 14 December 2022 on measures for a high common level of cybersecurity across the Union, amending Regulation (EU) No 910/2014 and Directive (EU) 2018/1972, and repealing Directive (EU) 2016/1148 (NIS 2 Directive) [2023] OJ L333/80.

the Union market";[15] and published the EU Council conclusions on ICT supply chain security, which call for the creation of an "ICT supply chain toolbox"[16] to reduce supply chain risks of ICTs, including those stemming from software.

Finally, policy makers are also addressing the topic through domestic interventions. In 2021, the US President issued Executive Order 14028 on *Improving the Nation's Cybersecurity*,[17] which contains, inter alia, public procurement guidelines for software for the US federal administration. In 2022, the Cyber Security Agency of Singapore launched the *Critical Information Infrastructure Supply Chain Programme*[18] to mitigate the supply chain risk of software and other ICT products.

Key Terms[19]

A **software supply chain** is made up of the people, resources (particularly software artifacts and infrastructure), and processes in the network of the developing entity of a given software; its suppliers (including the people who maintain and contribute to the components of the software in question), and distributors, retailers, system integrators, service providers, and users of the software. **Software-developing entities** include companies, government agencies,[20] non-governmental organizations (NGOs),[21] and individuals involved in the development of software. Not all of these operate commercially, such as non-profit organizations, foundations, and volunteer networks.[22] **Software users** can include other companies, government agencies, NGOs, and individuals. A **software supply chain compromise** is an operation in which the confidentiality, integrity, and/or availability of people, resources, or processes of an entity is violated with the goal of compromising, via software, the confidentiality, integrity, and/or availability of people, resources, or processes of one or more entities further down the SSC.

15 European Commission (2022): Proposal for a regulation of the European Parliament and of the Council on horizontal cybersecurity requirements for products with digital elements and amending Regulation (EU) 2019/1020.
16 Council of the European Union (2022): Council conclusions on ICT supply chain security.
17 The White House (2021): Executive Order 14208, "Improving the Nation's Cybersecurity".
18 Cyber Security Agency of Singapore (2022): Critical Information Infrastructure Supply Chain Programme Paper.
19 All definitions are explained in more detail in Section 2.
20 Cybersecurity and Infrastructure Agency (2023): GitHub; Federal Office for Information Security (BSI) (2023): GitHub; National Cybersecurity Agency of France (2023): GitHub.
21 Mozilla Corporation (2023): Firefox Browser; The Apache Software Foundation (2023): HTTP Server Project.
22 Some people are critical of the term software supply chain (SSC) due to its commercial connotation (see Iliana Etaoin (2022): There is no "software supply chain"). Nevertheless, we use this term as it has become the established reference in policy circles, which form the key target audience of this paper, but we remain mindful of the particular situation of non-commercial software-developing entities throughout this analysis.

Scope and Objective of this Analysis

This is hardly the first analysis of SSC security. Others have studied the security of the supply chain of ICTs more broadly,[23] compiled datasets,[24] and analyzed the risk landscape[25] of SSC compromises. Policy recommendations on SSC security often either focus on software-developing entities and users as their audience[26] instead of policy makers, or are mostly national in scope.[27]

In contrast, the objective of this analysis is to develop an SSC security toolbox for policy makers made up of instruments that policy makers can use – through different government actions – to increase SSC security. The components of the toolbox are based on technical evidence,[28] since they can be implemented in a transparent and verifiable way, and include instruments that are either already in use in some jurisdictions or are new proposals that have yet to be translated into national policy. We concentrate on policy recommendations that are ready to use in the near to medium future instead of developing long-term ideas.[29]

23 Ariel (Eli) Levite (2019): ICT Supply Chain Integrity: Principles for Governmental and Corporate Policies, Carnegie Endowment for International Peace; EastWest Institute (2020): Weathering TechNationalism; EastWest Institute (2016): Purchasing Secure ICT Products and Services: A Buyers Guide; Nele Achten (2021): Governance Approaches to the Security of Digital Products. A Comparative Analysis, ETH Zurich; Paul Rosenzweig and Benjamin Wittes (2022): How Can One Know When To Trust Hardware and Software?, Lawfare; Paris Call Working Group 6, Cigref, Kaspersky and GEODE (2021): Securing ICT supply chains; Oleg Demidov and Giacomo Persi Paoli (2020): Supply Chain Security in the Cyber Age: Sector Trends, Current Threats and Multi-Stakeholder Responses, United Nations Institute for Disarmament Research (UNIDIR); Organisation for Economic Co-operation and Development (OECD) (2021): Enhancing the digital security of products. A policy discussion.

24 Cloud Native Computing Foundation (CNCF) (2023): Catalog of Supply Chain Compromises, GitHub; Dan Geer, Bentz Tozer, and John Speed Meyers (2020): For Good Measure. Counting Broken Links: A Quant's View of Software Supply Chain Security, in: Login: 45 (4), pp. 83-86; ENISA (2021): Threat Landscape for Supply Chain Attacks; IQT Labs (2023): Software Supply Chain Compromises - A Living Dataset, GitHub; Trey Herr, Nancy Messieh, June Lee, Will Loomis, and Stewart Scott (2020): Breaking trust: The dataset, Atlantic Council.

25 ENISA (2021): Threat Landscape for Supply Chain Attacks; Sonatype (2022): 8th Annual State of the Software Supply Chain Report; The MITRE Corporation (2023): Supply Chain Security. System of Trust Framework.

26 Cybersecurity and Infrastructure Security Agency (CISA) (2021): Defending Against Software Supply Chain Attacks; ENISA (2021): Threat Landscape for Supply Chain Attacks; Stacy Simpson (ed.) (2010): Software Integrity Controls. An Assurance-Based Approach to Minimizing Risks in the Software Supply Chain, Software Assurance Forum for Excellence in Code (SAFECode).

27 Trey Herr, William Loomis, Stewart Scott and June Lee (2020): Breaking trust: Shades of crisis across an insecure software supply chain, Atlantic Council.

28 This stands in contrast to measures based on political evidence, such as limiting market access to suppliers for political reasons (see US Department of Commerce (DOC) (2022): Implementation of Certain 2021 Wassenaar Arrangement Decisions on Four Section 1758 Technologies), or mere self-declarations of conformity or trustworthiness (see Zweites Gesetz zur Erhöhung der Sicherheit informationstechnischer Systeme [Second Law on increasing the security of information technology systems] (18.05.2021), Bundesgesetzblatt 2021 I (25), pp. 1122-1138).

29 Examples of this would be new international bodies or an entirely changed ecosystem of building software.

This leads us to focus on five instruments that, together with the respective government actions, make up the toolbox:[30]

1. Three quality assurance instruments (technical standards, conformity assessments, and product security labeling schemes);
2. Secure software development practices;
3. Coordinated vulnerability disclosure (CVD);
4. Software bill of materials (SBOM); and
5. Product liability.

Since SSC security is an essentially international issue, we not only consider national government action but also – where appropriate – point to prospects for international coordination and cooperation.

Structure of this Analysis

Following this introduction, this analysis has four parts. First, we sketch the vulnerability of SSCs, explaining what the SSC and SSC chain compromises are, when they can occur during the software development life cycle (SDLC), and what are their possible root causes. Second, we present the toolbox for increasing SSC security, structured according to the five aforementioned instruments. For each instrument, we describe the instrument and explain how it can contribute to increasing SSC security. Then, we describe the room for government action – how the government can directly implement it or foster its implementation among software-developing entities. Finally, we identify which government action(s) policy makers should focus on. In this section, which is the center of this study, we also review international experiences with each instrument and government action, as applicable. Third, we explain why SSC security should be part of efforts to implement cyber norms and that it is therefore relevant to cyber diplomats. Finally, we summarize our findings in the conclusion and present three sets of priority government actions that cater to diverse government ambitions, resources, and capabilities.

30 Among other means of increasing SSC security are, for example, strengthening open-source security, strengthening operational security, and establishing regional transparency centers.

2 The Vulnerability of Software Supply Chains

To understand what policy makers can do to increase software supply chain (SSC) security, we need to understand why SSCs are vulnerable to compromises in the first place. To answer this question, we examine, first, the structure of SSCs; second, the nature of SSC compromises and their relationship to the software development life cycle; and third, the root causes of these compromises.

International initiatives and policies present a wide range of definitions of SSCs and of SSC compromises.[31] Yet, clear definitions of these terms – which scope the problem of SSC security – are necessary to formulate targeted policy recommendations. Therefore, we briefly review key existing definitions and, building on them, we present our own definitions, on which we base our recommendations.

2.1 The Software Supply Chain

According to the US National Institute of Standards and Technology (NIST), a **supply chain** can be defined as a "[l]inked set of resources and processes between and among multiple levels of organizations, each of which is an acquirer, that begins with the sourcing of products and services and extends through their life cycle."[32] The definition of the SSC is less consensual than this, as Table 1 shows, also because some definitions refer not to the SSC but to the ICT supply chain or the cybersecurity supply chain.

31 Paris Call for Trust and Security in Cyberspace, Cigref, Kaspersky, and GEODE (2021): Securing ICT supply chains.
32 Jon Boyens, Angela Smith, Nadya Bartol, Kris Winkler, Alex Holbrook, and Matthew Fallon (2022): Cybersecurity Supply Chain Risk Management Practices for Systems and Organizations.

CISA	"The ICT supply chain is the network of retailers, distributors, and suppliers that participate in the sale, delivery, and production of hardware, software, and managed services."[34]
ENISA	"In cybersecurity, the supply chain involves a wide range of resources (hardware and software), storage (cloud or local), distribution mechanisms (web applications, online stores), and management software."[35]
Atlantic Council	"The flow of goods, data, and finances related to software and systems delivery."[36]
Chainguard	"Software supply chain activities involve the transformation of dependencies, packages, components, binaries, build and packaging scripts, code and other software artifacts, and infrastructure into a finished software deliverable that is deployed into production. Participants in the supply chain include actors like developers, reviewers, testers, and maintainers who are working on the product at hand, but also [include] those who maintain and contribute to packages and package managers, and other software that may be incorporated into a given product. Software supply chains also include information relevant to the software, such as versioning, signatures, and hashes."[37]
SAFECode	"The IT system supply chain is a globally distributed and dynamic collection of people, processes and technology. Software is one component of a larger IT solution and each software vendor is only one part of a complex chain of suppliers, systems integrators and ultimate end users."[38]

Table 1: Overview of definitions of the software supply chain[33]

Building on these definitions, for the purposes of this paper, we arrive at the following definition: a **software supply chain** is made up of the people, resources (particularly software artifacts and infrastructure), and processes in the network of the developing entity of a given software, its suppliers (including the people who maintain and contribute to the components of the software in question), and distributors, retailers, system integrators, service providers, and users of the software.

Figure 1: A notional software supply chain

As illustrated in Figure 1, an SSC may encompass several actors, as many software-developing entities are also among the users of the software developed by other software-developing entities (and so forth). The more complex a (software) supply chain, the less visibility, understanding, and control

33 CISA – Cybersecurity and Infrastructure Security Agency; ENISA – European Union Agency for Cybersecurity; SAFECode – Software Assurance Forum for Excellence in Code.
34 CISA (2021): Defending Against Software Supply Chain Attacks.
35 ENISA (2021): Threat Landscape for Supply Chain Attacks.
36 Trey Herr, William Loomis, Stewart Scott and June Lee (2020): Breaking trust: Shades of crisis across an insecure software supply chain, Atlantic Council.
37 Chainguard (2023): Chainguard Glossary. Software supply chain security vocabulary.
38 Stacy Simpson (ed.) (2010): Software Integrity Controls. An Assurance-Based Approach to Minimizing Risks in the Software Supply Chain, SAFECode.

of the complete supply chain each organization in it is likely to have.[39] As a result, SSC risks may originate from each supplier, their respective supply chains, and their software.[40] However, an SSC may also be simpler, with all entities – except for the software-developing entity and the software users – optional (in Figure 1, the necessary components are shaded in dark blue, and the optional ones, in light blue).

2.2 Software Supply Chain Compromises

There are diverse definitions of an SSC compromise, as presented in Table 2. A comparison of these definitions showed that they differ significantly in scope. Therefore, we present our own definition, which encompasses a wide variety of compromises that policy makers can address using diverse instruments.

Table 2: Overview of definitions of a software supply chain compromise[41]

CISA	"A software supply chain attack occurs when a cyber threat actor infiltrates a software vendor's network and employs malicious code to compromise the software before the vendor sends it to [its] customers. The compromised software then compromises the customer's data or system. Newly acquired software may be compromised from the outset, or a compromise may occur through other means like a patch or hotfix. In these cases, the compromise still occurs prior to the patch or hotfix entering the customer's network."[42]
ENISA	"A supply chain attack is a combination of at least two attacks. The first attack is on a supplier that is then used to attack the target to gain access to its assets. The target can be the final customer or another supplier. Therefore, for an attack to be classified as a supply chain one, both the supplier and the customer have to be targets."[43]
NIST	"A cybersecurity incident in the supply chain (also known as [a] compromise) is an occurrence within the supply chain whereby the confidentiality, integrity, or availability of a system or [of] the information the system processes, stores, or transmits is jeopardized. A supply chain incident can occur anywhere during the life cycle of the system, product or service."[44]
Atlantic Council	"A software supply chain attack occurs when an attacker accesses and modifies software in the complex software development supply chain to compromise a target farther down on the chain by inserting [its] own malicious code."[45]
CNCF	"Software supply chain attacks occur when the materials or processes of producing software are themselves compromised, resulting in vulnerabilities targeting downstream consumers of the software produced."[46]

39 Jon Boyens, Angela Smith, Nadya Bartol, Kris Winkler, Alex Holbrook, and Matthew Fallon (2022): Cybersecurity Supply Chain Risk Management Practices for Systems and Organizations.
40 Jon Boyens, Angela Smith, Nadya Bartol, Kris Winkler, Alex Holbrook, and Matthew Fallon (2022): Cybersecurity Supply Chain Risk Management Practices for Systems and Organizations.
41 CISA – Cybersecurity and Infrastructure Security Agency; ENISA – European Union Agency for Cybersecurity; NIST – US National Institute of Standards and Technology; CNCF – Cloud Native Computing Foundation.
42 CISA (2021): Defending Against Software Supply Chain Attacks.
43 ENISA (2021): Threat Landscape for Supply Chain Attacks.
44 Jon Boyens, Angela Smith, Nadya Bartol, Kris Winkler, Alex Holbrook, and Matthew Fallon (2022): Cybersecurity Supply Chain Risk Management Practices for Systems and Organizations.
45 Trey Herr, William Loomis, Stewart Scott and June Lee (2020): Breaking trust: Shades of crisis across an insecure software supply chain, Atlantic Council.
46 CNCF (n.d.): Software Supply Chain Best Practices.

For the purposes of this paper, a **software supply chain compromise** is defined as an operation in which the confidentiality, integrity, and/or availability of people, resources, or processes of an entity is violated with the goal of compromising, via software, the confidentiality, integrity, and/or availability of people, resources, or processes of one or more entities further down the SSC. As a result, one compromise can affect a wide range of targets, often crossing borders.

There are many kinds of SSC compromises. To understand the problem they pose, we review examples of past such compromises. Rather than presenting them in isolation, however, we show how they relate to the process of software development so that we can point out how specifics in the software ecosystem may lead to compromises. To this end, we rely on the software development life cycle (SDLC) model. It is often used to illustrate the process of software development,[47] including – in the SSC security context –[48] by the US Cybersecurity and Infrastructure Security Agency (CISA)[49] and ENISA.[50] We use the SDLC model employed by ENISA, in which the SDLC has the following six phases.[51] For each phase, we provide a brief description and, in four cases, an example of an SSC compromise, as illustrated in Figure 2.

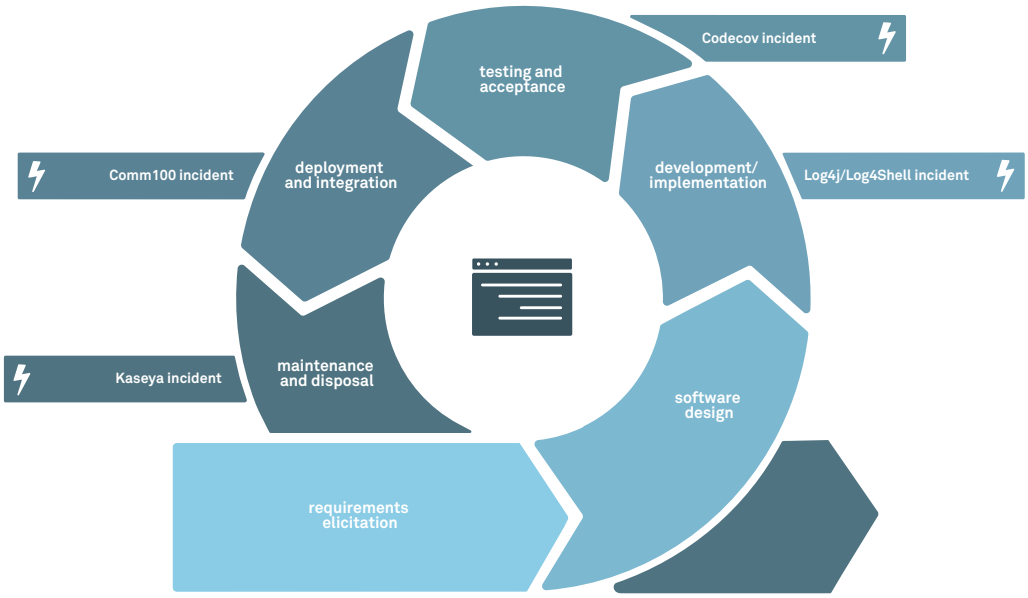


Figure 2: Compromises throughout the software development life cycle

47 Ian Sommerville (2011): Software Engineering, 9th edition. Pearson Education; Youssef Bassil (2012): A Simulation Model for the Waterfall Software Development Life Cycle, in: International Journal of Engineering & Technology, 2 (5).
48 Trey Herr, William Loomis, Stewart Scott and June Lee (2020): Breaking trust: Shades of crisis across an insecure software supply chain, Atlantic Council.
49 CISA (2021): Defending Against Software Supply Chain Attacks.
50 ENISA (2020): Advancing Software Security in the EU.
51 ENISA (2020): Advancing Software Security in the EU.

1. **Requirements Elicitation:** System specifications are identified, including goals, services, and constraints.[52]
2. **Software Design:** The requirements identified in the first phase are translated into a system architecture.[53] This involves designing algorithms, software architecture, database concepts, logical diagrams, concepts, graphical user interfaces, and data structures.[54]
3. **Development/ Implementation:** The requirements and design are realized "into a concrete executable program, database, website, or software component through programming and deployment."[55] In this stage, for example, vulnerable versions of open-source components may be integrated into a software, as in the Log4j/Log4Shell incident (see info box below).

The Log4j/Log4Shell incident

Log4j is a popular open-source logging library for Java applications developed within the framework of the non-profit Apache Software Foundation.[56] In November 2021, the security team of the Chinese cloud computing company Alibaba Cloud reported a Log4j vulnerability to the Apache Software Foundation.[57] Before the vulnerability was published in the US National Vulnerability Database, researchers referred to it as "Log4Shell"[58] and later suggested that it had existed since 2013.[59] The vulnerability in Log4j's Java Naming and Directory Interface allowed for remote code execution if users inserted targeted input into log messages, which permitted them to gain access to or even take complete control of a target server.[60] The Log4j library is widely used in an extensive range of software, so its vulnerability had security ramifications far down the SSC. Entities that used software that relied on the Log4j library saw, for instance, their devices used for cryptomining or creating botnets[61] or targeted by ransomware after the Conti group had exploited the vulnerability to gain access to user systems.[62]

52 Ian Sommerville (2011): Software Engineering, 9th edition. Pearson Education.
53 Ian Sommerville (2011): Software Engineering, 9th edition. Pearson Education.
54 Youssef Bassil (2012): A Simulation Model for the Waterfall Software Development Life Cycle, in: nternational Journal of Engineering & Technology, 2 (5).
55 Youssef Bassil (2012): A Simulation Model for the Waterfall Software Development Life Cycle, in: nternational Journal of Engineering & Technology, 2 (5).
56 Apache Software Foundation (2023): Apache log4j 1.2.
57 Edmund Brumaghin (2021): Threat Advisory: Critical Apache Log4j vulnerability being exploited in the wild, Talos.
58 Free Wortley, Chris Thompson, and Forrest Allison (2021): Log4Shell: RCE 0-day exploit found in log4j, a popular Java logging package, LunaSec.
59 Hannah Murphy (14.12.2021): Hackers launch over 840,000 attacks through Log4J flaw, ars technica.
60 Apache Software Foundation (2023): Apache Log4j Security Vulnerabilities.
61 Tushar Richabadas (2022): Threat Spotlight: Attacks on Log4Shell vulnerabilities, Barracuda; Liam Tung (03.03.2022): Log4Shell flaw: Still being used for crypto mining, botnet building... and Rickrolls, ZDNET.
62 Jonathan Greig (17.12.2022): Log4j: Conti ransomware attacking VMware servers and TellYouThePass ransomware hits China, ZDNET.

4. **Testing and Acceptance:** The objective of iterative testing is to verify and validate that a program meets its requirements and specifications.[63] It also involves debugging, "in which bugs and system glitches are found, corrected, and refined accordingly."[64] Each iteration tests and improves the software build. The Codecov incident (see info box below) is an example of a compromise that took advantage of the testing phase from the viewpoint of the software users.

The Codecov incident
US company Codecov provides testing tools for software-developing entities. In early 2021, unknown actors took advantage of a faulty internal process at the company whereby a docker image held the credentials to a data collection tool, Bash Uploader, so the perpetrators were able to exfiltrate the credentials.[65] With the access thus provided, they modified the script of Bash Uploader, whose function was to send customers' software testing reports to Codecov. The modified version also exfiltrated this information, which could include the developer credentials and keys.[66] Several companies were thus compromised: Twilio, which inter alia provides two-factor authentication services, whose customers' email addresses were exfiltrated;[67] HashiCorp, which develops open-source software tools, whose GPG key – used for signing and verifying software releases – was exposed;[68] and other companies that reported unauthorized access to their source code repositories.[69] According to US federal government investigators, "hundreds of networks"[70] of Codecov's customers were impacted.

63 Ian Sommerville (2011): Software Engineering, 9th edition. Pearson Education.
64 Youssef Bassil (2012): A Simulation Model for the Waterfall Software Development Life Cycle, in: nternational Journal of Engineering & Technology, 2 (5).
65 Codecov (2021): Bash Uploader Security Update; Ionut Ilascu (16.04.2021): Popular Codecov code coverage tool hacked to steal dev credentials, BleepingComputer.
66 Codecov (2021): Bash Uploader Security Update; Ionut Ilascu (16.04.2021): Popular Codecov code coverage tool hacked to steal dev credentials, BleepingComputer.
67 Ax Sharma (04.05.2021): Twilio discloses impact from Codecov supply-chain attack, BleepingComputer.
68 Ax Sharma (24.04.2021): Twilio discloses impact from Codecov supply-chain attack, BleepingComputer.
69 Confluent (2021): Confluent Update Regarding Codecov Incident; Rapid7 (2021): Rapid7's Response to Codecov Incident.
70 Joseph Menn and Raphael Satter (20.04.2021): Codecov hackers breached hundreds of restricted customer sites - sources, Reuters.

5. **Deployment and Integration:** The software is made available to users and their programs[71] The software may reach the entity through different infrastructure, including corporate software publishing servers, package infrastructure, and app stores.[72] An example of a compromise during this phase is the Comm100 incident (see info box below).

The Comm100 incident
The Canadian company Comm100's Live Chat application provides online customer chat functionalities. In September 2022, researchers discovered that the installer that was available for download via the company's website was compromised: when executed, this modified version of the installer downloaded an additional piece of malware to gain persistence and exfiltrate data.[73] The compromised software was signed with a valid Comm100 certificate.[74] According to Comm100, 2% of its customers who had installed the application were affected.[75] The IT security company CrowdStrike identified affected organizations "in the industrial, healthcare, technology, manufacturing, insurance and telecommunications sectors in North America and Europe."[76] CrowdStrike attributed the operation, with "moderate confidence,"[77] to an actor with "a China nexus."[78] The Chinese government denied responsibility.[79]

6. **Maintenance and Disposal:** The objective of this final phase in the life cycle is to correct undetected errors, improve the implementation of the software, and add functionalities to it to meet new or emerging requirements.[80] This process is particularly challenging for firmware,[81] because update cycles are longer, among other factors.[82] A common compromise

71 IBM (2021): Deploying software.
72 John Speed Meyers (n.d.): What is software supply chain security. A beginner's guide to software supply chain security, Chainguard.
73 Crowdstrike (2022): CrowdStrike Falcon® Platform Identifies Supply Chain Attack via a Trojanized Comm100 Chat Installer.
74 Crowdstrike (2022): CrowdStrike Falcon® Platform Identifies Supply Chain Attack via a Trojanized Comm100 Chat Installer.
75 Comm100 (2022): Security Incident on September 29, 2022.
76 Crowdstrike (2022): CrowdStrike Falcon® Platform Identifies Supply Chain Attack via a Trojanized Comm100 Chat Installer.
77 Crowdstrike (2022): CrowdStrike Falcon® Platform Identifies Supply Chain Attack via a Trojanized Comm100 Chat Installer.
78 Crowdstrike (2022): CrowdStrike Falcon® Platform Identifies Supply Chain Attack via a Trojanized Comm100 Chat Installer.
79 Raphael Satter and Christopher Bing (01.10.2022): Suspected Chinese hackers tampered with widely used customer chat program, researchers say, Reuters.
80 Ian Sommerville (2011): Software Engineering, 9th edition. Pearson Education.
81 Firmware is "[c]omputer software that provides low-level control for the hardware and device(s) of a host, such as BIOS or UEFI/EFI". See The MITRE Corporation (2023): Firmware.
82 Binarly (2021): The Firmware Supply-Chain Security is broken: Can we fix it?

vector is the update process, which can be hijacked. An example is the operation that targeted the update process of Kaseya's Virtual System Administrator (VSA) software (see info box below). Other examples that have received much press coverage and attention from policy makers are the NotPetya[83] and SolarWinds/SUNBURST[84] incidents. Once a software has reached its end of life, it is usually disposed of (although there are cases in which software-developing entities still supply updates for products that have already reached their end of life).

The Kaseya incident

In July of 2021, Irish software service provider Kaseya recommended that its customers "IMMEDIATELY shut down"[85] their on-premise servers for Kaseya's VSA software, which its customers use to remotely monitor and manage networks and endpoints.[86] Malicious actors had exploited a vulnerability in Kaseya's systems[87] that allowed them to send an automatic update to the company's customers, which contained ransomware.[88] Many of Kaseya's customers are managed service providers (MSPs), who use VSA to remotely monitor and manage their customers' IT systems. The ransomware affected not only Kaseya's clients but also, in the case of clients that were MSPs, those organizations' customers. The ransomware reached supermarkets in Sweden, schools in New Zealand, municipalities in the US, and hospitals in Romania, among others.[89] The US Department of Justice later charged an individual linked to the REvil ransomware group with orchestrating the operation.[90]

This overview of SSC compromises throughout the SDLC does not suffice to explain why SSC compromises arise in the first place. To understand this and identify levers for government action, the root causes of these compromises must be examined.

83 Andy Greenberg (22.08.2018): The Untold Story of NotPetya, the Most Devastating Cyberattack in History, Wired.
84 Mandiant (2020): Highly Evasive Attacker Leverages SolarWinds Supply Chain to Compromise Multiple Global Victims With SUNBURST Backdoor.
85 Kaseya (2021): Important Notice August 4th, 2021.
86 Charlie Osborne (23.07.2021): Updated Kaseya ransomware attack FAQ: What we know now, ZDNET.
87 The MITRE Corporation (2021): CVE-2021-30116.
88 Mark Loman, Sean Gallagher, and Anand Ajjan (2021): Independence Day: REvil uses supply chain exploit to attack hundreds of businesses, Sophos.
89 Kellen Browning (02.07.2021): Hundreds of Businesses, From Sweden to U.S., Affected by Cyberattack, The New York Times; Ellen Nakashima and Rachel Lerman (21.09.2021): FBI held back ransomware decryption key from businesses to run operation targeting hackers, The Washington Post.
90 US Department of Justice (2021): Ukrainian Arrested and Charged with Ransomware Attack on Kaseya.

2.3 Root Causes of Software Supply Chain Compromises

SSC compromises have several root causes that lie in the software development ecosystem. These root causes are the bases of the toolbox we developed, which we will present in the next chapter.

First, software developers rely significantly on software **components developed, delivered, and maintained by others** and not by themselves.[91] While this practice is very efficient, it has three central implications for SSC security:

a. Developers make the security – understood as the confidentiality, integrity, and availability of data – of their software and of the entire SSC dependent on, inter alia, the security of these incorporated software components, also referred to as "**dependencies**."[92] These dependencies may be directly referenced in the final software in question, which are referred to as direct dependencies, or may be dependencies of dependencies, also called transitive dependencies.[93] The final software inherits the security risks of all its dependencies. Software dependencies may have, or may be modified to include, **vulnerabilities**.[94] Due to a vulnerability, software can "act in ways that designers and developers did not intend it to, or even expect."[95] Software vulnerabilities can be introduced intentionally – by malicious actors to compromise an entity or entities of an SSC or by developers with malicious intent – or unintentionally through developer errors or bad coding practices. In addition, each software component may be targeted to impact the final software. Popular targets include open-source libraries[96] and cloud services.[97] Both intentionally and unintentionally introduced vulnerabilities can, if exploited, lead to risks further down the SSC.[98] While there are tools that can facilitate assessment of the security risks inherent in software components,[99] complete security is not feasible.

91 Julius Musseau, John Speed Meyers, George P. Sieniawski, C. Albert Thompson, and Daniel German (2022): Is open source eating the world's software?: measuring the proportion of open source in proprietary software using Java binaries, MSR '22: Proceedings of the 19th International Conference on Mining Software Repositories, pp. 561–565.
92 Maya Kaczorowski (2020): Secure at every step: What is software supply chain security and why does it matter?, Github; Sonatype (2022): 8th Annual State of the Software Supply Chain.
93 Sonatype (2023): What are Transitive Dependencies?
94 According to the US NIST, a software vulnerability is "[a] security flaw, glitch, or weakness found in software code that could be exploited by an attacker (threat source)". See Kelley Dempsey, Paul Eavy, George Moore, and Eduardo Takamura (2020): Automation Support for Security Control Assessments: Software Vulnerability Management, National Institute of Standards and Technology (NIST).
95 Software Engineering Institute (2023): Security Vulnerabilities.
96 Trey Herr, William Loomis, Stewart Scott and June Lee (2020): Breaking trust: Shades of crisis across an insecure software supply chain, Atlantic Council.
97 Rani Osnat (2021): Supply Chain Attacks and Cloud Native: What You Need to Know, The New Stack.
98 John Speed Meyers (n.d.): What is software supply chain security. A beginner's guide to software supply chain security, Chainguard.
99 Sherif Koussa (n.d.): 13 tools for checking the security risk of open-source dependencies, TechBeacon.

b. Since some software components and their functions are generic, they are widely distributed and integrated across a **wide range of software applications**.[100] Vulnerabilities in such widely spread components can have large-scale security implications across the ecosystem.

Second, software-developing entities often d**o not prioritize security when developing their products**.[101] This has several reasons:

a. Faced with a trade-off between investing in software quality and security (to the extent that it is considered at all) on the one hand and achieving low development costs and short time-to-market on the other hand, **companies may have little incentive to invest in the security of their software.**[102] Companies that prioritize security may even have a market disadvantage in terms of, for example, higher cost.[103] These considerations can explain why software programs often have vulnerabilities and why such vulnerabilities may remain undetected for a long time or, even if they are discovered, may not be remediated immediately.[104] While there are companies that explicitly invest in and market their products based on security considerations,[105] this is not the case for the broader software ecosystem.

b. When software-developing entities incorporate software components into their products, they often lack incentives to make security-related demands on their suppliers. Accordingly, there are **often no contractual obligations** related to security (if there are any contractual agreements among entities of an SSC at all).[106]

c. In many jurisdictions, there is a **lack of regulation** that would force companies to prioritize security throughout the SSC and hold them liable if they fail to do so.[107] Related to this, even below the threshold of regulation, in many jurisdictions, there is little guidance available especially to small and

100 Liam Tung (03.03.2022): Log4Shell flaw: Still being used for crypto mining, botnet building... and Rickrolls, ZDNET.
101 Bruce Schneier (2018): Click Here to Kill Everybody. Security and Survival in a Hyper-connected World. W. W. Norton & Company; Matthew Green and Matthew Smith (2016): Developers are Not the Enemy!: The Need for Usable Security APIs, in: IEEE Security & Privacy, 14 (5), pp. 40-46.
102 Mark McFadden, Sam Wood, Robindhra Mangtani, and Grant Forsyth (2019): The economics of the security of consumer-grade IoT products and services, Internet Society; European Commission (2021): Study on the need of cybersecurity requirements for ICT products, Cloud Security Industry Summit Supply Chain Technical Working Group (2019): Secure Firmware Development Best Practices.
103 Bruce Schneier (2018): Click Here to Kill Everybody. Security and Survival in a Hyper-connected World. W. W. Norton & Company.
104 Nikolaos Alexopoulos, Manuel Brack, Jan Philipp Wagner, Tim Grube, and Max Mühlhäuser (2022): How Long Do Vulnerabilities Live in the Code? A Large-Scale Empirical Measurement Study on FOSS Vulnerability Lifetimes, in: Proceedings of the 31st USENIX Security Symposium; Yaman Roumani (2021): Patching zero-day vulnerabilities: an empirical analysis, in: Journal of Cybersecurity 7 (1).
105 John M. Blythe, Shane D. Johnson, and Matthew Manning (2020): What is security worth to consumers? Investigating willingness to pay for secure Internet of Things devices, in: Crime Science 9.
106 For an example for contractual cybersecurity obligations in the automotive sector, see ENX Association (2023): Trusted Information Security Assessment Exchange.
107 Paris Call for Trust and Security in Cyberspace, Cigref, Kaspersky, and GEODE (2021): Securing ICT supply chains.

medium enterprises (SMEs) on how they can design their processes to prioritize security throughout the SSC.[108]

d. Most **training programs for software developers** do not prioritize secure software development practices.[109] Furthermore, available security training programs tend to focus more on detecting and responding to vulnerabilities than on other supply chain concerns, such as on developing secure software builds. This is true for large parts of tertiary education as well as for private certifications,[110] which play an important role in the IT sector. While several initiatives are underway to change this,[111] it still holds true for many educational resources.

Third, software-using entities often find it **difficult to assess the security** of a given software. These circumstances explain the previous point: if users cannot base software sourcing decisions on security, there is little market pressure for software-developing entities and a lack of incentive for other stakeholders to prioritize security. There are several underlying reasons for users' struggle with assessing the security of software:

a. In the software market, software-using entities often lack meaningful information about the quality of products – in this case, the security of software – and therefore, low-quality products (in terms of security) may dominate the market.[112] This illustrates the presence of **information asymmetries** between consumers and suppliers in such market.[113] Also, there are often no readily available channels for software-developing entities to reliably provide buyers information about the security of their products, and there are few legal requirements to do so.[114]
b. For software users, be they individuals or organizations such as software-developing companies or governments, it is often **unclear which entities** (organizations or individuals) **are part of a given SSC**. This can be due to a lack of transparency in the components of a given software: if users do not know the components of their software, they cannot assess, for example, the security practices of the entities involved. This can be especially challenging for transitive dependencies.

108 ENISA (2021): Cybersecurity for SMEs - Challenges and Recommendations.
109 ENISA (2020): Cybersecurity Skills Development in the EU; Audun Jøsang, Marte Ødegaard, and Erlend Oftedal (2015): Cybersecurity Through Secure Software Development, in: Matt Bishop, Natalia Miloslavskaya, and Marianthi Theocharidou (eds.): Information Security Education Across the Curriculum. Springer, pp. 53-63; Cyber Safety Review Board (2022): Review of the December 2021 Log4j Event.
110 Alena Naiakshina (2020): Don't Blame Developers! Examining a Password-Storage Study Conducted with Students, Freelancers, and Company Developers.
111 Murugiah Souppaya, Karen Scarfone, and Donna Dodson (2022): Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities, NIST; The Linux Foundation (2023): Developing Secure Software (LFD121); (ISC)² (2023): CSSLP – The Industry's Premier Secure Software Development Certification.
112 DebateSecurity (2020): Cybersecurity Technology Efficacy: Is cybersecurity the new „market for lemons"?
113 ENISA (2020): Advancing Software Security in the EU.
114 European Commission (2021): Study on the need of cybersecurity requirements for ICT products; Vaibhav Garg and Andreas Kuehn (2021): Squeezing the Cybersecurity Lemons – A Labeling Regime for IoT Products, USENIX.

These root causes demonstrate the complexity of SSC security problems. In the next section, we analyze what policy makers can do about such problems.

3 A Toolbox for Policy Makers for Increasing Software Supply Chain Security

Software supply chain (SSC) security is a difficult problem for the three reasons outlined in the previous section: SSCs are complex, SSC compromises may affect different stages of the software development life cycle, and the characteristics of the software ecosystem facilitate compromises of SSCs. This is why increasing SSC security will require a combination of different interventions to make a difference. We opted for a toolbox approach for policy makers to illustrate the role of governments in this space. Such an approach also allows us to emphasize that a combination of different interventions is necessary to tackle this problem comprehensively. Moreover, a toolbox approach has the advantage of being flexible, so policy makers can choose from the different elements, considering their respective national circumstances such as political culture, policy priorities, institutional setup, resources, and capabilities. At the same time, by comparing the distinct elements of the toolbox, we were able to identify priority sets of recommendations (as detailed in the Conclusion chapter).

The toolbox (see Figure 3) consists of instruments and government actions. The five included instruments are quality assurance instruments (technical standards, conformity assessments, and product security labeling schemes), secure software development practices, coordinated vulnerability disclosure (CVD), software bill of materials (SBOM), and product liability. As mentioned in the Introduction, other instruments could be added to this list. We chose these five instruments because of the crucial role they already play or could play in the future in policy responses to the SSC security problem, according to expert assessments.[115]

These instruments are not specific to government action and can also be implemented by other stakeholders, particularly software-developing entities. In fact, in many cases, the objective of government intervention is to foster the use of these instruments by software-developing entities (although product liability is a slight exception to this logic because it can be mainly implemented by government actors). Still, governments can take diverse actions to increase SSC security. This is where the second element of the toolbox comes in: government action. For each instrument, we analyze the role

115 The recommendations are based on assessments of the experts who are part of the cyber diplomacy working group *Government's Role in Increasing Software Supply Chain Security* of Stiftung Neue Verantwortung. They contributed through background conversations, online collaboration, and a joint virtual workshop.

that governments can play in the immediate or mediated implementation of the instrument both at the national and international levels. To this end, the government actions we analyze are: convening stakeholders, issuing guidance, providing funding, education and workforce development, adapting governmental processes, issuing public procurement guidelines, and developing policies and regulation.

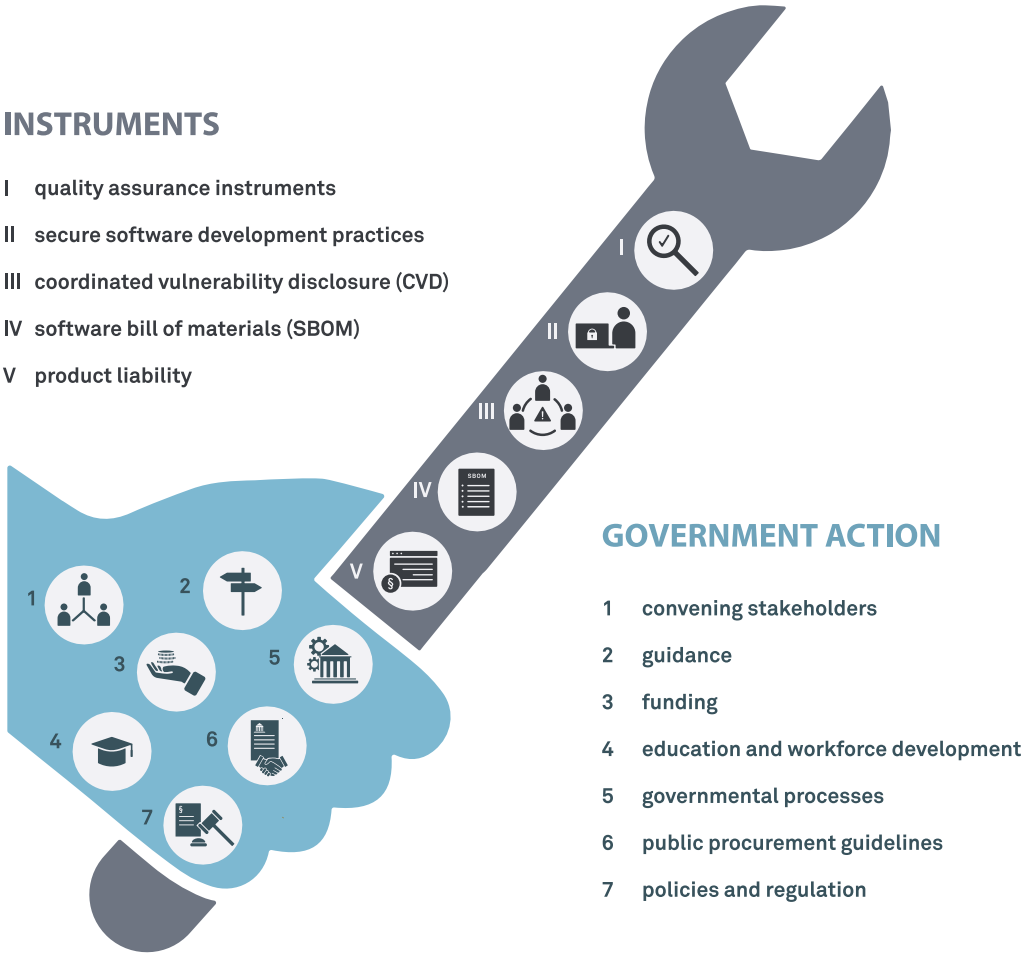


Figure 3:
A toolbox for policy makers for increasing software supply chain security

It should be noted that the instruments are interrelated. The three quality assurance instruments are general and can be applied to different issue areas, so they can play a role in the implementation of the four other instruments. Secure software development practices can be in diverse issue areas, and two of such practices are CVD and SBOM. Still, we consider the latter two individual instruments because they can be addressed through more specific government actions than can general secure software development practices. Finally, product liability is an instrument for enforcing the implementation of the other four instruments.

For each instrument, we first describe the instrument and explain how it can contribute to increasing SSC security. Second, we analyze through which government actions policy makers can implement the instrument. Third, we formulate recommendations on which government action each instrument should be paired with for maximum impact on SSC security. A caveat to these recommendations is the general lack of empirical impact assessments for the respective combinations of instruments and government actions. Therefore, we further recommend the conduct of such impact assessments of our recommendations, which are based on expert assessments.

3.1 Quality Assurance Instruments: Technical Standards, Conformity Assessments, and Product Security Labeling Schemes

From an SSC security perspective, quality assurance encompasses technical standards, conformity assessments, and product security labeling schemes. While these are distinct instruments, they should be considered together because they are interrelated and work toward the common goal of assuring quality and thereby increasing transparency for software users.[116] This is also why many of the government actions listed below approach the instruments in conjunction. Table 3 presents examples of all three quality assurance instruments that are relevant in an SSC security context.

Table 3: Examples of quality assurance instruments relevant for software supply chain security

Quality assurance instrument	Example
Technical standard	Common Criteria for Information Technology Security Evaluation standards[117]
Conformity assessment scheme	*IT-Grundschutz*,[118] a process-based conformity assessment scheme for organizations' information security management systems developed by the German Federal Office for Information Security (BSI) based on the ISO/IEC 27000-series[119]
Product security labeling scheme	Label of the CE marking scheme[120]

116 International Organization for Standardization (ISO) and United Nations Industrial Development Organization (UNIDO) (2010): Building Trust. The Conformity Assessment Toolbox.
117 Common Criteria (2023): Certified Products, ISO (2022): ISO/IEC 15408-1:2022 Information security, cybersecurity and privacy protection — Evaluation criteria for IT security — Part 1: Introduction and general model.
118 BSI (2023): IT-Grundschutz. A systematic basis for information security.
119 isms.online (2023): ISO IECD 27000.
120 YourEurope (2022): CE marking.

Quality Assurance Instruments: Description of the Instruments and Relevance for Software Supply Chain Security

A **technical standard** is "[a] document, established by consensus and approved by a recognized body, that provides, for common and repeated use, rules, guidelines, or characteristics for activities or their results, aimed at the achievement of the optimum degree of order in a given context."[121] Standards can make products interoperable, identify safety issues in products, and share good practices and solutions.[122]

There is an entire ecosystem of organizations that develop standards. First, there are standards development organizations (SDOs), which can be international[123] (e.g., the International Organization for Standardization, ISO), regional (e.g., the European Telecommunications Standards Institute, ETSI), or national (i.e., government bodies such as the US NIST or independent organizations such as the German Institute for Standardization, DIN). Second, there are community-driven SDOs (e.g., the Internet Engineering Task Force, IETF). With the exception of government agencies, SDOs are usually multi-stakeholder forums, which means that representatives of academia, civil society, government, and industry can join them – although in reality, governments and industry dominate these spaces, partly due to the significant resources necessary for participating. Third, standards can emerge from ad hoc collaboration, for example, if certain industry sectors are interested in quickly arriving at a common standard. These are also referred to as *de facto standards*. An example is CycloneDX, an SBOM data format standard.[124]

Technical standards provide a framework for security requirements for software, but they cannot guarantee that a given product, software-developing entity or software developer is complying with them. A **conformity assessment**, which can be defined as a "demonstration that specified requirements are fulfilled,"[125] can serve as a means of verifying compliance with established technical standards and providing assurance to software users. The requirements that provide the baseline for the assessment can be defined in standards, protection profiles (which apply to a whole product category, independent of the concrete implementation in a given product[126]), regula-

121 ISO (2004): ISO/IEC Guide 2:2004 Standardization and related activities -- General vocabulary.
122 ISO (2019): ISO in brief.
123 At the same time, national standards development organizations (SDOs) may play a double role because they can also be members or international and regional SDOs.
124 CycloneDX (2023): CycloneDX.
125 ISO (2020): ISO/IEC 17000:2020 Conformity assessment — Vocabulary and general principles.
126 BSI (2023): Protection profiles according to Common Criteria (CC) for IT products; National Information Assurance Partnership (2023): Approved Protection Profiles.

tion, or technical specifications. A well-known example of a conformity assessment scheme is the CE marking, through which product manufacturers attest to their products' conformity with European health, safety, and environmental protection standards.[127]

For conformity assessments of software products and software-developing entities, the most relevant activities are inspection, in which the object is examined to assess conformity with the requirements in question, and certification, in which a third party issues a statement attesting to the demonstrated fulfillment of the requirements in question.[128]

Conformity assessments often result in extensive and excessively technical documents that may not be easily understood by (prospective) users of software who are seeking information on the security properties of such software. **Product security labeling schemes** aim to bridge this information gap.[129] In an SSC security context, a product security label is a visual indicator, attached to or embedded in a software, which "indicates to consumers that [a product] has been demonstrated to meet specified requirements."[130] Labels can be based on conformity assessments or other processes that determine the degree of fulfillment of given criteria. In addition to government-led product security labeling schemes, there are also private labeling schemes, such as the Open Source Security Foundation (OpenSSF) Best Practices Badge Program,[131] through which open-source projects show that they follow secure software development best practices.

Product security labeling schemes for software can be binary or layered.[132] In a binary scheme, a product either receives a label – indicating that the software in question has met the criteria that provide the basis for the label, similar to a seal of approval – or not. An example of this is the label of the CE marking scheme. In contrast, layered schemes provide additional information – often through a qualitative distinction such as a scoring or traffic light system (e.g., labels on the energy consumption of products or the nutritional value of food).

127 YourEurope (2022): CE marking. This conformity assessment scheme also regards software in the case of medical devices (see Medical Device Coordination Group (2019): MDCG 2019-11 Guidance on Qualification and Classification of Software in Regulation (EU) 2017/745 – MDR and Regulation (EU) 2017/746 – IVDR) and wireless smart devices (see Commission Delegated Regulation (EU) 2022/30 of 29 October 2021 supplementing Directive 2014/53/EU of the European Parliament and of the Council with regard to the application of the essential requirements referred to in Article 3(3), points (d), (e) and (f), of that Directive [2022] OJ L7/6).
128 ISO (2020): ISO/IEC 17000:2020 Conformity assessment — Vocabulary and general principles.
129 Andreas Kuehn (2022): Security by Labeling, in: Communications of the ACM, 65 (9), pp. 23-25.
130 NIST (2022): Recommended Criteria for Cybersecurity Labeling of Consumer Software.
131 OpenSSF (2023): OpenSSF Best Practices Badge Program. However, the OpenSSF best practices badge is rare even among popular open-source projects.
132 NIST (2022): Recommended Criteria for Cybersecurity Labeling of Consumer Software.

All three quality assurance instruments address the underlying problems of consumers lacking access to meaningful information about the security of software and of software-developing entities lacking appropriate channels for providing security-related information to software users. Therefore, quality assurance instruments may increase SSC security in the long term by making the security properties of software more transparent and thus, allowing software-using entities to consider security features in their sourcing decisions.[133]

From an SSC security standpoint, all three instruments can be applied both to software products and to the processes of software-developing entities. Product-based quality assurance instruments evaluate the security characteristics of a particular product at a given point in time but do not allow inferences on the future security status of that product. In contrast, when the instruments are process-based, the security of a given product is derived from the general software development, maintenance, operation, and information security processes of the software-developing entity in question. This approach may allow for making limited inferences into the future but not regarding the specific security characteristics of the product in question.[134]

Furthermore, technical standards, conformity assessments, and product security labeling schemes can be voluntary or mandatory. Yet, even if they are only voluntary, they may become de facto requirements in certain sectors through private contractual requirements. In an SSC security context, the requirements of a conformity assessment or the security criteria of a labeling scheme can be technical standards and/or can refer to other tools discussed in this paper, such as secure software development practices and CVD policies.[135]

Conformity assessments and product security labeling can be conducted in different constellations. Among these, third-party conformity assessments – completed by a conformity assessment body, an independent entity without interest in the object, which is designated by a relevant government body[136] – may provide software users with a certain level of trust. However, third-party assessments are also the costliest and are not easy to scale. A potentially more scalable option is first-party assessments, which are

133 Andreas Kuehn (2022): Security by Labeling, in: Communications of the ACM, 65 (9), pp. 23-25.
134 ENISA (2020): Advancing Software Security in the EU.
135 NIST (2022): Recommended Criteria for Cybersecurity Labeling of Consumer Software.
136 ISO (2020): ISO/IEC 17000:2020 Conformity assessment — Vocabulary and general principles.

self-declarations by the software-developing entity.[137] Yet, to be effective, these need to be combined with market surveillance policies, through which national authorities "ensure that products comply with the requirements set out in the applicable [respective] legislation."[138] Market surveillance authorities target specific products, take samples, request documentation from manufacturers or providers, assess the sample products' compliance with relevant requirements, and take follow-up measures if products are found to violate relevant requirements.[139] Through such measures, market surveillance authorities can mandate product withdrawals or recalls by the manufacturer or provider, or can sanction them.[140] To have a meaningful impact, market surveillance requires investing significant resources.

The objective of all three instruments is to reduce the information asymmetries that are often prevalent in software markets. By providing software-using entities information on the security maturity of software, they could make more informed decisions about which software to buy and/or use. In the long term, the use quality assurance instruments may make other software users more aware of SSC security and may make security an additional factor for buying and/or using software. Moreover, the widespread use of all three instruments may contribute to mainstreaming transparency among software-developing entities regarding the security properties of their products, which might have benefits for SSC security in the long run.

At the same time, all three instruments face a chicken-and-egg problem:[141] as long as they are used only by a few software-developing entities, consumer awareness of the quality assurance instrument may remain low. This, in turn, may impede the quality assurance instrument from becoming the basis of consumers' software choices, which would incentivize software-developing entities to use the instruments more.

Another challenge for quality assurance instruments is the question of how to maintaining confidence in the security of a product over time. This is due to, first, changes in the update and maintenance of the software; second, possible changes in the product's (transitive) dependencies; and third,

137 The third option is second-party conformity assessments, conducted by an entity that has a user interest in the object of the assessment, such as software users or purchasers. See ISO (2020): ISO/IEC 17000:2020 Conformity assessment — Vocabulary and general principles.

138 Regulation (EU) 2019/1020 of the European Parliament and of the Council of 20 June 2019 on market surveillance and compliance of products and amending Directive 2004/42/EC and Regulations (EC) No 765/2008 and (EU) No 305/2011 [2019] OJ L169/1.

139 Zentralstelle der Länder für Sicherheitstechnik (2017): Good Practice for Market Surveillance.

140 European Commission (2023): Market surveillance for products.

141 Jurgita Lapienyte (28.09.2021): 'Cybersecurity Made In Europe' label goes live, cybernews.

changes in the operational environment of the product.[142] For these reasons, process-based schemes should include information on how long updates will be provided for a product. Moreover, some product-based schemes require reassessment after changes in any of the aforementioned domains.[143]

Finally, all three instruments are fairly resource intensive. At the same time, little empirical information is available regarding their long-term effects on SSC security. In the absence of such evaluations, governments risk creating paper tigers.

Quality Assurance Instruments: Government Action

Figure 4: Overview of government action for quality assurance instruments

Governments can foster the uptake of all three quality assurance instruments by convening stakeholders, issuing guidance, providing funding, taking action on education and workforce development, issuing public procurement guidelines, and developing policies and regulation.

142 ENISA (2020): Advancing Software Security in the EU.
143 ENISA (2020): Advancing Software Security in the EU.

Governments seeking to foster the use of all three quality assurance instruments can start by **convening stakeholders**. According to ENISA, there is a lack of coordination and effective exchange among stakeholders of many government-backed and private quality assurance schemes.[144] To address this shortcoming, governments could build communities of practice in their respective jurisdictions. Such communities could facilitate coordination among national stakeholders, share information and build capabilities on standards development processes for organizations less familiar with them, and contribute to trust-building within industries and across various sectors, which can promote the further implementation of the quality assurance instruments. Such efforts can build on existing private-sector coordination through trade groups, chambers of commerce, and industry verticals.

Moreover, in such forums, stakeholders could not only share information about current standards development efforts but also coordinate the participation of organizations in corresponding SDOs. This could help stakeholders – including governments – make more informed choices about which standards development efforts and forums to focus their resources on, which would allow for division of labor among different stakeholders and sustainable, long-term dedication to their selected forums. Moreover, stakeholders could share de facto standards and/or coordinate efforts to gain SDO endorsements. Finally, such a forum could also allow non-governmental entities to share insights with policy makers, for example, on what topics could be relevant in the future, which are essential information for all actors in the time-consuming development of technical standards and quality assurance schemes. In doing so, government agencies should be mindful of the multistakeholder character of the current SDO ecosystem and should thus actively foster its multistakeholder character, for example, through outreach activities to non-governmental entities. The consultations that would be held should be substantive to ensure that policy is grounded in the realities of software development and integration processes.

The same approach could also be taken at the international level. This is even more urgent because international coordination can help reconcile existing incompatible quality assurance schemes and help prevent future ones. Doing so should be a priority for governments because incompatible schemes can create costs for organizations in at least two ways: first, for studying the international and domestic landscapes in each jurisdiction, and second, for either customizing products for each jurisdiction or, when failing to do so, risking adverse effects thereof. Examples of these adverse effects

144 ENISA (2020): Advancing Software Security in the EU.

are sanctions by market surveillance authorities, failure to access the market, and inability to supply government entities based on their procurement guidelines. An example of international coordination on conformity assessments among EU member states is the European Cybersecurity Certification Group[145] and its multistakeholder equivalent, the Stakeholder Cybersecurity Certification Group.[146]

Against this backdrop, different governments – especially in like-minded constellations[147] – could convene their respective national SDOs and stakeholders that are active in quality assurance for the described coordination and division of labor purposes described above and to try and prevent or resolve incompatible schemes. Such coordination is already happening, for instance, in the context of the Organisation for Economic Co-operation and Development (OECD)[148] and the EU.[149] The more states take part in such coordination efforts, the more significant the impact will be. In the long term, the involved stakeholders could establish a "common repository"[150] for security requirements that feed into technical standards and schemes for conformity assessment and labeling. Such a repository could facilitate the exchange and application of best practices and the practical implementation of existing schemes while allowing for the identification of gaps in the existing landscape. These insights could also be used for cyber capacity-building activities to support states in establishing national quality assurance schemes and effective market surveillance, if included. Such cyber capacity-building activities should also involve non-state actors that are active in the quality assurance space.

Regarding the development of technical standards, governments can play a role in SDOs. They can also convene stakeholders to share their insights into the developments at these organizations, such as through the aforementioned multistakeholder forum. Governments can also actively encourage the participation of non-governmental actors, particularly, civil society and user representatives, who are often underrepresented in SDOs. For instance, governments could provide financial support to contributing civil organiza-

145 European Commission (2023): The European Cybersecurity Certification Group.
146 European Commission (2023): Stakeholder Cybersecurity Certification Group.
147 SDOs are becoming increasingly politicized. See Sorina Teleanu (2021): The geopolitics of digital standards: China's role in standard-setting organisations, DiploFoundation/Geneva Internet Platform and Multilateral Dialogue Konrad Adenauer Foundation Geneva; Giulia Neaher, David A. Bray, Julian Mueller-Kaler, and Benjamin Schatz (2021): Standardizing the Future. How Can the United States Navigate the Geopolitics of International Technology Standards?, Atlantic Council. This is why like-minded formats are good starting points rather than broader constellations.
148 OECD Regulatory Policy Division (2020): International Regulatory Co-operation Adapting rulemaking for an interconnected world.
149 European Commission (2023): Key players in European Standardisation.
150 ENISA (2020): Advancing Software Security in the EU.

tions, foster education programs on the importance of their involvement in standardization, and create a platform for connecting civil society organizations that are active in standards development. In bodies that are not open to non-governmental actors, or in cases when non-governmental actors face participation barriers, governments could also consult with these entities prior to sessions and thus, bring their viewpoints to the table.

On the global stage, governments could regularly convene large parts of the ecosystem of SDOs – including informal ones – from different parts of the world that are working on technical standards relevant for SSC security to coordinate on their priority areas and allocate resources accordingly. This can build on existing coordination efforts, such as the World Standards Cooperation.[151] Especially useful would be an international dialogue among SDOs, governments, and other stakeholders on existing and forthcoming regulatory projects for coordination and to receive inputs.

Furthermore, governments can issue **guidance** on quality assurance instruments to software-developing entities. In this way, governments can foster the implementation of already established standards and conformity assessment and labeling schemes. The role of government here would be to help organizations navigate the existing landscape. This is particularly necessary in jurisdictions with voluntary or mandatory schemes in place. An example of such guidance is New Zealand's *Information Security Manual*, which references relevant technical standards for software security.[152]

Such guidance should include recommendations on which technical standards, conformity assessments, and labeling schemes are suitable for which kinds of organization, considering different organization types, sectors, and sizes, among other factors. Such guidance should pay particular attention to SMEs, because they are likely to have limited resources. Considering the current diverse levels of maturity in SSC security across organizations, regulatory schemes could either start small or focus on baseline criteria for improving the practices of organizations for whom SSC security is currently not a strong priority. Alternatively, governments could opt for a layered approach to conformity assessments with varying assurance levels. This could also incentivize organizations that are already prioritizing SSC security to further improve since they would be able to signal their aspirations to their customers. In addition, guidance could include practical advice and templates for implementing the quality assurance instruments in question.

151 World Standards Cooperation (2023): What We Do.
152 New Zealand Government Communications Security Bureau (2022): ISM Document.

Moreover, governments could provide **funding** for research on the effects and success factors of technical standards, conformity assessments, and labeling schemes. Particularly insightful would be studies that compare schemes established in different countries and regions. In addition, the above-mentioned challenge of anticipating future standards development needs could be addressed by funding research, possibly identifying issue areas that could be included in future software development standards, exploring potential linkages to existing standards to propose amendments in regular standards updating processes, or producing concrete proposals for additional standards that could subsequently be introduced to SDOs.

In contrast to conformity assessments and labeling schemes, which governments usually develop through national policies and regulation, they can develop technical standards as part of regulation but also independently, especially through SDOs. They can also fund and dedicate skilled personnel to the development of new technical standards. For instance, they can task or invite their national SDOs to develop certain standards, if necessary, including commissioning research on relevant issues. They could use the suggested stakeholder forums to identify gaps in the standards landscape and to decide which issue areas these standards should focus on. Governments can do this directly depending on their respective domestic setups (i.e., if a national SDO is a government entity) or indirectly, by directing a non-governmental national SDO to perform the task or suggesting the same to it. Areas that are ripe for the development of new standards include product standards and process standards for the software build process. Government funding could also be used for research and development activities that feed into the standards development process.

Throughout such activities, governments should ensure active outreach to the multistakeholder standards development community to include their perspectives, even if a given national SDO is a government agency and thus, per se, does not include other stakeholders. This can be done, for example, through consultations similar to that which ENISA held for a software certification scheme[153] or through paid temporary placements of experts in the relevant government agencies. Moreover, governments could use funding to directly incentivize stakeholders to participate in SDOs, for instance, by providing tax breaks for personnel hours dedicated to SDOs or through a sponsorship program.

153 ENISA (2021): Public Consultation on the draft Candidate EUCC Scheme.

Governments could also use **education and workforce development** to promote adherence to quality assurance instruments. Such efforts will depend strongly on the structure of the national education system. In any case, governments could work toward the inclusion of information on technical standards, conformity assessments, and product security labeling schemes in the curricula of software development degree courses and certification programs. In addition, governments could develop and/or sponsor courses on the three instruments for professionals already working in software-developing entities to make them champions for the cause. Covering the cost of the course and reimbursing employers for the working time that their employees missed to attend the course could increase the uptake of the offer. Such courses could build on or use existing resources.[154]

In addition, governments could use **public procurement guidelines** to incentivize software-developing entities seeking to sell their products to the public sector to use quality assurance instruments. This can be another lever, in addition to directly providing funding, for changing the financial calculus of software-developing entities regarding the cost of implementing these instruments. Such guidelines could mandate the implementation of technical standards or adherence to conformity assessments or labeling schemes. An example is the *Cybersecurity Maturity Model Certification 2.0 program*[155] established by the US Department of Defense. It is a process-based conformity assessment scheme regarding requirements for handling of security information by companies that are part of the defense industrial base. Another example is the 2021 US *Executive Order 14028*[156] in combination with the 2022 White House Office of Management and Budget *Memorandum M-22-18*,[157] the NIST *Secure Software Development Framework (SSDF) Version 1.1*,[158] and the NIST *Software Supply Chain Security Guidance Under Executive Order 14028*.[159] Together, these documents require federal agencies to ensure that their software suppliers implement secure software development practices and can solicit first-party attestation to that effect. This includes the use of tools such as CVD and SBOM, which are described in more detail in following sections.

154 The Linux Foundation (2023): Secure Software Development: Requirements, Design, and Reuse, edX.
155 US Department of Defense Chief Information Officer (2023): About CMMC.
156 The White House (2021): Executive Order 14208, "Improving the Nation's Cybersecurity".
157 Executive Office of the President, Office of Management and Budget (2022): M-22-18. Memorandum for the Heads of Executive Departments and Agencies.
158 Murugiah Souppaya, Karen Scarfone, and Donna Dodson (2022): Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities, NIST.
159 NIST (2022): Software Supply Chain Security Guidance Under Executive Order (EO) 14028 Section 4e.

Such public procurement guidelines can also be harmonized internationally. An example of this is the Quad[160] grouping that includes Australia, India, Japan, and the US, which committed to aligning their public procurement guidelines to incorporate common baseline software security standards.[161]

Finally, governments can also develop **policies and regulation** to incentivize or require software-developing entities to implement the quality assurance instruments. This can be done through a common regulatory scheme that involves all three instruments, or through individual legislation for each instrument. A regulation covering all three instruments may specify, inter alia, the following aspects:[162]

1. The scope of the regulation, that is, the software products and/or processes in question;
2. Which entities are responsible for undergoing the process (e.g., vendors, retailers, or operators);
3. The organization that implements and enforces the regulation;
4. Requirements that must be satisfied, for instance, in the format of a technical standard;
5. Modalities for assessing conformity with the requirements;
6. Modalities for market surveillance (if applicable);
7. Sanctions in case of non-compliance; and
8. Labeling requirements, specifically, the design of and information on the label and the process of obtaining and displaying it.

A key challenge for regulation on quality assurance, as on other issues regarding the software ecosystem, is balancing how the regulation will affect not only large software-developing companies but also SMEs and individual developers. The latter two will typically have less resources to implement the regulation and may thus be disproportionately affected by regulatory burdens to the extent that their ability to continue operating may be jeopardized. Accordingly, governments, when establishing regulations, should be mindful of the different capabilities and resources of software-developing entities.

Similar to quality assurance guidance, regulation could focus on baseline requirements or follow a layered approach to encompassing different risk pro-

160 The Quad is short for 'Quadrilateral Security Dialogue,' a political group bringing together Australia, India, Japan, and the US on economic, diplomatic, and military issues.
161 The White House (2022): Quad Joint Leaders' Statement.
162 ISO and UNIDO (2010): Building Trust. The Conformity Assessment Toolbox; BSI (2023): IT-Sicherheitszertifizierung; NIST (2022): Recommended Criteria for Cybersecurity Labeling of Consumer Software.

files. Product-based conformity assessment or product labeling schemes also need to clarify under which conditions they lose their validity (e.g., after a given time or when a product modification, such as a software update, constitutes a substantial modification).[163]

Instead of one all-encompassing regulation, governments can tackle the issues one at a time. For example, they may set up voluntary process-based conformity assessment schemes such as the *IT-Grundschutz*[164] developed by the German Federal Office for Information Security (BSI), which is based on the *ISO/IEC 27000-series*[165] and applies to organizations' information security management systems. Similarly, the *CyberSecure Canada* conformity assessment scheme[166] is meant for SMEs and assesses their conformity with a process standard that lays out baseline cybersecurity requirements. When designing conformity assessment schemes, policy makers should carefully consider the requirements for conformity assessment bodies – such as technical knowledge – to ensure high-quality assessments.

Moreover, governments can establish labeling schemes. For example, Singapore established a voluntary labeling scheme for smart devices that includes their software.[167] In this layered scheme, the two lower levels are based on first-party assessment, and the two upper levels require third-party testing. Similarly, Germany has developed a binary IT security label for ICT products that comprises so far, inter alia, smart devices and that will be broadened to more product groups.[168] The US government has issued *Executive Order 14028* directing the development of a binary labeling scheme for consumer software.[169] Moreover, governments could establish mandatory labeling schemes. The EU Cyber Resilience Act draft legislation envisions a future product security labeling scheme that has yet to be specified.[170] Efforts are also underway to develop a common European conformity assessment scheme that also covers software products.[171]

In addition to such voluntary efforts, governments could also pass legislation that would make the observance of technical standards or compliance

163 Commonwealth of Australia (2021): Strengthening Australia's cyber security regulations and incentives.
164 BSI (2023): IT-Grundschutz. A systematic basis for information security.
165 isms.online (2023): ISO IECD 27000.
166 Government of Canada (2022): CyberSecure Canada. Frequently asked questions.
167 Cyber Security Agency of Singapore (2023): Cybersecurity Labelling Scheme (CLS).
168 BSI (2023): IT Security Label.
169 NIST (2022): Recommended Criteria for Cybersecurity Labeling of Consumer Software.
170 European Commission (2022): Proposal for a Regulation of the European Parliament and of the Council on horizontal cybersecurity requirements for products with digital elements and amending Regulation (EU) 2019/1020.
171 European Commission (2023): The EU cybersecurity certification framework. ENISA has already presented a candidate for a certification scheme. See ENISA (2021): Cybersecurity Certification: Candidate EUCC Scheme V1.1.1.

with conformity assessment and product security labeling schemes mandatory. Considering the low uptake of these schemes to date, such regulations should be passed with generous implementation timelines. Rather than requiring adherence from all software-developing entities, regulation could start with individual sectors (particularly those that are already subject to a regulator, such as the finance or telecommunications sector in many jurisdictions) or with software-developing entities that supply critical infrastructure providers. Subsequently, the requirements could be broadened to cover more software-developing entities and products. Such legislation needs to be combined with the creation of structures and processes for – in the case of first-party attestation – market surveillance, or, in the case of third-party attestation, an accreditation scheme for attestation bodies and sanction mechanisms. The EU Cyber Resilience Act draft legislation envisions mandatory conformity assessments for products with digital elements based on technical standards.[172] Depending on the criticality[173] of the product in question, these assessments are to be carried out by the first party or by a third party, with national bodies performing market surveillance functions.

At the international level, governments could work toward harmonizing quality assurance schemes. This would not only prevent incompatible schemes, as explained above, but also lower the compliance costs of software-developing entities. After all, even if the schemes in different jurisdictions are compatible, vendors still need to undergo and pay multiple times for conformity assessments and the assessments that are the bases of labeling schemes.

A step in this direction would be the mutual recognition of quality assurance schemes. This can occur on a bilateral basis; for example, the French and German national cybersecurity agencies recognize each other's cybersecurity conformity assessments.[174] Regarding the mutual recognition of product security labeling schemes, examples are Singapore's mutual recognition arrangements with Finland[175] and Germany.[176] Alternatively, states can organize mutual recognition of quality assurance instruments in a group, as is the case with the *Common Criteria for Information Technology*

172 European Commission (2022): Proposal for a Regulation of the European Parliament and of the Council on horizontal cybersecurity requirements for products with digital elements and amending Regulation (EU) 2019/1020.
173 This is applicable to critical products of Class II, as defined in Annex III of the draft legislation. See European Commission (2022): Proposal for a Regulation of the European Parliament and of the Council on horizontal cybersecurity requirements for products with digital elements and amending Regulation (EU) 2019/1020.
174 Agence Nationale de la Sécurité des Systèmes d'Information (n.d.): Signature d'un accord de reconnaissance mutuelle des certificats de sécurité entre l'ANSSI et le BSI.
175 Eileen Yu (06.10.2021): Singapore inks pact with Finland to mutually recognise IoT security labels, ZDNET.
176 Cyber Security Agency of Singapore (2022): Singapore and Germany Sign Mutual Recognition Arrangement on Cybersecurity Labels for Consumer Smart Products.

Security Evaluation (usually referred to only as Common Criteria). Common Criteria comprises product-based standards for a large spectrum of ICT products, including software (e.g., operating systems or firewalls)[177] and was also published as an ISO standard.[178] Moreover, the scheme includes the *Common Criteria Recognition Arrangement*, in which 16 states[179] recognize each other's conformity assessments aligned with the Common Criteria standard, and 15 further states[180] recognize the assessments issued by the authorities of the former 16 states. However, many software products are still beyond the scope of Common Criteria. Therefore, there is still room for governments to work toward a common quality assurance environment in which technical standards, conformity assessment schemes, and software product labeling schemes are harmonized among several states.

Quality Assurance Instruments: Priorities

Given the lack of assessments of the impact of quality assurance instruments on SSC security, governments should dedicate funding to research on the efficacy of all three quality assurance instruments in achieving SSC security. If such evaluations confirm the added value of these three tools to SSC security, governments could take three actions to advance the implementation of all three tools.

Regarding the development of technical standards, governments could first convene national and international stakeholders to prevent entities (including governments) from losing track of ongoing standards development efforts, missing engagement opportunities in relevant forums, or duplicating efforts by creating similar but non-harmonized standards in different forums. Second, governments could directly fund the development of relevant international standards by skilled personnel in national SDOs for subsequent endorsement by international SDOs.

As for conformity assessments and product security labeling schemes, governments with the necessary resources could establish such schemes at home and, ideally, coordinate these efforts internationally to learn from best practices and to prevent a fragmented quality assurance landscape that will increase the cost of compliance of software-developing entities.

177 Common Criteria (2023): Certified Products.
178 ISO (2022): ISO/IEC 15408-1:2022 Information security, cybersecurity and privacy protection — Evaluation criteria for IT security — Part 1: Introduction and general model.
179 Australia, Canada, France, Germany, India, Italy, Japan, Malaysia, the Netherlands, Norway, South Korea, Singapore, Spain, Sweden, Turkey, and the US. See Common Criteria (2023): Members of the CCRA.
180 Austria, Czech Republic, Denmark, Ethiopia, Finland, Greece, Hungary, Indonesia, Israel, New Zealand, Pakistan, Poland, Qatar, Slovakia, and the United Kingdom. See Common Criteria (2023): Members of the CCRA.

3.2 Secure Software Development Practices

Secure Software Development Practices: Description of the Instrument and Relevance for Software Supply Chain Security

Secure software development practices are guidelines for software-developing entities on how to build secure software. Accordingly, they include practices for software development by the software development entities in question but also for strengthening their relationships with other entities upstream and downstream in the supply chain. Examples of these practices are the establishment of requirements for suppliers or acquisitions and responses to vulnerability discovery.[181] In short, the objective is to include security in each phase of the SDLC. Secure software development practices address the root cause of SSC compromises: lack of training of software developers on security. This is essential, as many other instruments for fostering SSC security require trained personnel to implement them. Secure software development practices can also be subsumed under what is often referred to as practices that are secure by design and by default.

According to the US NIST's *Secure Software Development Framework*[182] secure software development practices can entail practices that seek to increase the security of the software-developing entity as a whole, software components, software releases, and the process of responding to vulnerabilities. According to guidance by CISA, from a software developer perspective, secure software development practices can mean, inter alia, developing secure code, verifying third-party components, and hardening the build environment.[183] The Cloud Native Computing Foundation distinguishes between the parts of software development that should be secured: the source code, materials, build pipelines, artifacts, and deployments.[184] A key component of secure software development practices is the secure use of open-source components. Secure software development practices commonly include CVD and SBOM,[185] which we will discuss in more detail in Sections 3.3 and 3.4.

181 Murugiah Souppaya, Karen Scarfone, and Donna Dodson (2022): Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities, NIST.
182 Murugiah Souppaya, Karen Scarfone, and Donna Dodson (2022): Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities, NIST.
183 CISA (2022): Securing the Software Supply Chain. Recommended Practices Guide for Developers.
184 CNCF (n.d.): Software Supply Chain Best Practices.
185 Murugiah Souppaya, Karen Scarfone, and Donna Dodson (2022): Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities, NIST; CISA (2022): Securing the Software Supply Chain. Recommended Practices Guide for Developers.

There is a broad body of knowledge on secure software development practices that government actions can build on. First, several technical standards contain guidance on the issue, such as ISO/IEC 27034-1:2011,[186] which contains guidance on application security, and IEC 62443-4-1:2018,[187] which focuses on industrial automation and control systems.

Second, several non-governmental entities have issued secure software development practices guidance. The *Supply-chain Levels for Software Artifacts* (SLSA) framework is a set of standards for, and good practices in, secure software development for software-developing entities. These are organized into four levels, which correspond to security guarantees.[188] The framework is the result of a collaboration among several software-developing entities, including Google, and is being maintained within the OpenSSF. The SLSA framework can be used for different purposes, including for internal use to improve an organization's SSC security posture and as the basis for third-party attestation as part of a conformity assessment to share information about secure software development practices with software-using entities.[189] The *Open Web Application Security Project (OWASP) Top 10*[190] is a regularly updated ranking of the top 10 security risks for web applications and their respective remediation guidelines, thus contributing to mitigating SSC security risks. Such ranking is elaborated by voluntary contributors under the auspices of OWASP. Moreover, the Software Assurance Forum for Excellence in Code (SAFECode), a non-profit organization that brings together private-sector representatives on software security issues, published *Fundamental Practices for Secure Software Development*.[191] The best-practice guide contains recommendations on software design, secure coding practices, risk management for third-party components, testing and validation, management of security findings, vulnerability response and disclosure, and advice on the practical implementation and deployment of secure software development practices.

Third, there are individual practical software tools that can contribute to securing software development practices. An example is *Sigstore,*[192] a set of

186 ISO (2011): ISO/IEC 27034-1:2011 Information technology — Security techniques — Application security — Part 1: Overview and concepts.
187 International Electrotechnical Commission (2018): IEC 62443-4-1:2018. Security for industrial automation and control systems - Part 4-1: Secure product development lifecycle requirements.
188 Supply chain Levels for Software Artifacts (2023): Requirements.
189 Supply chain Levels for Software Artifacts (2023): Use Cases.
190 Open Web Application Security Project (2021): OWASP Top Ten.
191 SAFECode (2018): Fundamental Practices for Secure Software Development Essential Elements of a Secure Development Lifecycle Program.
192 Sigstore is affiliated with the Linux Foundation and is currently led by Google, Red Hat, and Purdue University. See The Linux Foundation (2023): sigstore.

open-source software tools catering to open-source projects that seek to simplify the process of signing software and verifying such signatures. This process, referred to as *code signing*, "provides both data integrity to prove that the code was not modified, and source authentication to identify who was in control of the code at the time it was signed."[193] Another example is open-source tools aimed at ensuring reproducible builds.[194] A software build is "an operational version of a system or component that incorporates a specified subset of the capabilities that the final product will provide."[195] A reproducible build means that "given the same source code, build environment, and build instructions, any party can recreate bit-by-bit identical copies of all specified artifacts."[196]

Secure Software Development Practices: Government Action

Figure 5: Overview of government action for secure software development practices

193 David Cooper, Andrew Regenscheid, Murugiah Souppaya, Christopher Bean, Mike Boyle, Dorothy Cooley, and Michael Jenkins (2018): Security Considerations for Code Signing, NIST.
194 Reproducible Builds (2023): Tools.
195 Institute of Electrical and Electronics Engineers (1990): 610.12-1990 - IEEE Standard Glossary of Software Engineering Terminology.
196 Reproducible Builds (2023): Definitions.

Governments that want to foster the observance of secure software development practices can issue guidance, provide funding, take action on education and workforce development, adapt governmental processes, issue public procurement guidelines, and develop policies and regulation.

Governments can build on a wide body of knowledge on secure software development practices to issue **guidance** on this topic. A starting point can be translating guidance prepared by stakeholders into their respective national languages. By issuing such guidance, governments can support organizations that are relative novices in secure software development practices (e.g., SMEs or companies that only recently broadened their product portfolio to include software, as is the case for some providers of Internet of Things (IoT) devices). Catering to these audiences, governments can point to existing quality assurance instruments, frameworks, and practical software tools. For example, governments could endorse or develop technical standards and, building on those, conformity assessment schemes – and potentially, also product security labeling schemes – that attest to software-developing entities observing a set of secure software development practices. Government guidance can also build on secure software development practices guidance issued by non-state actors. These can, in the long term, "improve the baseline"[197] of secure software development practices. Alternatively, similar to quality assurance instruments, such guidance can be layered to be relevant to organizations of different sizes and maturity levels regarding secure software development practices.

For example, the US NIST published the *SSDF*.[198] It consists of a set of high-level principles that organizations can implement in various ways, thereby allowing for flexibility. It includes implementation examples and references for each practice. CISA published a report titled *Securing the Software Supply Chain: Recommended Practices Guide for Developers*,[199] which takes an approach similar to that of the NIST guidance. The Cyber Safety Review Board, a multistakeholder expert body tasked with analyzing the underlying causes of major cybersecurity incidents, in its report on Log4j, also formulated recommendations for secure software development practices.[200] ENISA issued recommendations on secure software development practices for different stakeholders in a study.[201]

197 Trey Herr, William Loomis, Stewart Scott and June Lee (2020): Breaking trust: Shades of crisis across an insecure software supply chain, Atlantic Council.
198 Murugiah Souppaya, Karen Scarfone, and Donna Dodson (2022): Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities, NIST.
199 CISA (2022): Securing the Software Supply Chain. Recommended Practices Guide for Developers.
200 Cyber Safety Review Board (2022): Review of the December 2021 Log4j Event.
201 ENISA (2021): Threat Landscape for Supply Chain Attacks.

Also, international coordination can be fruitful in this context to either coordinate the development of quality assurance instruments, as explained above, or exchange best practices and harmonize guidance.

Governments can also use **funding** to promote secure software development practices. This can include research on which secure software development practices actually correlate with less known vulnerabilities,[202] and the development of international standards on this issue – whether in national or international setups.

Moreover, governments can take action on **education and workforce development**.[203] Governments can directly invest in education and training programs that address secure software development practices as part of software engineering and computer science programs. Alternatively, they can incentivize education institutions to include secure software development practices in their curricula and certifications. Other government actions related to secure software development practices may incentivize the private sector, which requires adequately trained personnel, to invest in and use relevant educational and professional training resources.

Regarding the adaptation of **governmental processes**, governments should implement secure software development practices when developing software in-house. To put this into practice, it might be necessary for governments to first map which of their agencies develop software, as this information may not be readily available.[204] An internal inventory of all government entities that develop software facilitates their implementation of secure software development practices in their processes. A government entity could be assigned the responsibility of collecting and updating the needed information and serving as a point of contact for questions and/or advice for other governmental actors regarding secure software development practices. Taking stock of the status quo of in-house software development can also open up opportunities for strategic decisions and can highlight potential blind spots and risks that could be detrimental to SSC security. As explained below, such a list can also facilitate the development of CVD policies and the use of SBOMs by public entities nationwide.

Furthermore, governments can issue **public procurement guidelines** that mandate the observance of secure software development practices. The

202 Sonatype (2022): 8th Annual State of the Software Supply Chain.
203 Cyber Safety Review Board (2022): Review of the December 2021 Log4j Event.
204 Markus Borg (2018): Digitalization of Swedish Government Agencies – A Perspective Through the Lens of a Software Development Census.

2021 US *Executive Order 14028*[205] in combination with the 2022 White House Office of Management and Budget *Memorandum M-22-18*,[206] SSDF Version 1.1,[207] and the NIST *Software Supply Chain Security Guidance Under Executive Order 14028*,[208] require federal agencies to ensure that their software suppliers implement certain quality assurance mechanisms and can solicit evidence to that effect. The SSDF was issued in this context and provides a basis for subsequent attestation, showing the nexus between guidance and harder procurement guidelines.

Finally, governments can also spread the observance of secure software development practices through **policies and regulation**. Policy makers can develop quality assurance and product security labeling schemes based on technical standards or other guidance focused on secure software development practices. Requiring such certification for software-developing entities providing software to the public sector or to critical infrastructure would give such regulation more teeth and would strongly contribute to spreading secure software development practices through the market.

Secure Software Development Practices: Priorities

Overall, secure software development practices are a cornerstone of efforts to increase SSC security. Governments should foster the observance of secure software development practices to impact software development practices and thereby spur the development of more secure software, which would decrease possible entry vectors for SSC compromises. If software-developing entities fail to implement secure software development practices, it can be due to at least the following three factors, which should be addressed through tailored government actions. These should be the priorities of governments aiming to increase SSC security.

The first possible reason for software developing entities failing to implement secure software development practices is their lack of skilled personnel. This is why governments should focus on education regarding secure software development practices. Education is also the foundation of all other government actions, as people trained in secure software development practices are needed to scale these practices among software-developing entities. This, in

205 The White House (2021): Executive Order 14208, "Improving the Nation's Cybersecurity".
206 Executive Office of the President, Office of Management and Budget (2022): M-22-18. Memorandum for the Heads of Executive Departments and Agencies.
207 Murugiah Souppaya, Karen Scarfone, and Donna Dodson (2022): Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities, NIST.
208 National Institute of Standards and Technology (2022): Software Supply Chain Security Guidance Under Executive Order (EO) 14028 Section 4e.

turn, is a requirement for the successful implementation of more specialized aspects of secure software development practices, such as CVD and SBOM.

The second possible reason is lack of knowledge about relevant secure software development practices that match the organization's profile and lack of guidance on the practical implementation of these practices. Therefore, governments should prioritize providing guidance for different types of organizations. This is particularly relevant for SMEs and organizations that only recently incorporated software into their product portfolio.

The third possible reason may be unwillingness to observe secure software development practices. In this case, only regulation can change the entity's calculus, especially when mandatory and combined with sanctions for non-compliance. For entities supplying to the public sector, public procurement guidelines can have a similar effect. These government actions are likely to have a major effect on the product offering in the market and can change the calculus of entities with dominant positions in the market.

3.3 Coordinated Vulnerability Disclosure

Cordinated Vulnerability Disclosure: Description of the Instrument and Relevance for Software Supply Chain Security

When software products are brought to the market, they may – and likely do – have undiscovered vulnerabilities.[209] Software-developing entities mitigate this in most cases[210] by issuing patches, which remediate the vulnerability when deployed in user systems. A coordinated way of handling the process, from vulnerability discovery to remediation, is referred to as *coordinated vulnerability disclosure* (CVD). The EU NIS2 Directive defines CVD as follows:

> *"[C]oordinated vulnerability disclosure specifies a structured process through which vulnerabilities are reported to organisations in a manner allowing the organisation to diagnose and remedy the vulnerability before detailed vulnerability information is disclosed to third parties or to the public. Coordinated vulnerability disclosure should also comprise coordination between the reporting entity and the organisation as regards the timing of remediation and publication of vulnerabilities."*[211]

209 Bruce Schneier (2018): Click Here to Kill Everybody. Security and Survival in a Hyper-Connected World. W. W. Norton & Company.
210 Andrey Solovev (2022): The Basics of The Firmware Development Process, Hackernoon.
211 Directive (EU) 2022/2555 of the European Parliament and of the Council of 14 December 2022 on measures for a high common level of cybersecurity across the Union, amending Regulation (EU) No 910/2014 and Directive (EU) 2018/1972, and repealing Directive (EU) 2016/1148 (NIS 2 Directive) [2023] OJ L333/80.

In the absence of CVD, finders of a software vulnerability may not share information about the vulnerability with entities that can provide remediation. Instead, they may either keep the information to themselves or give – or sell – it to third parties (non-disclosure) or, alternatively, they may share the information with the public before a remediation is available (full disclosure).[212] In both cases, information about the vulnerability reaches a third party, which can potentially exploit it, before a remediation is available.

In contrast to these two scenarios, the objective of CVD is to make it easier for finders of software vulnerabilities to share information about the vulnerability in a timely manner with entities that can provide remediation. From an SSC security perspective, it is desirable for software-developing entities to establish CVD to decrease the chances of SSC compromises through vulnerability exploitation.

At the same time, in complex SSCs, the same vulnerability may affect multiple products. Under these circumstances, the CVD process becomes more complex because more actors are involved:[213] in so-called vertical supply chains, "a vulnerability exists in multiple products because they all share a dependency on a vulnerable library or component."[214] In this case, one vendor develops an original remediation, which many other vendors need to incorporate into their respective products as individual remediations. In contrast, in the case of horizontal supply chains, "multiple products implement the same vulnerability,"[215] such as a design flaw. This situation requires original remediation from each individual vendor.

212 Andrew Cencini, Kevin Yu, and Tony Chan (2005): Software Vulnerabilities: Full-, Responsible-, and Non-Disclosure.
213 Forum of Incident Response and Security Teams (FIRST) (2020): Guidelines and Practices for Multi-Party Vulnerability Coordination and Disclosure.
214 Allen D. Householder, Garret Wassermann, Art Manion, and Chris King (2017): The CERT® Guide to Coordinated Vulnerability Disclosure, Carnegie Mellon University.
215 Allen D. Householder, Garret Wassermann, Art Manion, and Chris King (2017): The CERT® Guide to Coordinated Vulnerability Disclosure, Carnegie Mellon University.

According to the non-profit CERT Coordination Center, the CVD process consists of roughly the following six phases:[216]

1. Discovery: Somebody outside the affected codebase (e.g., an independent security researcher, employee of another company, or government official) discovers a vulnerability in a software;
2. Reporting: They report information about the vulnerability to the product vendor, open-source project maintainer, or a third party, such as a trusted coordinator;[217]
3. Validation and triage: The receiver of the information validates and prioritizes it;
4. Remediation: A remediation plan is developed – a software patch or other mechanisms, such as configuration changes;
5. Public awareness: Information about the vulnerability and the remediation plan is disclosed to the public; and
6. Deployment: The remediation is rolled out to the deployed systems.

In addition, stakeholders have formulated principles[218] and developed international standards[219] that can guide the CVD process.

216 Allen D. Householder, Garret Wassermann, Art Manion, and Chris King (2017): The CERT® Guide to Coordinated Vulnerability Disclosure, Carnegie Mellon University.

217 A coordinator "acts as a relay or information broker between other stakeholders". Allen D. Householder, Garret Wassermann, Art Manion, and Chris King (2017): The CERT® Guide to Coordinated Vulnerability Disclosure, Carnegie Mellon University. Diverse types of organizations can assume the role of coordinator, including computer security incident response teams (CSIRTs), security research organizations, information sharing and analysis organizations, and commercial brokers. See Allen D. Householder, Garret Wassermann, Art Manion, and Chris King (2017): The CERT® Guide to Coordinated Vulnerability Disclosure, Carnegie Mellon University.

218 Allen D. Householder, Garret Wassermann, Art Manion, and Chris King (2017): The CERT® Guide to Coordinated Vulnerability Disclosure, Carnegie Mellon University; FIRST (2020): Guidelines and Practices for Multi-Party Vulnerability Coordination and Disclosure.

219 Two ISO standards concern CVD: ISO/IEC 29147:2018 provides guidance for vendors on the disclosure process. See ISO (2018): ISO/IEC 29147:2018 Information technology — Security techniques — Vulnerability disclosure. ISO/IEC 30111:2019 contains good practices for processing vulnerability reports. See ISO (2019): ISO/IEC 30111:2019 Information technology — Security techniques — Vulnerability handling processes.

Coordinated Vulnerability Disclosure: Government Action

Figure 6: Overview of government action for coordinated vulnerability disclosure

If governments want to promote the use of CVD, they can convene stakeholders, issue guidance, provide funding, adapt governmental processes, issue public procurement guidelines, and develop policies and regulation.

First, governments can **convene stakeholders** – specifically, security researchers, vendors, and the open-source community – to discuss the current status of CVD and to explore good practices. This should also be done at an international level – possibly in a like-minded format – to allow discussion of cross-border vulnerability disclosure. Existing venues that could be used for more in-depth discussions on these topics are regional networks of computer security incident response teams (CSIRTs), the EU NIS Cooperation Group,[220] the OECD's Committee on Digital Economy Policy,[221] and

220 European Commission (2023): NIS Cooperation Group.
221 OECD (2023): Committee on Digital Economy Policy (CDEP).

multistakeholder organizations such as the Global Forum on Cyber Expertise[222] or the Cybersecurity Tech Accord.[223] These settings could also be used to address the challenge of processing vulnerability disclosures of critical open-source projects and the possible allocation of funds and personnel.[224]

In addition, governments can promote CVD adoption by providing **guidance** on how to establish organizational CVD policies for software-developing entities. Such policies clarify expectations about appropriate behavior from all stakeholders involved in the CVD process.[225] They can provide information on the scope of the policy and the legal considerations for vulnerability finders, as well as on practical issues such as quality requirements for vulnerability reports, contact information, and response timelines.[226] Guidance on how to develop and publish organizational CVD policies is especially relevant to small organizations, such as SMEs or companies that only recently included IoT devices in their portfolio. As an example, the US NIST published the compilation *Foundational Cybersecurity Activities for IoT Device Manufacturers,*[227] in which it raised inter alia the issue of CVD. Catering to SMEs, CISA published SSC security guidance that also addresses CVD.[228]

In preparing such guidance, governments can draw on broad available materials from non-governmental actors[229] or even international organizations,[230] including language protecting security researchers[231] or templates[232] for vulnerability report forms, vulnerability disclosure documents, and CVD policies. The US government provides such guidance through a template CVD

222 Global Forum on Cyber Expertise (2017): GFCE Global Good Practices. Coordinated Vulnerability Disclosure (CVD).
223 Cyber Tech Accord (2023): Vulnerability Disclosure Policies.
224 OECD (2022): Recommendation of the Council on the Treatment of Digital Security Vulnerabilities.
225 For an overview of organizations with CVD policies in place, see, for instance, Disclose.io (2023): #diodb search.
226 Allen D. Householder, Garret Wassermann, Art Manion, and Chris King (2017): The CERT® Guide to Coordinated Vulnerability Disclosure, Carnegie Mellon University.
227 Michael Fagan, Katerina N. Megas, Karen Scarfone, and Matthew Smith (2020): Foundational Cybersecurity Activities for IoT Device Manufacturers, NIST.
228 CISA (n.d.): Securing Small and Medium-Size Business Supply Chains. A resource handbook to reduce information and communication technology risks.
229 Allen D. Householder, Garret Wassermann, Art Manion, and Chris King (2017): The CERT® Guide to Coordinated Vulnerability Disclosure, Carnegie Mellon University; FIRST (2022): Traffic Light Protocol (TLP). First Standards Definitions and Usage Guidance — Version 2.0; FIRST (2020): Guidelines and Practices for Multi-Party Vulnerability Coordination and Disclosure; ETSI (2022): ETSI TR 103 838 V1.1.1. Cyber Security; Guide to Coordinated Vulnerability Disclosure.
230 OECD (2022): Recommendation of the Council on the Treatment of Digital Security Vulnerabilities; OECD (2021): Encouraging Vulnerability Treatment. Responsible Management, Handling and Disclosure of Vulnerabilities; Organization for Security and Co-operation in Europe (2023): OSCE Cyber/ICT Security CBM 16: Coordinated Vulnerability Disclosure.
231 This can build on private-sector initiatives such as HackerOne's Gold Standard Safe Harbor. HackerOne (2023): Gold Standard Safe Harbor.
232 Allen D. Householder, Garret Wassermann, Art Manion, and Chris King (2017): The CERT® Guide to Coordinated Vulnerability Disclosure, Carnegie Mellon University; OpenSSF (2022): oss-vulnerability-guide/templates/security_policies/, Github; securitytxt.org (2023): security.txt.

policy by the US National Telecommunications and Information Administration (NTIA).[233] The Dutch National Cyber Security Centre's *Coordinated Vulnerability Disclosure: The Guideline*[234] has become an international good practice guide for establishing organizational CVD policies. For cross-border vulnerabilities disclosure, it would be helpful if different governments – in collaboration with non-state actors – would agree on internationally harmonized baseline reporting templates.

In addition, governments can provide **funding** to improve CVD practices. First, related government structures should receive increased funding and personnel, especially for a national CVD coordinator (if such an entity exists).[235] Second, governments could contribute to the further development of international CVD standards through funds and personnel.

Furthermore, government agencies that develop software should lead by example and establish CVD in their own **governmental processes** by publishing organizational CVD policies. In doing so, software-developing government agencies could incentivize security researchers to search for vulnerabilities in their software. Government agencies should also allocate available capacity, funds, and resources for implementing these organizational CVD policies.[236] Instead of isolated efforts, governments can create a whole-of-government approach to CVD and mandate all government agencies to develop CVD policies, as the US government has done via CISA.[237] Alternatively, the government could designate one entity to handle CVD for all other government entities, as is the case with the Japan Computer Emergency Response Team (JPCERT), the national CSIRT of Japan.[238]

Moreover, governments can leverage **public procurement guidelines** to incentivize software vendors to develop public CVD policies.[239] For instance, under the 2021 US *Executive Order 14028*[240] in combination with the 2022 White House Office of Management and Budget *Memorandum M-22-18*,[241] US federal agencies can solicit evidence from software vendors that they

233 National Telecommunications and Information Administration (NTIA) (2016): Coordinated Vulnerability Disclosure "Early Stage" Template and Discussion.
234 National Cyber Security Centre (2018): Coordinated Vulnerability Disclosure: The Guideline.
235 Cyber Safety Review Board (2022): Review of the December 2021 Log4j Event.
236 OECD (2022): Recommendation of the Council on the Treatment of Digital Security Vulnerabilities.
237 CISA (2020): Binding Operational Directive 20-01 - Develop and Publish a Vulnerability Disclosure Policy.
238 ENISA (2022): Coordinated Vulnerability Disclosure Policies in the EU.
239 OECD (2022): Recommendation of the Council on the Treatment of Digital Security Vulnerabilities.
240 The White House (2021): Executive Order 14208, "Improving the Nation's Cybersecurity".
241 Executive Office of the President, Office of Management and Budget (2022): M-22-18. Memorandum for the Heads of Executive Departments and Agencies.

have organizational CVD policies in place, depending on the criticality[242] of the software in question. Governments could even go one step further by requiring all software vendors supplying to governmental entities to have CVD policies.

Furthermore, governments should use **policies and regulation** to spread the use of CVD in their jurisdiction. Specifically, governments should establish legal frameworks for CVD.[243] These can have at least three elements: mandatory CVD policies for vendors, clarification of the legal status of security researchers, and appointing a national CVD coordinator.

First, legal frameworks for CVD may require software vendors to have organizational CVD policies. To illustrate, the EU Cyber Resilience Act draft legislation would prescribe this for all vendors of products with digital elements.[244]

Second, the legal framework for CVD should clarify the legal status of security researchers through so-called "safe harbor" provisions.[245] Currently, in many jurisdictions, security researchers face a patchwork of regulations that may limit or even criminalize vulnerability disclosure.[246] The EU NIS2 Directive[247] "encourages"[248] such legal provisions, and the US Department of Justice recently announced that it will no longer charge good faith security researchers who act in good faith under the Computer Fraud and Abuse Act, although the latter has yet to be codified in law.[249] A move in the opposite direction is China's 2021 *Regulations on the Management of Network Product*

242 Which software products are to be considered critical is considered in the White House Office of Management and Budget Memorandum M-21-30. See Executive Office of the President, Office of Management and Budget (2021): M-21-30. Memorandum for the Heads of Executive Departments and Agencies.

243 These are often also referred to as national CVD policies. See ENISA (2022): Coordinated Vulnerability Disclosure Policies in the EU. We use the term *legal framework* to avoid confusion between organizational and national policies.

244 Products with digital elements are defined as "products [...] whose intended or reasonably foreseeable use includes a direct or indirect logical or physical data connection to a device or network". See European Commission (2022): Proposal for a Regulation of the European Parliament and of the Council on horizontal cybersecurity requirements for products with digital elements and amending Regulation (EU) 2019/1020.

245 OECD (2022): Recommendation of the Council on the Treatment of Digital Security Vulnerabilities.

246 Sunoo Park and Kendra Albert (2020): A Researcher's Guide to Some Legal Risks of Security Research, Harvard Law School Berkman Klein Center for Internet & Society Cyberlaw Clinic and Electronic Frontier Foundation.

247 The directive is not directly binding but needs to be transposed into domestic legislation by member states by October 17, 2024 at the latest. See Directive (EU) 2022/2555 of the European Parliament and of the Council of 14 December 2022 on measures for a high common level of cybersecurity across the Union, amending Regulation (EU) No 910/2014 and Directive (EU) 2018/1972, and repealing Directive (EU) 2016/1148 (NIS 2 Directive) [2023] OJ L333/80.

248 Directive (EU) 2022/2555 of the European Parliament and of the Council of 14 December 2022 on measures for a high common level of cybersecurity across the Union, amending Regulation (EU) No 910/2014 and Directive (EU) 2018/1972, and repealing Directive (EU) 2016/1148 (NIS 2 Directive) [2023] OJ L333/80.

249 US Department of Justice (2022): 9-48.000 - Computer Fraud and Abuse Act.

Security Vulnerabilities,[250] which limits vulnerability disclosure by China's active community of security researchers.

Third, policy makers may appoint a national CVD coordinator who can resolve issues between stakeholders in the CVD process – typically between security researchers and software-developing entities – and who can assist in the coordination of complex CVD processes in vertical or horizontal supply chains.[251] Different bodies can assume the role of coordinator, including national CSIRTs, national cybersecurity agencies, ministries, and NGOs. If a government entity will assume this role, a legal framework for security research must be in place. The coordinator must also be a trusted actor, and there should be no legal provisions in place that would hamper the coordinator's ability to fulfill its role. To illustrate this, Japan already established a national CVD process in 2004 and appointed the Information-Technology Promotion Agency as the contact point for vulnerability finders and the JPCERT Coordination Center (JPCERT/CC) as coordinator.[252] In the Netherlands, the National Cyber Security Centre performs this function.[253] The EU NIS2 Directive requires member states to nominate a CSIRT as coordinator.[254] At the international level, these coordinators can also assist stakeholders in cross-border vulnerability disclosures by contacting their international counterparts.

Taken together, these three elements can form a national legal framework for CVD. An example of a state with such a framework in place is Japan.[255] The recent EU NIS2 Directive[256] also mandates member states to establish one, which will mean a significant step-up since, as of April 2022, only four EU member states had such frameworks in place.[257] National legal frameworks

250 For an English summary, see ENISA (2022): Coordinated Vulnerability Disclosure Policies in the EU. However, the authors of an Atlantic Council report did not find that the regulation had a significant impact on the vulnerabilities disclosure practices of the Chinese security researcher community. Stewart Scott, Sara Ann Brackett, Yumi Gambrill, Emmeline Nettles, and Trey Herr (2022): Dragon tails: Preserving international cybersecurity research, Atlantic Council.

251 OECD (2022): Recommendation of the Council on the Treatment of Digital Security Vulnerabilities.

252 Japan Computer Emergency Reponse Team Coordination Center (JPCERT/CC) (n.d.): Information Security Early Warning Partnership. Overview of Vulnerability Handling Process.

253 National Cyber Security Centre (2023): Reporting a vulnerability (CVD).

254 Directive (EU) 2022/2555 of the European Parliament and of the Council of 14 December 2022 on measures for a high common level of cybersecurity across the Union, amending Regulation (EU) No 910/2014 and Directive (EU) 2018/1972, and repealing Directive (EU) 2016/1148 (NIS 2 Directive) [2023] OJ L333/80.

255 ENISA (2022): Coordinated Vulnerability Disclosure Policies in the EU.

256 Directive (EU) 2022/2555 of the European Parliament and of the Council of 14 December 2022 on measures for a high common level of cybersecurity across the Union, amending Regulation (EU) No 910/2014 and Directive (EU) 2018/1972, and repealing Directive (EU) 2016/1148 (NIS 2 Directive) [2023] OJ L333/80.

257 ENISA (2022): Coordinated Vulnerability Disclosure Policies in the EU. The four states were Belgium, France, Lithuania, and the Netherlands.

for CVD should be based on relevant international standards.[258] For example, Japan's CVD framework and the EU NIS2 Directive both endorse the respective ISO standards.[259]

Since CVD processes are likely to cross borders, policy makers should also strive for international awareness or, ideally, harmonization of CVD regulations. A first step could be to share information about states' respective national CVD policies. In a second step, governments – potentially starting with like-minded constellations – should internationally harmonize their respective laws to facilitate cross-border vulnerabilities disclosure, with the objective of creating "a common disclosure environment."[260] Such harmonization efforts should focus on protecting security researchers and requiring software-developing entities to publish and comply with their respective organizational CVD policies.[261] At the same time, policy makers should ensure that international treaties do not counter these efforts, for instance, by criminalizing security research. This should be considered, as states are negotiating an international cybercrime treaty at the UN.[262]

Cordinated Vulnerability Disclosure: Priorities

Considering the importance of vulnerabilities in SSC compromises, a structured CVD process can contribute to reducing future compromises. At the same time, CVD may incentivize security researchers to actively search for vulnerabilities in software, especially when a national legal framework for CVD is in place. At the same time, establishing organizational CVD policies requires limited resources from software-developing entities.

Policy makers have two levers at their disposal to encourage CVD in their jurisdictions, each with two government actions. First, they should incentivize software-developing entities to put organizational CVD policies in place. To reach organizations that lack the knowledge or resources to do so, governments should issue guidance on how to develop such policies. Pro-

258 ISO (2018): ISO/IEC 29147:2018 Information technology — Security techniques — Vulnerability disclosure; ISO (2019): ISO/IEC 30111:2019 Information technology — Security techniques — Vulnerability handling processes.
259 JPCERT/CC (n.d.): Information Security Early Warning Partnership. Overview of Vulnerability Handling Process; Directive (EU) 2022/2555 of the European Parliament and of the Council of 14 December 2022 on measures for a high common level of cybersecurity across the Union, amending Regulation (EU) No 910/2014 and Directive (EU) 2018/1972, and repealing Directive (EU) 2016/1148 (NIS 2 Directive) [2023] OJ L333/80.
260 Stewart Scott, Sara Ann Brackett, Yumi Gambrill, Emmeline Nettles, and Trey Herr (2022): Dragon tails: Preserving international cybersecurity research, Atlantic Council.
261 OECD (2022): Recommendation of the Council on the Treatment of Digital Security Vulnerabilities; Stewart Scott, Sara Ann Brackett, Yumi Gambrill, Emmeline Nettles, and Trey Herr (2022): Dragon tails: Preserving international cybersecurity research, Atlantic Council.
262 United Nations Office on Drugs and Crime (2023): Ad Hoc Committee to Elaborate a Comprehensive International Convention on Countering the Use of Information and Communications Technologies for Criminal Purposes.

viding ready-to-use templates for policies and language for safe-harbor statements significantly reduces the hurdles for organizations to put these in place. Such guidance should specifically cater to SMEs and software novices. In addition, governments should adapt public procurement guidelines to require that software-developing entities supplying to the public sector to have CVD policies in place. This does not have to be required for all software-developing entities but it will provide a strong incentive for other organizations to implement CVD, especially those organizations that have the means to establish CVD policies but did not have an incentive to do so. At the same time, such guidance would give proponents of CVD in organizations a strong argument to convince potential internal opponents.

Second, governments should lead by example and embrace CVD. They should do so first by developing and implementing organizational CVD policies for all software-developing government agencies. This would serve as a good practice and, not least, could incentivize security researchers to scrutinize government-built software more thoroughly, thereby increasing their security. In addition, governments should lay the groundwork for CVD in their respective jurisdictions by establishing a national legal framework for CVD. Such frameworks, by creating legal certainty for security researchers and appointing a national CVD coordinator, create an environment conducive to CVD participation by all involved stakeholders. Policy makers, depending on their level of ambition, can consider whether they want to go as far as requiring all software-developing entities to put organizational CVD policies in place.

3.4 Software Bill of Materials

Software Bill of Materials: Description of the Instrument and Relevance for Software Supply Chain Security

SSC compromises can have such significant implications because they can, for example, target one software component that is used in many other software products. This is why it is crucial to know the components and dependencies of a given software. However, it used to be uncommon to provide such information. This distinguishes software from other products – for instance, machinery, for which manufacturers routinely compile bills of materials. These lists detail each component of a given product and are a cornerstone of supply chain management. A software bill of materials (SBOM) transposes this idea to software: it is "a formal record containing the details and supply

chain relationships of various components used in building software."[263] It results in a nested inventory of the ingredients of software components at a given time.

SBOMs are produced by software vendors (at build time or, for legacy software, through source code or binary code analysis) –, by open-source developers and maintainers, or by companies dedicated to providing this service.[264] Since SBOMs paint a static picture, they need to be updated for each software release. Among the possible consumers of SBOM information are "end users, customers, auditors, regulators, policy makers, and suppliers."[265] SBOM consumers can use the information on the composition and provenance of a software listed in an SBOM for different purposes, including incident response; making procurement decisions; managing assets, vulnerabilities, and licenses; and complying with export control regulations.[266]

SBOMs are relevant to SSC security because they address a root cause of SSC compromises –lack of transparency of software and its dependencies and supplier structure – while giving users better tools for identifying and mitigating SSC compromises once they have been discovered. For example, if information about a vulnerability or exploitation of a software component surfaces, SBOM data would provide users good information on the components of their software and can allow users to track a vulnerability – based on suitable software vulnerability information – through the supply chain.

While SBOMs have passed the proof-of-concept stage, the following barriers to their wide adoption across software-developing entities remain.[267] First, there are high barriers of entry: both SBOM production and consumption require software tools and, more importantly, processes in place and resources for providing or integrating SBOM information, which may require a restructuring of business processes. Second, there is a so-called "SBOM chicken-and-egg problem":[268] as long as customers do not or only rarely demand SBOMs, vendors have less incentive to invest resources into producing them – although use cases for SBOM beyond security, such as for license management, already provide incentives to use them. At the same time, as

263 NTIA (2020): Software Bill of Materials (SBOM).
264 Ariadne Conill (2022): Not All SBOMs Are Created Equal, Chainguard.
265 ISO (2023): ISO/IEC FDIS 27036-3 Cybersecurity — Supplier relationships — Part 3: Guidelines for Hardware, Software, and Services Supply Chain Security, Annex C.
266 NTIA (2019): Survey of Existing SBOM Formats and Standards; Amelie Koran, Wendy Nather, Stewart Scott, and Sara Ann Brackett (2022): The Cases for Using the SBOMs We Build, Atlantic Council; Apertis (2021): Export Controls.
267 Stephen Hendrick (2022): Software Bill of Materials (SBOM) and Cybersecurity Readiness, The Linux Foundation, OpenSSF, Openchain, and SPDX.
268 Elias Groll and John Hewitt Jones (22.12.2022): Software Bills of Material Face Long Road to Adoption, Fedscoop; Tom Alrich (2020): An Opportunity to Be Part of the Solution.

long as SBOMs are seldom provided, consumers have little reason to invest resources into creating the structures needed for consuming SBOMs. Third, open questions remain as to the content and format of SBOM data,[269] including the depth of dependency tracking (i.e., how many dependencies down the software-developing entities are required to map) and the inclusion of SBOMs in software development and governance processes as well as risk management and compliance processes.[270] Fourth, open-source maintainers have particular challenges in SBOM use.[271]

As mentioned, a central factor that determines the usefulness – and therefore, also the likelihood of widespread adaptation – of SBOMs is the standardization of data formats.[272] Standardized data formats increase the usefulness of SBOMs, as they ensure that the data contained can be processed and used effectively. Moreover, the more internationally accepted these standards are, the more useful they will likely be. Currently, there are several data format standards for SBOMs, which converge in that they contain information on the supplier, component name, identifier, version, component hash, relationship, and SBOM author.[273]

Another key factor is the development of further technical tools that take advantage of SBOM data. For example, a software component listed in an SBOM may contain a vulnerability, as published in a vulnerability database such as the Malware Information Sharing Platform instance operated by the Forum of Incident Response and Security Teams (FIRST).[274] However, the information relevant to users is whether the vulnerability is exploitable in their deployed software. This is where the Vulnerabilities Exploitability eXchange (VEX) comes in. VEX is a standard for sharing data on the exploitability of vulnerabilities.[275] If the vulnerability is found to be exploitable in the soft-

269 NTIA (2021): Comments on Software Bill of Materials Elements and Considerations.
270 Tom Alrich (2022): Face It: SBOMs Will Never Be Regulated into Use; Stephen Hendrick (2022): Software Bill of Materials (SBOM) and Cybersecurity Readiness, The Linux Foundation, OpenSSF, Openchain, and SPDX. Both the EU Cyber Resilience Act draft legislation and the US NTIA guidance foresee SBOMs that contain only one level of dependencies. European Commission (2022): Proposal for a Regulation of the European Parliament and of the Council on Horizontal Cybersecurity Requirements for Products with Digital Elements and Amending Regulation (EU) 2019/1020; US Department of Commerce and NTIA (2021): The Minimum Elements for a Software Bill of Materials (SBOM). Pursuant to Executive Order 14028 on Improving the Nation's Cybersecurity. This is problematic because analysis of the vulnerable Log4j component revealed that in most cases, the dependency was more than one level down, and in some cases, as many as nine levels down (see James Wetter and Nicky Ringland (2021): Understanding the Impact of Apache Log4j Vulnerability, Open Source Insights).
271 Robin Gandhi, Matt Germonprez, and Georg J.P. Link (2018): Open Data Standards for Open Source Software Risk Management Routines: An Examination of SPDX, in: Proceedings of the 2018 ACM International Conference on Supporting Group Work.
272 Cyber Safety Review Board (2022): Review of the December 2021 Log4j Event.
273 NTIA (2019): Survey of Existing SBOM Formats and Standards; ISO (2023): ISO/IEC FDIS 27036-3 Cybersecurity — Supplier relationships — Part 3: Guidelines for Hardware, Software, and Services Supply Chain Security.
274 FIRST (2023): FIRST Malware Information Sharing Platform (MISP) Instance.
275 CISA (2022): Vulnerability Exploitability eXchange (VEX) – Use Cases.

ware in question, the software user would be interested to learn about ways of remediating the vulnerability. The Common Security Advisory Framework (CSAF) is a standard for automating the production, consumption, and processing of cybersecurity advisories.[276] Combined with SBOM and VEX data, CSAF data can speed up the remediation process.

Software Bill of Materials: Government Action

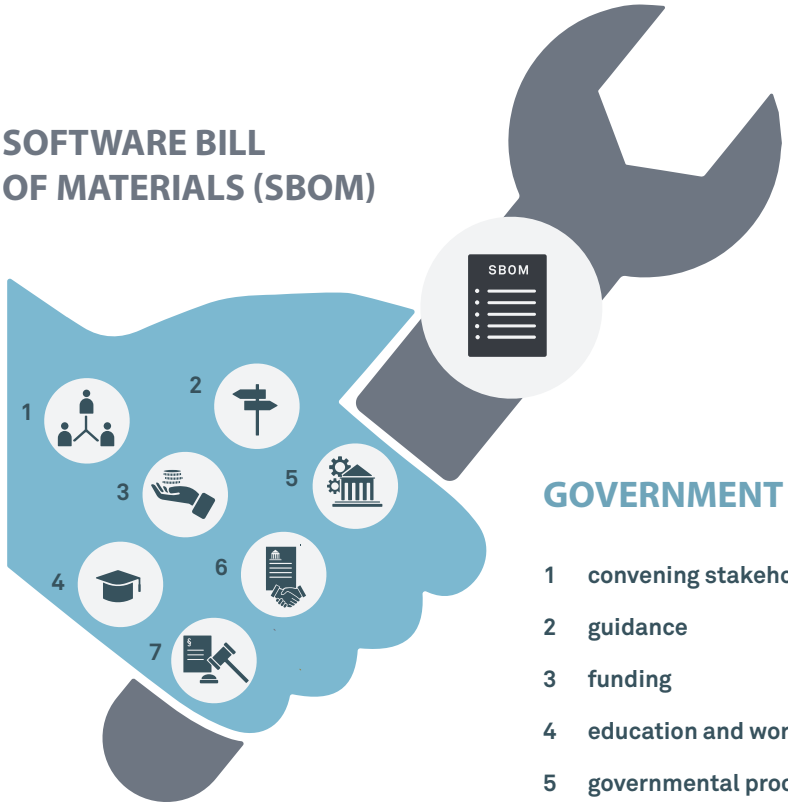


Figure 7: Overview of government action for software bill of materials

Governments have significant sway in promoting the adoption of SBOMs: by convening stakeholders, issuing guidance, providing funding, taking action on education and workforce development, adapting governmental processes, issuing public procurement guidelines, and developing policies and regulation.

276 Organization for the Advancement of Structured Information Standards (2023): Common Security Advisory Framework (CSAF).

First, governments can **convene stakeholders** to foster communities of practice around SBOM usage. For example, the government of Japan acted as patron for the Open Source Security Summit in 2022, which brought together the international open-source community to, inter alia, promote SBOM usage.[277] Governments can use existing forums to foster an SBOM ecosystem, such as by discussing the status quo and challenges of SBOM usage, or they can establish new communities focused on SBOM use.

Furthermore, governments can convene stakeholders to aggregate SBOM data in order to identify critical software components that are widely used in the ecosystem.[278] Such a list could show, for instance, which components need to be most urgently secured. Accordingly, it can provide guidance for other government and private-sector actions, such as by directing the attention of security researchers through CVD or better tailoring education and training materials to the current state of SBOM data. SBOM data aggregation can be done at the level of individual organizations (e.g., companies) or at a combined level (e.g., nationwide or even internationally, such as for the framework of international CSIRT networks). Governments can encourage the private sector to conduct the former, and they can spearhead and coordinate the latter (e.g., through their national CSIRTs or cybersecurity agencies). Such aggregated lists would need to be secured well, because such data would also reveal vulnerable targets to malicious actors.

Building on these efforts, governments can issue **guidance** to software-developing entities regarding SBOM. First, such guidance can recommend the use of SBOMs in general terms, as in the case of Australia's *Information Security Guidelines – Guidelines for Software Development*.[279] Second, governments can provide practical guidance to software-developing entities on how to produce and/or consume SBOM data. For instance, they can recommend the use of existing (de facto) standards or technical tools that build on SBOM data. An example of this is the designation by the US CISA of the adoption of VEX and CSAF as two of "three critical steps to advance the vulnerability management ecosystem."[280] Considering the high barriers of entry to SBOM production and consumption, these efforts should be combined with governmental guidance and potential funding for SMEs. Such guidance documents could outline good practices for SBOM production and consumption

277 OpenSSF (2022): The Linux Foundation and Open Source Software Security Foundation (OpenSSF) Gather Japanese Industry and Government Leaders for Open Source Software Security Summit Japan.
278 Trey Herr, Robert Morgus, Stewart Scott, and Tianjiu Zuo (2022): Buying Down Risk: Software Provenance and Composition, Atlantic Council.
279 Australian Signals Directorate and Australian Cyber Security Centre (2022): Information Security Manual. Guidelines for Software Development.
280 CISA (2022): Transforming the Vulnerability Management Landscape.

and point to practical information (e.g., data formats, open-source software for producing or using SBOMs, and even examples of how SBOM production and generation can be incorporated in existing operational processes). Such guidance for SMEs is especially necessary in jurisdictions where public procurement guidelines or regulation mandating SBOM use (as mentioned in the next section) are in place, to enable compliance with such provisions.

While efforts to standardize SBOM elements and formats are still ongoing, governments can support these processes with **funding** and personnel by participating in the standards development and refining process. This is true both for the development of standards for SBOM data formats and for technical tools that build on SBOM. As for SBOM data formats, in the standards development community, one of the three main SBOM data formats – SPDX – has been published as an international standard by ISO.[281] Moreover, ISO is currently preparing the publication of a standard that, in its current form, contains details on SBOM formats and elements.[282] As explained above, governments can support standards development processes by incentivizing the private sector or academic experts to participate and by coordinating with other states on the forum for and content of standards development.

As for technical tools that build on SBOM data, governments can similarly take part in the international standards development process. In the case of CSAF, for example, the German BSI joined other international experts in the task force responsible for developing the standards, under the auspices of the non-profit consortium Organization for the Advancement of Structured Information Standards (OASIS).[283] Thereby, the German government dedicated skilled personnel to the development of this tool. Finally, governments could dedicate funding and personnel to the development of further technical tools that provide actionable information on SBOM content for different use cases.[284] An example could be freely available tools for generating reliable SBOM data.[285]

As long as SBOM production and consumption are not yet widespread or mature, this area is ripe for governmental interventions in **education and work-**

281 ISO (2021): ISO/IEC 5962:2021 Information technology — SPDX® Specification V2.2.1.
282 ISO (2023): ISO/IEC FDIS 27036-3 Cybersecurity — Supplier relationships — Part 3: Guidelines for Hardware, Software, and Services Supply Chain Security. This standard will replace the 2013 ISO/IEC 27036-3:2013 standard (see ISO (2013): ISO/IEC 27036-3:2013 Information Technology — Security Techniques — Information Security for Supplier Relationships — Part 3: Guidelines for Information and Communication Technology Supply Chain Security).
283 BSI (2023): Common Security Advisory Framework (CSAF).
284 Amelie Koran, Wendy Nather, Stewart Scott, and Sara Ann Brackett (2022): The cases for using the SBOMs we build, Atlantic Council.
285 European Commission (2022): Identify (and Find Ways to Help Fix) Critical Open Source Software Used by European Public Services.

force development. Government agencies can promote knowledge about SBOMs among potential producers and consumers of SBOM data, such as by facilitating exchange platforms with industry and the technical community and by publishing readily accessible information. Governments can also include or incentivize actors to include SBOM-related content in software development, cybersecurity, and risk management conformity assessments; technical trainings; and curricula for academic degrees. This content should include guidance on the implementation of SBOM production and consumption processes as well as on education on the use cases of SBOMs to encourage the active consumption of SBOMs.[286] Governments could collaborate internationally when building a curriculum for technical training materials for software developers and risk managers on the issue. Moreover, they could include SBOM training in broader international cyber capacity-building activities that target the private sector. These education and workforce development activities also contribute to creating an SBOM community of practice.

Governments can also lead by example by adapting **governmental processes** to provide SBOMs for software developed by government agencies.

Finally, governments can use **public procurement guidelines** to leverage SBOM adoption. This follows the idea that making the provision of SBOMs a requirement for vendors wanting to sell their products to the government will lead to greater overall adoption in the market. Essentially, public procurement guidelines create demand for SBOMs, which provides an incentivize for their production not only by companies currently selling to the government or planning to do so in the near future but also by companies that may see this requirement as a first step to more widespread requirements in the market, whether through regulation or buyer demand. For example, according to the US *Executive Order 14028*[287] in combination with the White House Office of Management and Budget *Memorandum M-22-18*,[288] federal agencies can require their suppliers to provide SBOM documentation. The decision to require SBOMs shall be based on the criticality[289] of the software.[290]

286 The Linux Foundation and OpenSSF (n.d.): The Open Source Software Security Mobilization Plan; Amelie Koran, Wendy Nather, Stewart Scott, and Sara Ann Brackett (2022): The Cases for Using the SBOMs We Build, Atlantic Council.
287 The White House (2021): Executive Order 14208, "Improving the Nation's Cybersecurity".
288 Executive Office of the President, Office of Management and Budget (2022): M-22-18. Memorandum for the Heads of Executive Departments and Agencies.
289 Which software products are to be regarded as critical is considered in the White House Office of Management and Budget Memorandum M-21-30 (see Executive Office of the President, Office of Management and Budget (2021): M-21-30. Memorandum for the Heads of Executive Departments and Agencies).
290 Executive Office of the President, Office of Management and Budget (2022): M-22-18. Memorandum for the Heads of Executive Departments and Agencies.

Going further, governments could mandate vendor provision of SBOMs for all or specific government agencies or systems. Long timeframes would be key to allowing companies to adapt their processes, acquire tooling, and allocate resources to SBOM production. International coordination is essential here: as it is unlikely that public procurement guidelines especially of smaller countries would impact the SBOM practices of large software vendors, the more governments will join forces to develop similar recommendations, the likelier those recommendations are to have the desired market-shaping effect.

Moreover, governments can use **policies and regulation** to mandate suppliers to provide and users to consume SBOMs. The EU plans to introduce market access legislation to this end: the EU Cyber Resilience Act draft legislation envisions that "manufacturers should identify and document components contained in the products with digital elements, including by drawing up a software bill of materials."[291] Such regulation would need to be passed with long implementation timeframes, however, considering the current low prevalence of SBOM data. Beyond market access legislation for specific products, governments could mandate SBOM usage by critical infrastructure providers or by their suppliers. Among the key points that such regulation needs to address are:[292]

1. Who is responsible for producing SBOMs;
2. How regularly must the SBOMS be updated;
3. How compliance and conformity will be ensured, including who is responsible for enforcing the legislation and for determining the cost of non-compliance; and
4. What will be the special role and capacities of the open-source community.

Regulation can also serve to endorse and thereby strengthen data format standards. For instance, according to the US *Executive Order 14028*[293] in combination with the White House Office of Management and Budget *Memorandum M-22-18*,[294] software vendors from whom US federal agencies choose to solicit SBOM documentation need to provide this information in one of three data formats specified in a guidance document[295] by the US NTIA. The

291 European Commission (2022): Proposal for a Regulation of the European Parliament and of the Council on Horizontal Cybersecurity Requirements for Products with Digital Elements and Amending Regulation (EU) 2019/1020.
292 Amelie Koran, Wendy Nather, Stewart Scott, and Sara Ann Brackett (2022): The Cases for Using the SBOMs We Build, Atlantic Council.
293 The White House (2021): Executive Order 14208, "Improving the Nation's Cybersecurity".
294 Executive Office of the President, Office of Management and Budget (2022): M-22-18. Memorandum for the Heads of Executive Departments and Agencies.
295 U.S. Department of Commerce and NTIA (2021): The Minimum Elements for a Software Bill of Materials (SBOM). Pursuant to Executive Order 14028 on Improving the Nation's Cybersecurity. The three recommended data formats are CycloneDX, SPDX, and SWID.

EU Cyber Resilience Act draft legislation envisions a future implementing act by the European Commission specifying the SBOM format or elements.[296] While regulation is usually in the national domain, states can share their national approaches with each other, for instance, through platforms such as the OECD. Also, coordination – potentially starting with sets of like-minded states – can lend more weight to regulatory efforts. This is particularly relevant as vendors that are already required to provide SBOM data according to the regulation in one jurisdiction can more easily provide such data to consumers in other jurisdictions if the requirements converge.

Software Bill of Materials: Priorities

SBOM data can play an important role in identifying and, together with other data, mitigating the negative effects of SSC compromises. At the same time, a chicken-and-egg problem stands in the way of broadened SBOM use among software-developing entities who produce SBOMs, and software-using entities who consume SBOMs. Furthermore, SBOMs have become more useful to software-developing entities as more technical tools build on the data contained in them, so the development of such tools should be a priority.

To foster the use of SBOM data within the software ecosystem, as with CVD, governments have two levers at their disposal, each with two government actions. First, they can lower the hurdles for software-developing entities to get started with SBOM use. To this end, they can issue guidance on data formats, technical tools building on SBOM data, and practical implementation advice, for instance, regarding the integration of SBOM processes into existing business operations. In addition, they can advance the development of international technical standards for data formats for SBOM data and their respective technical tools. These efforts, which also require international coordination, can make SBOM use more attractive to organizations, as unified data formats will likely increase adaptation.

Second, policy makers can make SBOM use mandatory for certain software-developing entities. To start, policy makers could require SBOM use for those entities selling to the public sector through public procurement guidelines. Such public procurement guidelines could also be harmonized internationally for increased leverage vis-à-vis international software-developing entities. Moreover, governments could require certain software-de-

296 European Commission (2022): Proposal for a Regulation of the European Parliament and of the Council on Horizontal Cybersecurity Requirements for Products with Digital Elements and Amending Regulation (EU) 2019/1020.

veloping entities to use SBOMs, for instance, starting with those that supply critical infrastructure providers.

3.5 Product Liability

Product Liability: Description of the Instrument and Relevance for Software Supply Chain Security

The fifth instrument for increasing SSC security is product liability. In general terms, product liability is "[t]he liability of manufacturers of products for harms caused to their customers."[297] Simply put, product liability allows victims of defective products to sue the manufacturers and sellers of such products. If successful, victims receive compensation. Product liability provisions may follow one of two main logics: negligence or strict liability. In negligence regimes, the manufacturer or seller is liable if their product is defective and they failed to observe a certain level of care.[298] In contrast, in strict liability regimes, the manufacturer or seller can be held liable independently of whether they have exercised a certain level of care, since the defectiveness of the product is essential.[299] Depending on the design of the provisions, non-complying vendors can be sanctioned either through legal proceedings by the harmed consumers, referred to as the private right of action, or through enforcement by a government regulator.[300] In general, the importance of product liability and the regulatory approaches vary among jurisdictions and legal systems.[301] A separate instrument that is often discussed in the context of product liability is cyber liability insurance, through which manufacturers and sellers can mitigate potential liability charges.[302]

From the SSC security standpoint, product liability provisions can incentivize manufacturers – and possibly sellers – of software to prioritize security. In the negligence framework, this would mean exercising due care, but in the strict liability framework, this would require developing software that is free from defects – hardly a realistic objective.[303] Instead, the focus should be

297 A. Mitchell Polinsky and Steven Shavell (2010): The Uneasy Case for Product Liability, in: Harvard Law Review 123 (6), pp. 1437-1492.
298 Cornell Law School (2023): Negligence.
299 Cornell Law School (2023): Products Liability; Benjamin C. Dean (2018): Strict Products Liability and the Internet of Things, Center for Democracy & Technology.
300 Trey Herr, Robert Morgus, Stewart Scott, and Tianjiu Zuo (2022): Buying Down Risk: Cyber Liability, Atlantic Council.
301 Helmut Koziol, Michael D. Green, Mark Lunney, Ken Oliphant, and Lixin Yang (Eds.) (2017): Product Liability: Funda-mental Questions in a Comparative Perspective. De Gruyter.
302 Josephine Wolff (2022): Cyberinsurance Policy. Rethinking Risk in an Age of Ransomware, Computer Fraud, Data Breaches, and Cyberattacks. MIT Press.
303 Trey Herr, Robert Morgus, Stewart Scott, and Tianjiu Zuo (2022): Buying Down Risk: Cyber Liability, Atlantic Council.

on reducing known risks, so the negligence framework seems more apt for software products.[304]

In any case, this raises the question of what a reasonable standard of care entails. In this context, a standard of care for developing software may build upon instruments assessed earlier: compliance with technical standards, undertaking conformity assessments, and implementing secure software development practices, including CVD and SBOM.[305] Depending on the legal system, such a standard of care may also emerge through established industry practice.[306] For example, "[a]n effective standard [of care] might well create legal obligations to set 'end-of-life' dates for software, remove copyright protections that inhibit security research, or block the use of certain software languages that have inherent flaws or make it difficult to produce code with few errors."[307]

Product liability regimes offer the advantage of possibly changing the behavior of software-developing entities toward SSC security because, if they fail to comply, they risk having to pay significant damages. Put plainly, product liability can fundamentally revise the incentive structure of markets for software products because "it makes no sense to pay liability compensation for damage done when spending money on security is cheaper."[308] Product liability can therefore be seen as a way to enforce the other instruments discussed above if they are specified as the relevant standards of care.

However, translating the product liability logic to software presents several challenges:

1. Existing product liability regimes (e.g., in the EU in general and Germany in particular)[309] refer only to tangible products, which opens up debates about whether "standalone software and applications"[310] qualify as such. Such regimes would therefore need to be adapted.
2. The complexity of SSCs raises questions about which entities can ultimately be held liable.[311] This is why some experts, including the US Cyber Solarium Commission, argue for a "final goods assembler approach," in which the entity that places the product on the market is liable.[312]

304 Trey Herr, Robert Morgus, Stewart Scott, and Tianjiu Zuo (2022): Buying Down Risk: Cyber Liability, Atlantic Council.
305 ENISA (2020): Advancing Software Security in the EU.
306 Jakob Theurer, Johannes Reinsberg, Leopold Borst, and Philipp Bosch (2021): Perspektiven der Produkthaftung und Produktsicherheit in der Industrie 4.0, Wolters Kluwer.
307 Trey Herr (2020): Software Liability Is Just a Starting Point, Lawfare.
308 Bruce Schneier (2003): Liability Changes Everything, Schneier on Security.
309 Anne-Kathrin Müller (2019): Software als "Gegenstand" der Produkthaftung. Zugleich eine Betrachtung des Verhältnisses von § 823 ABs. 1 BGB zum Produkthaftungsgesetz. Deutscher Wissenschafts-Verlag.
310 Anke Krause and Oliver Becker (2022): Liability for Software under the current Product Liability Directive, Linklaters.
311 Trey Herr, Robert Morgus, Stewart Scott, and Tianjiu Zuo (2022): Buying Down Risk: Cyber Liability, Atlantic Council.
312 US Cyberspace Solarium Commission (2020): Report; Trey Herr, Robert Morgus, Stewart Scott, and Tianjiu Zuo (2022): Buying Down Risk: Cyber Liability, Atlantic Council.

3. In many existing product liability regimes – especially those following the strict liability approach – defective products are considered those that cause physical harm, death, or property damage.[313] However, in many cases, defective software products and specifically, SSC compromises, do not cause physical harm, death, or property damage.
4. The different entities that are part of SSCs are in very different positions to meet product liability requirements and to face potential sanctions. For example, a multinational software company will be in a different position from that of an SME or of most entities in the open-source ecosystem.[314] Accordingly, a software product liability regime needs to be mindful of the needs and requirements of these types of entities.

Product Liability: Government Action

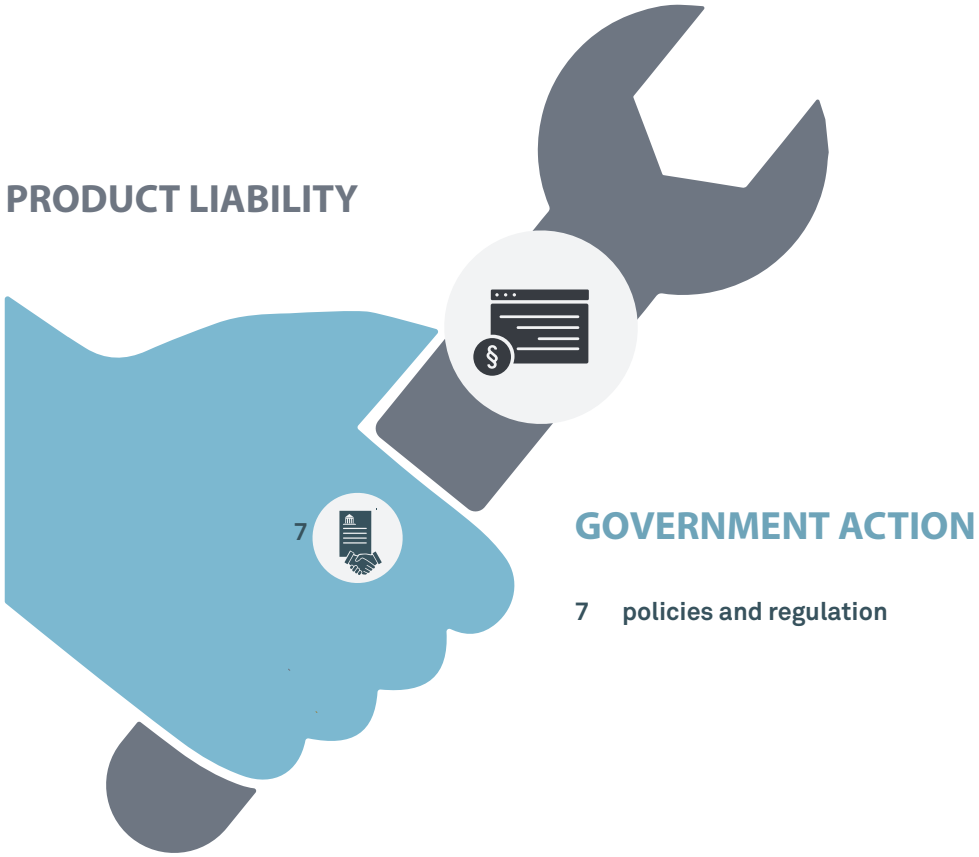


Figure 8: Overview of government action for product liability

313 Council Directive 85/374/EEC of 25 July 1985 on the approximation of the laws, regulations and administrative provisions of the Member States concerning liability for defective products [1985] OJ L210/29; Benjamin C. Dean (2018): Strict Products Liability and the Internet of Things, Center for Democracy & Technology.

314 Bruce Schneier (2008): Software Liabilities and Free Software, Schneier on Security; Chinmayi Sharma, John Speed Meyers, James Howison (2022): The Securing Open Source Software Act Is Good, but Whatever Happened to Legal Liability?, Lawfare.

Governments can implement a product liability regime explicitly covering software through **regulation**. Such a product liability regime has to clarify at least five issues:[315]

1. The software products covered;
2. The entities that can be held liable;
3. In the case of a negligence regime: the specific standard of care;
4. The mode of enforcement (private right of action versus public enforcement); and
5. In the case of a negligence regime: The duration of the entity's required application of the standard of care after bringing a product to market.

The standard of care should build on and reflect the other instruments discussed above, particularly, the technical standards and secure software development practices. In addition, the product liability regime should adequately consider the four aforementioned challenges specific to software in the product liability context.

An example of a proposed product liability regime explicitly covering software products is that proposed by the European Commission. The EU's current product liability regime dates back to 1985[316] and generally does not cover software products that are standalone, in the sense that they are not sold as part of hardware products.[317] In addition, the regime covers only safety defects and applies only in cases of damages related to consumer health, loss of life, and the destruction of items and property. Damages that can result from SSC compromises, such as data loss, are not covered.[318] The EU's 2022 Cyber Resilience Act draft legislation envisions a strict liability regime that would also cover standalone software.[319]

It should be noted that mere discussions about establishing a product liability regime that would cover software products, as in the case of the EU, may lead software-developing entities to consider its potential effects and possibly take steps toward greater SSC security – or leave the market should the potential liability constitute a significant business risk, which could be

315 Trey Herr (2020): Software Liability Is Just a Starting Point, Lawfare; Jane Chong (2020): The Challenge of Software Liability, Lawfare; Trey Herr, Robert Morgus, Stewart Scott, and Tianjiu Zuo (2022): Buying down risk: Cyber liability, Atlantic Council.
316 Council Directive 85/374/EEC of 25 July 1985 on the approximation of the laws, regulations and administrative provisions of the Member States concerning liability for defective products [1985] OJ L210/29.
317 Anke Krause and Oliver Becker (2022): Liability for Software under the current Product Liability Directive, Linklaters.
318 ENISA Advisory Group's Working Group on a cybersecurity consumer perspective (2019): Opinion. Consumers and IoT security.
319 European Commission (2022): Proposal for a regulation of the European Parliament and of the Council on Horizontal Cybersecurity Requirements for Products with Digital Elements and Amending Regulation (EU) 2019/1020.

described as the "shadow of liability." Therefore, policy makers considering establishing such a regime could share a draft version with software-developing entities to capture some of these benefits and help the organizations prepare for eventual legislation.

Product liability regimes for software are not likely to be harmonized internationally among a large number of states because of differences in the respective general product liability provisions of different jurisdictions. Nevertheless, there could be dialogue – and eventually harmonization – regarding, for instance, the entities to be held liable or the standard of care as a way to move toward common overarching principles.

Product Liability: Priorities

Since in many cases, established product liability regimes do not cover software, there is limited experience with product liability for software and, accordingly, there is a lack of analyses on the effects of such regulation. Still, judging from the effect of product liability on product quality in other industries and analyses on the issue,[320] a liability regime is expected to have a significant impact on the incentive structure of software-developing entities for implementing the previous four instruments. In this sense, product liability can be a significant enforcing instrument and thus, has the potential to more broadly change the security practices of the software industry. At the same time, any product liability regime needs to have specific protections for the open-source software ecosystem and SMEs. This is a complex issue for which good solutions have yet to be developed.

320 Trey Herr, Robert Morgus, Stewart Scott, and Tianjiu Zuo (2022): Buying down risk: Cyber liability, Atlantic Council; Benjamin C. Dean (2018): Strict Products Liability and the Internet of Things, Center for Democracy & Technology; Trey Herr (2020): Software Liability Is Just a Starting Point, Lawfare; US Cyberspace Solarium Commission (2020): Report; Chinmayi Sharma, John Speed Meyers, James Howison (2022): The Securing Open Source Software Act Is Good, but Whatever Happened to Legal Liability?, Lawfare; Anke Krause and Oliver Becker (2022): Liability for Software under the current Product Liability Directive, Linklaters.

4 Software Supply Chain Security as a Cyber Norm Implementation Issue

SSC security is not only a matter of domestic policy but also a key element of states' foreign policy, as policy makers have already committed internationally to tackling this issue. The final report of the 2014–2015 UN Group of Governmental Experts on Developments in the Field of Information and Telecommunications in the Context of International Security (GGE) includes a provision on SSC security, norm (i), which specifies that:

> *"States should take reasonable steps to ensure the integrity of the supply chain so that end users can have confidence in the security of ICT products. States should seek to prevent the proliferation of malicious ICT tools and techniques and the use of harmful hidden functions[.]"*[321]

All UN member states agreed to the cyber norms – that is, to the collective expectations of the international community about rules for appropriate behavior regarding the use of ICTs by states[322] – laid out in the report.[323] However, it is unclear exactly what states could or should do to bring this abstract commitment to life. This was not thoroughly remediated by a 2021 document that aimed to provide further guidance on what each of the norms means. The final report of the 2019–2021 GGE proposed that states can put in place frameworks for supply chain risk management, develop policies that promote good practices among vendors, facilitate international competition and innovation, and exchange good practices internationally.[324] However, the report did not specify which issues must be addressed concretely in these measures. Further multistakeholder cyber norm initiatives that touched on the topic of SSC security remained similarly vague.[325]

321 UNGA (2015): Group of Governmental Experts on Developments in the Field of Information and Telecommunications in the Context of International Security (A/70/174).

322 Tim Maurer (2020): A Dose of Realism: The Contestation and Politics of Cyber Norms, in: Hague Journal on the Rule of Law 12, pp. 283-305.

323 While the report was drafted by a small group of representatives from 20 states (see UNGA (2015): Group of Governmental Experts on Developments in the Field of Information and Telecommunications in the Context of International Security (A/70/174)), it was endorsed by the entire UNGA (see UNGA (2015): Resolution 70/237: Developments in the field of information and telecommunications in the context of international security (A/RES/70/237)).

324 UNGA (2021): Group of Governmental Experts on Advancing Responsible State Behaviour in Cyberspace in the Context of International Security (A/76/135). This report was endorsed by all UN member states (see UNGA (2021): Resolution 76/19: Developments in the field of information and telecommunications in the context of international security, and advancing responsible State behaviour in the use of information and communications technologies (A/RES/76/19)).

325 Paris Call for Trust and Security in Cyberspace (n.d.): The 9 principles; Cybersecurity Tech Accord (n.d.): Cybersecurity Tech Accord; Charter of Trust (n.d.): Our 10 Principles; Global Commission on the Stability of Cyberspace (2019): Advancing Cyberstability – Final Report.

In short, having agreed that SSC security is important, governments now need to translate the very abstract cyber norms into concrete policies, a process referred to as "norms implementation."[326] States have recognized that the effective implementation of cyber norms is a necessary step for strengthening international peace and stability[327] and have made concrete advances. Individual states have produced reports on how they implement each of the 11 norms in the 2015 GGE report.[328] Also, under the auspices of the UN, states have developed a *National Survey of Implementation of United Nations* recommendations on responsible use of ICTs by states in the context of international security,[329] which allows states to share information about their implementation efforts. Norms implementation is also repeatedly discussed in the current cybersecurity forum at the UN, the Open-ended Working Group on security of and in the use of information and communications technologies (OEWG).[330] However, none of these efforts addresses SSC security in detail.

This paper therefore provides guidance to states interested in demonstrating commitment to an international agreement to ensure the integrity of SSCs. The recommendations herein specifically address prospects for international cooperation and coordination and thus, for diplomatic action, which can happen in different constellations, including in multilateral, regional, and like-minded settings. Put differently, states that follow some or all of the recommendations outlined in this paper will not only improve their domestic policy but will also support global cyber security resilience and boost their cyber diplomacy ambition by strengthening cyber norms.

326 Bart Hogeveen (2022): The UN norms of responsible state behaviour in cyberspace. Guidance on implementation for Member States of ASEAN, Australian Strategic Policy Institute.

327 UNGA (2021): Group of Governmental Experts on Advancing Responsible State Behaviour in Cyberspace in the Context of International Security (A/76/135); UNGA (2021): Resolution 76/19: Developments in the field of information and telecommunications in the context of international security, and advancing responsible State behaviour in the use of information and communications technologies (A/RES/76/19); Council of the European Union (2022): Council conclusions on ICT supply chain security.

328 Among the states that have published national implementation reports are Australia (see Department of Foreign Affairs and Trade (2020): Australian Implementation of Norms of Responsible State Behaviour in Cyberspace); Canada (see Global Affairs Canada (2019): Canada's implementation of the 2015 GGE norms); and the United Kingdom (see U.K. Foreign & Commonwealth Office (2019): Non-Paper on Efforts to Implement Norms of Responsible State Behaviour in Cyberspace, as Agreed in UN Group of Government Expert Reports of 2010, 2013 and 2015).

329 UNIDIR (2023): National Survey of Implementation of United Nations recommendations on responsible use of ICTs by states in the context of international security.

330 United Nations Office for Disarmament Affairs (2023): Open-ended Working Group on security of and in the use of information and communications technologies.

5 Conclusion

In this analysis, we developed a toolbox for policy makers seeking to increase software supply chain (SSC) security.[331] SSCs are complex, the software development life cycle can be compromised at various stages, and SSC compromises are aggravated by underlying root causes inherent to the entire software ecosystem.

There is ample room for government action to increase SSC security. Our toolbox shows policy makers a selection of instruments at their disposal and how they can put these into practice. The five instruments include three quality assurance instruments, secure software development practices, coordinated vulnerability disclosure (CVD), software bill of materials (SBOM), and product liability. If policy makers use these instruments, they can make a lasting contribution to increasing SSC security.

Which instruments policy makers choose to implement – and if so, through which government action – will depend on factors that are specific to their jurisdiction, such as the salience of the SSC security issue, the political culture, and available resources and capabilities. The toolbox explicitly encourages such a "pick-and-choose" approach that allows individual states to tailor the individual tools to their specific requirements while ensuring the relevance of the tools to a wide range of governments. At the same time, certain combinations of instruments and government actions stand out for their impact on SSC security. We recommend the following three sets of priority government actions to cater to the varying requirements of different governments.[332]

331 For an overview of the toolbox and an explanation of its components, see Section 3.
332 All of these government actions are explained in detail in Section 3.

Level 1: Basics First

Any government interested in increasing SSC security should take these actions, as they constitute the most fundamental government actions that many other government actions build on. These actions require limited resources and capabilities, can be implemented in a short timeframe, and can draw on existing best practices.

Secure software development practices: Include secure software development practices in software developer education and workforce development.

CVD: Issue guidance for organizations on how to set up organizational CVD policies, including templates.

SBOM: Issue guidance specifying data formats for and technical tools building on SBOM data.

Level 2: Ambitious but Tried and Tested

Governments that want to take SSC security a step further than the basics should continue here. These government actions require limited resources and capabilities and can be implemented within short to medium timeframes. All these actions have been implemented in different jurisdictions, thus offering inspiration for their concrete implementation.

Quality assurance instruments: Convene national and international stakeholders involved in quality assurance for coordination and exchange of good practices.

Secure software development practices: Issue guidance tailored to the needs of different types of organizations, on how to implement secure software development practices.

CVD: Adapt governmental processes to require software-developing government agencies to develop and publish their organizational CVD policies, and issue public procurement guidelines that require organizations supplying to the public sector to have organizational CVD policies in place.

SBOM: Convene stakeholders in existing or new forums to discuss challenges and possible future avenues for SBOM use.

Level 3: Breaking New Ground

This set of government actions caters to governments that want to lead the way in increasing SSC security. These most ambitious actions require significant resources and high capabilities on the part of the implementing government entities. In some cases, these actions would cover new ground, as good practices are not yet available, and many of them would take time to implement.

Quality assurance instruments: Dedicate funding to assessing the effects of quality assurance tools on SSC security. If the results are positive, provide funds for the development of new standards or adaptations of existing standards relevant to SSC security, and establish a national – and ideally, an internationally harmonized – conformity assessment scheme and product security labeling scheme for software.

Secure software development practices: Develop regulation that would mandate software-developing entities to implement secure software development practices while considering the peculiarities of different types of software-developing entities.

CVD: Develop a national legal framework for CVD that, inter alia, requires software-developing entities to put in place organizational CVD policies.

SBOM: Allocate funding to advancing the development of SBOM data formats and technical tools that build on SBOM data, and develop regulation mandating SBOM use, for instance, starting with software-developing entities that supply to critical infrastructure providers.

Product liability: Develop or amend an existing product liability regime that covers software and is mindful of the situations of different types of software-developing entities.

Furthermore, SSCs often cross borders, so SSC security poses an international challenge. Accordingly, in many cases, the policy response will be most effective **if international coordination and cooperation will be considered from the start**. Representative examples of this include:

- The development of international technical standards, whether for product security or SBOM data formats;
- The mutual recognition and eventual harmonization of conformity assessment and product security labeling schemes;
- The exchange of guidance and best practices for secure software development practices;
- The harmonization of regulation on CVD to arrive at a common disclosure environment; and
- The harmonization of SBOM requirements in public procurement guidelines.

As a starting point, all of these require dialogue platforms to address SSC security. In many cases, like-minded coalitions will provide the most fruitful starting point for such international efforts.

Finally, increasing SSC security is not just a matter of domestic policy but also serves to implement an international cyber norm, which all UN member states formally endorsed in 2015. This is why policy makers' implementation of the toolbox will serve two purposes at once. First, they will contribute to increasing cybersecurity for stakeholders in their respective jurisdictions. Second, they will implement the UN GGE norm and will thus strengthen and advance the broader cyber diplomacy framework. Following decades[333] of cyber diplomacy discussions at the UN and in other forums, the abstract cyber norms must be translated into concrete policies to demonstrate credible commitment to the framework and to incentivize more states to adhere to the norms.

333 The UNGA discussed the first resolution on cybersecurity in 1998. See UNGA (1998): General Assembly official records, 53rd session : 79th plenary meeting, Friday, 4 December 1998, New York.

About Stiftung Neue Verantwortung

The Stiftung Neue Verantwortung (SNV) is an independent, non-profit think tank working at the intersection of technology and society. The core method of SNV is collaborative policy development, involving experts from government, tech companies, civil society and academia to test and develop analyses with the aim of generating ideas on how governments can positively shape the technological transformation. To guarantee the independence of its work, the organization has adopted a concept of mixed funding sources that include foundations, public funds and corporate donations.

About the Project "Pathways to Implementation – From Cyber Diplomacy Commitments to National Policies"

The goal of this project is to develop concrete proposals for translating cyber diplomacy commitments such as the 11 UN cyber norms into national policies. Over the course of this project, SNV experts also follow developments around the UN Open-Ended Working Group on cybersecurity. This project is funded by the German Federal Foreign Office.

About the Authors

Dr. Alexandra Paulus is Project Director for Cybersecurity Policy and Resilience at Stiftung Neue Verantwortung. Her expertise covers cyber diplomacy, German and European cyber foreign policy, and cyber norms implementation. She leads SNV's cyber diplomacy projects.

Christina Rupp is Project Manager for Cybersecurity Policy and Resilience at Stiftung Neue Verantwortung. Her work focuses on cyber diplomacy and cyber foreign policy, especially cyber norms, as well as Germany's cybersecurity architecture.

Contact the Authors:

Dr. Alexandra Paulus
Project Director Cybersecurity Policy and Resilience
Cyber Diplomacy Projects
apaulus@stiftung-nv.de
+49 (0) 30 81 45 03 78 80
Twitter: @ale_paulus

Christina Rupp
Project Manager Cybersecurity Policy and Resilience
crupp@stifung-nv.de
+49 (0) 30 81 45 03 78 80
Twitter: @christinacrupp

Imprint

Stiftung Neue Verantwortung e.V.
Beisheim Center
Berliner Freiheit 2
10785 Berlin

T: +49 (0) 30 81 45 03 78 80
F: +49 (0) 30 81 45 03 78 97
https://www.stiftung-nv.de/en
info@stiftung-nv.de

Design:
Make Studio
www.make-studio.net

Layout:
Celeste Meisel