September 2022

COMMUNICATIONS SECURITY, RELIABILITY, AND INTEROPERABILITY COUNCIL VIII

REPORT ON RECOMMENDED BEST PRACTICES TO IMPROVE COMMUNICATIONS SUPPLY CHAIN SECURITY

DRAFTED BY
WORKING GROUP 5: MANAGING SOFTWARE & CLOUD SERVICES SUPPLY CHAIN SECURITY FOR COMMUNICATIONS INFRASTRUCTURE

Table of Contents

1 Executive Summary

The telecommunication service providers in the U.S. provide critical voice and data communication services which hundreds of millions of people depend upon daily. The National Emergency Number Association (NENA) estimates that 240 million 911 calls are made in the U.S. each year.[1] These voice and data services are made possible through an ecosystem of service providers, equipment manufacturers, and software vendors.

Recent breaches of trusted software vendors have exposed risks in segments of the supply chain that have resulted in previously trusted systems becoming compromised. These recent breaches have highlighted that the threat is pervasive and extends well beyond the telecommunications network itself and into software components and cloud-based services that service providers rely on to manage and operate their networks. Attacks on these operational networks could have a significant impact on emergency 911 calls and national security communications.

As service providers transform and evolve into the next generation of service offerings, new vulnerabilities are emerging, and the surface area of attack is growing. The transition from a traditional proprietary single vendor appliance model to a virtualized compute environment consisting of software from multiple software vendors and possibly cloud service providers results in vertical and horizontal disaggregation in the service provider's network. As the Nation emerges from the COVID19 pandemic and recovers from major cyberattacks on various widely used software products, we are now fully realizing the potential impacts of supply chain security issues.

The FCC tasked CSRIC VIII, delegated to Working Group 5, to produce two reports focused on supply chain security in the context of telecommunications. This first report is focused on software supply chain security in this new ecosystem with service providers, cloud service providers, and software vendors to identify recommended best practices to improve communications software supply chain security. The second report, in May 2023, will focus on infrastructure (hardware) and network management systems supply chain security.

Summary of Key Findings and Recommendations

The working group has reviewed several relevant and recent related industry news, security events, and publications as part of their research. This first report discusses examples of software supply chain attacks or vulnerabilities in the 2020-2022 timeframe, including SolarWinds, Kaseya, and Apache log4j. The discussion about SolarWinds and Kaseya illustrates cyber attacks on commercial software products and the impacts such attacks can have on downstream customers of those firms. The Apache log4j vulnerability illustrates the risks associated with widespread use of open source software in numerous commercial software products.

The working group has identified some of the most common software supply chain vulnerabilities and corresponding recommendations on how to address those vulnerabilities. The research and analyses are documented in Section 4 and Section 5 of this report. Key findings along with identifying key vulnerability and associated recommendation are available in detail in Section 6 of this report.

[1] NENA, *911 Statistics* (Feb., 2021). https://www.nena.org/general/custom.asp?page=911statistics

Key findings and recommendations:

- New **vulnerabilities are emerging,** and the surface area of attacks is growing impacting the current software supply chain.
- Service providers may source software from different vendors and cloud service providers as they transform and evolve into the next generation of service offering. **SBOM guidance and oversight by governmental and industry actors should therefore consider the broad set of software vendors and cloud service providers** that have important roles in the supply chain.
- Even with these published supply chain enhancements, there are still gaps in the industry today that need modernizing. **SBOM operationalization is a work in progress and additional work is required**. This work group has created a list of **SCRM enhancement considerations**.
- For the service provider, software vendor, and cloud service provider, there is no clear and concise definition for the minimum data fields required for a SBOM. This **lack of industry standardization** needs to be addressed.
- **Cybersecurity operations** could have been a capability that may have alerted the service provider, software vendor, and/or cloud service provider to the attack(s) and mitigated recent software supply chain cyber attacks. This work group suggests that broader discussions within the industry should be conducted to possibly engage in some studies on runtime security.
- While this work group was not specifically charted to provide a security report on open source software, this report does provide several open source security recommendations but the topic itself should be researched independently in a future CSRIC session.

This report includes Table 6 - Recommendations for Service Providers, Software Vendors and Cloud Service Providers, grouping them into 5 groups with 18 key vulnerabilities and recommendations associated with them.

2 Introduction

In the past two years, there have been a number of high profile cyber attacks on the software supply chain which has impacted virtually every service provider in the U.S. As the trend in software supply chain attacks rise, the industry in general has been working diligently to address the supply chain vulnerabilities. This report will evaluate several of the high profile software supply chain cyber attacks in an attempt to better understand the tactics, techniques, and procedures (TTPs) of the attackers. The work group completed analysis on the key industry efforts to mitigate the known software supply chain vulnerabilities. Using the TTPs and the completed analysis of the industry activities, the work group is able to identify key findings and recommendations that can be used by the industry to further strengthen the software supply chain against future cyber attacks.

For those in the telecommunications industry, the evolution and introduction of new technologies is dizzying. In recent years, the telecommunications infrastructure has been evolving from bare metal to virtual commodity-based compute platforms. This virtualization enables the service providers to scale their networks to meet customer usage demands, implement new capabilities

(e.g., 5G Advanced), increase workload capabilities, improve performance, and reduce operational costs. For private clouds inside service providers' networks, this new model of virtualized compute platforms combined with an operating system (e.g., RedHat, Windows) and a network function application (e.g., Access and Mobility Function) creates significant complexities in the software supply chain for service providers and software vendors. When hyperscale cloud providers (HCP) are introduced into this new model to offer both private and public cloud instances, managing the risk to the software supply chain only becomes more challenging.

According to a Synopsys 2022 Open Source Security and Risk Analysis Report[2], 97% of the codebases they audited in 2021 contained open source software and 81% of the codebases had at least one risk open source vulnerability. A typical compute stack consists of the virtualization infrastructure, cloud computing platform, operating systems, and applications. This compute stack will require a management and orchestration platform which consists of multiple management components and automation tools that allow the applications to scale in an elastic nature. The entire compute stack and management and orchestration functions are built upon some of the same open source software that is reported in the Synopsys report. Synopsys reports that 85% of the audited codebases contained open source that was more than four years out-of-date which means that the software is not being patched.[3] Unpatched vulnerabilities are one of the five ways that organizations get initially compromised[4] and should be seen as a critical opportunity to improve supply chain security.

These technological advancements have created a plethora of complications and challenges particularly to end-to-end interoperability in a multi-vendor environment. The industry was focused on the cybersecurity controls, requirements, and specifications for the individual advancements which mandated broader security controls by design. With the industry's focus on designing in additional security requirements and controls into the products and services being delivered, software supply chain security was not necessarily a top priority collectively. As a result of this fact and the recent software supply chain attacks, the FCC CSRIC VIII has established this work group to publish two reports on the topic. This first report will identify some recommended best practices to improve the communications software supply chain security and the second report, due in Q2 2023, will focus on infrastructure and network management systems.

2.1 CSRIC Structure

CSRIC VIII was established at the direction of the Chairperson of the Federal Communications Commission (FCC) in accordance with the provisions of the Federal Advisory Committee Act, 5 U.S.C. App. 2. The purpose of CSRIC VIII is to provide recommendations to the FCC regarding ways the FCC can strive for security, reliability, and interoperability of communications systems. CSRIC VIII's recommendations will focus on a range of public safety and homeland security-

[2] Synopsys 2022, *Open Source Security and Risk Analysis Report*, https://www.synopsys.com/software-integrity/resources/analyst-reports/open-source-security-risk-analysis.html?intcmp=sig-blog-supplychain
[3] Depending on the organization's appetite for risk, priorities and the vulnerability's severity or exploitability, vulnerabilities need not be patched.
[4] Tim Rains, *Cybersecurity Threats, Malware Trends, and Strategies*, Packt Publishing (2020).

related communications matters. The FCC created informal subcommittees under CSRIC VIII, known as working groups, to address specific tasks. These working groups must report their activities and recommendations to the Council as a whole, and the Council may only report these recommendations, as modified or ratified, as a whole, to the Chairperson of the FCC.

Communications Security, Reliability, and Interoperability Council (CSRIC) VIII					
CSRIC VIII Working Groups					
Working Group 1: 5G Signaling Protocols Security	Working Group 2: Promoting Security, Reliability, and Interoperability of Open Radio Access Network Equipment	Working Group 3: Leveraging Virtualization Technology to Promote Secure, Reliable 5G Networks	Working Group 4: 911 Service Over Wi-Fi	Working Group 5: Managing Software & Cloud Services Supply Chain Security for Communications Infrastructure	Working Group 6: Leveraging Mobile Device Applications and Firmware to Enhance Wireless Emergency Alerts
Co-chairs: Brian Daly, AT&T & Travis Russell, Oracle	Co-chairs: Mike Barnes, Mavenir & George Woodward, RWA	Co-chairs: Micaela Giuhat, Microsoft & John Roese, Dell	Co-chairs: Mary Boyd, Intrado & Mark Reddish, APCO	Co-Chairs: Todd Gibson, T-Mobile & Padma Sudarsan, VMware	Co-chairs: Farrokh Khatibi, Qualcomm & Francisco Sanchez, SBA
FCC Liaison: Ahmed Lahjouji	FCC Liaison: Zenji Nakazawa	FCC Liaison: Jeff Goldthorp	FCC Liaison: Rasoul Safavian	FCC Liaison: Saswat Misra	FCC Liaison: James Wiley

Table 1 - Working Group Structure

2.2 Working Group 5 Team Members

Working Group 5 consists of the members listed below.

Name	Company
Rob Alderfer	Charter Communications
Tom Anderson	Alliance for Telecommunications Industry Solutions
John-Luc Bakker	BlackBerry Corporation
Donna Bethea-Murphy	Inmarsat
Shirley Bloomfield	NTCA – The Rural Broadband Association
Matt Carothers	Cox Communications
Josh Cech	S&T Telephone Cooperative Association
Dana Golub	Public Broadcasting Service
Anu Jagannath	ANDRO Computational Solutions
Mohammad Khaled	Ericsson
Jason Lish	Lumen Technologies, Inc.
Martin McGrath	Nokia
Maureen Mclaughlin	Satellite Industry Association
George Popovich	Motorola Solutions
Travis Reutter	ACA Connects – America's Communications Assoc.
Nasrin Rezai	Verizon Communications

John Roznovsky	Mavenir
Sean Scott	SecuLore Solutions
Jim Stringer	AT&T, Inc.
Richard (Dick) Tenney	DHS CISA
Claire Vishik	Intel
Kelly Williams	National Association of Broadcasters
Timothy Wilson-Johnston	Cisco Systems, Inc.
Henry Young	BSA | The Software Alliance
Tim Youngblood	T-Mobile
Timothy May	NTIA
Colin Andrews	Telecommunications Industry Association
Padma Sudarsan (Co-Chair)	VMware
Todd Gibson (Co-Chair)	T-Mobile

Table 2 - List of Working Group Members

Alternates for members are listed below.

Name	Company
Tom Breen	Secure Lore Solutions
Mark Carmel	Cybersecurity and Infrastructure Security Agency, U.S. Department of Homeland Security
Cathleen Dwyer	Verizon Communications
Brandon Hinton	Satellite Industry Association
Tamber Ray	NTCA – The Rural Broadband Association
Mark Roy	Public Broadcasting Service
John Schiel	Lumen Technologies, Inc.
Reza Arefi	Intel
Brian Hurley	ACA Connects
Jason VonBargen	Charter Communications
Mike Regan	Telecommunications Industry Association

Table 3 - List of Working Group Alternates

2.3 Subject Matter Expert Contributors

The working group heard from several subject matter experts during their research.

Name	Company
Allan Friedman, PhD	Cybersecurity and Infrastructure Security Agency, U.S. Department of Homeland Security
Tamara Philip	Synopsys

Table 4 - List of Subject Matter Experts

3 Objective, Scope, and Methodology

3.1 Objective and Scope

The FCC tasked CSRIC VIII, delegated to the Working Group 5 (WG5), with identifying key security vulnerabilities and recommended best practices to improve communications supply chain security. This report focuses on recommendations for the service providers, software vendors, and cloud service providers that provide hardware and software solutions used in a service provider's network. Early on, WG5 realized that the discovery effort for this report would be broad and extensive for two reasons. First, the service provider industry is transitioning from traditional proprietary bare metal platforms from a single vendor to a virtualized compute environment consisting of software from multiple software vendors and possibly cloud service providers. This transition is introducing multiple vendors in the compute and network stacks which could introduce new vulnerabilities by the vertical and horizontal disaggregation in the service provider's network. Second, as the Nation emerges from the COVID19 pandemic and recovers from major cyberattacks on various widely used software products, we are now fully realizing the potential impacts of supply chain security issues.

The key objective of WG5 has been to identify recommended best practices, rank order them based on the participants' experience and corporate backgrounds and subdivide the ranking into those most applicable to large and small service providers, software vendors, and cloud service providers across the industry.

3.2 Methodology

The basic research plan for this report has been to solicit real-world inputs and contributions from WG5 members and invite guest speakers and subject matter experts to share insights during the work group meetings. The work group members evaluated recent industry Executive Orders, government agency publications, industry publications, industry forum's responses, standards development organization's specifications, and recent supply chain cyber attacks. The work group captured their analysis highlighting the key aspects including their findings and recommendations to further strengthen the specific artifact reviewed. The work group identified a few key findings and recommendations from all of the evaluated artifacts with the goal to move the needle forward to providing sustainable and repeatable supply chain security ecosystem for the service providers, software vendors, and cloud service providers.

4 Recent Software Supply Chain Attacks & Vulnerabilities

The following section discusses examples of software supply chain attacks or vulnerabilities in the 2020-2022 timeframe, including SolarWinds, Kaseya, and Apache log4j. The discussion about SolarWinds and Kaseya illustrates cyber attacks on commercial software products and the impacts such attacks can have on downstream customers of those firms. The Apache log4j vulnerability illustrates the risks associated with widespread use of open source software in numerous commercial software products.

4.1 SolarWinds

SolarWinds announced in December 2020 that their Orion Platform network monitoring product

had been modified by a state-sponsored threat actor by embedding backdoor code into a legitimate SolarWinds library. This backdoor enabled remote access into the victim's environment and a foothold in the network, which was used by the attackers to load other malicious software both in memory and in storage resulting in potentially significant compromise of the target system.

This global attack campaign was thought to be initiated as early as March 2020 and affected thousands of public and private organizations.

This multi-phase attack started with a software supply chain compromise attack. Attackers used various sophisticated defense evasion techniques such as masquerading, code signing, obfuscated files or information, indicator removal on host, and virtualization/sandbox evasion.

Phase 1 - Supply Chain Insertion: A SolarWinds build server was compromised with a backdoor to allow the attacker to engineer a "virtually undetectable" insertion of malware into a valid software library file. The attack on the SolarWinds development environment build server was called "SUNSPOT". The vulnerability was inserted into the object code, not source code. Source code refers to the human readable instructions used to create machine executable code that runs on a compute platform. Software build tools create object code from source code. Since no hint of the vulnerability was in the source code, the object code (normally built by software tools) was thought to be clean.

Phase 2 - Distribution of the Malware: Since the infected library file was properly signed, the file was distributed using the normal software update process.

Once deployed, the supply chain backdoor (called SUNBURST) took great pains to remain undetected by using techniques such as:

- Time delay before activating
- Checks of local configuration to verify target is valid and not a test environment
- Check for malware monitoring/scanning tools to prevent detection

Phase 3 - Calling Home: Once all the various checks have passed, the malware attempted to contact a C2 (Command and Control) domain. If this contact attempt fails (e.g., the software does not have Internet access due to network segmentation constructs), the malware goes dormant.

Phase 4 - Propagation and Malware Insertion: If contact to the C2 domain is successful, the threat actor then leverages a memory-only payload called TEARDROP to deliver a variety of other malicious payloads into the target system. The net result is that various targeted malware is loaded resulting in the potential:

- Acquisition of sensitive information in storage and in memory
- Acquisition of privileged credentials to support lateral movement to propagate the intrusion

Mitigation Recommendations:

- Successful mitigation of the attack was realized by companies that segregated their management network (which the SolarWinds Orion platform used to provide their monitoring and management function) preventing the malicious code from contacting the C2 domain which resulted the malicious code going dormant. In general, network segregation techniques can be applied not only to management traffic, but also to the various classes of

control plane as well as user plane traffic. Network segregation can help isolate affected software and thus make it difficult if not impossible for the malicious software to contact a command-and-control server as well as limit lateral movement within the system.

- Since the threat actor was able to insert the vulnerability into the automated software build process, the software supplier may have been able to prevent the build server compromise by applying secure software development best practices. Examples of software development best practices can be found at the National Institute of Standards and Technology (NIST) Computer Resource Center.[5] In particular, National Institute of Standards and Technology (NIST) Special Publication (SP) 800-218,[6] provides a core set of high-level secure software development practices that can be integrated into each software development life cycle implementation.

Generally, this guidance, which provides an update based on the directives from Executive Order 14028 (see section 5.1 for detailed discussion) is founded on established secure software development practice documents from organizations such as BSA – The Software Alliance, the Open Web Application Security Project (OWASP), and the Software Assurance Forum for Excellence in Code (SAFECode). Similarly, NIST Special Publication 800-204C[7] concerns the implementation of development, security, and operations (DevSecOps) for a microservices-based application with service mesh. Following these practices should help software producers reduce the number of vulnerabilities in released software, mitigate the potential impact of the exploitation of undetected or unaddressed vulnerabilities, and address the root cause of vulnerabilities to prevent future recurrences. Supply chain attacks represent another "initial access" threat surface that are difficult to detect and protect against. One potential mitigation technique is to utilize runtime application self-protection (RASP) capabilities to detect these types of events. The software vendor generally has the most knowledge of their code including any embedded free/open source software. A such, they would be in the best position to implement a RASP capability to notify the system when functions/libraries/binaries are misbehaving.

4.2 Apache Log4j Flaw

Apache Log4j (log4j) is part of the Apache Logging Services, a project of the Apache Software Foundation. The widely reported and discussed Apache Log4j vulnerability was first reported by Alibaba in November 2021. The Apache Foundation released an initial patch in December 2021 to remedy the remote code execution (RCE) bug reported under CVE 2021-44228. Subsequent vulnerabilities in different versions of log4j were discovered which has resulted in four CVEs to date[8].

The Java Naming and Directory Interface (JNDI) is a Java API for directory services. Using

[5] https://csrc.nist.gov/
[6] NIST SP 800-218, *Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities*
[7] NIST SP 800-204C, *Implementation of DevSecOps for a Microservices-based Application with Service Mesh*, March 8, 2022. https://csrc.nist.gov/publications/detail/sp/800-204c/final
[8] CISA Creates Webpage for Apache Log4j Vulnerability CVE-2021-44228 | Published December 13, 2021.

JDNI, log4j provides software developers with features for identifying and assessing a Java application's performance including debugging, maintenance, retrieval, and storage of information about an application's runtime performance. It also allows Java clients to query the data by a name. It is independent of any specific directory implementation, allowing a variety of directories to be accessed in a common way, including Lightweight Directory Access Protocol (LDAP).

As an example of its exploitation, the ability for Java clients to query data by name combined with LDAP provides a communication protocol that applications use to communicate with other directory servers. Key to remote code execution is the LDAP (like other directories), which stores user identification, passwords, and computer accounts and shares that information with other entities on a network. Various versions of log4j are vulnerable to a remote code execution (RCE) attack where an attacker who gains permissions to modify the logging configuration file can construct a malicious exploit.

Federal Government Response:

To manage its response to log4j, CISA leveraged its Known Exploited Vulnerabilities (KEV)[9] catalog, which contains a listing of products known to harbor an exploitable vulnerability. CISA Director Jen Easterly said more than 2,800 cases of problems linked to log4j in various commercial offerings have been submitted for inclusion in the catalog.[10]

In mid-December 2021, CISA advised it expected the log4j vulnerability to be widely exploited and that potentially millions of devices were likely affected due to use of the log4j in enterprise products like Oracle, Cisco, RedHat, IBM, VMware, and Splunk.[11] Additionally, HCPs utilize log4j as well as their security appliances and developer tools. The flaw highlights the potential risks arising from software supply chains when a key piece of software is used within multiple products across multiple vendors and deployed by their customers around the world.

In December 2021, CISA published the following statement by CISA Director Jen Easterly to its web page:

> "[CISA] and the Joint Cyber Defense Collaborative …. have established a JCDC senior leadership group to coordinate collective action and ensure shared visibility into both the prevalence of the … [log4j]… vulnerability and threat activity… we are also convening a national call with critical infrastructure stakeholders … where CISA's experts provide further insight and address questions. … To be clear, this vulnerability poses a severe risk. We will only minimize potential impacts through collaborative efforts between government and the private sector. We urge all organizations to join us in this essential effort and take action."

CISA further stated:

> "This effort also underscores the urgency of building software securely from the start and more widespread use of Software Bill of Materials (SBOM), both of which were directed

[9] CISA KEV Catalog. https://www.cisa.gov/known-exploited-vulnerabilities-catalog
[10] CISA's Jen Easterly Warns of Log4j Vulnerability's Long-Term Risk (2022). https://executivegov.com/2022/01/cisas-jen-easterly-warns-of-log4j-vulnerabilitys-long-term-risks/
[11] US warns Log4j flaw puts hundreds of millions of devices at risk, ZDNet, published Dec 14, 2021.

by President Biden in his Executive Order issued in May 2021. A SBOM would provide end users with the transparency they require to know if their products rely on vulnerable software libraries."[12]

4.3 Kaseya VSA – Remote Monitoring & Management Software

The REvil Attack:

In July 2021, Kaseya's Virtual System Administrator (VSA), a remote monitoring and management software, was attacked by REvil, a Russia-based ransomware operation. The source of the outbreak was identified to be a zero-day authentication bypass vulnerability in the VSA software, which allowed attackers to compromise VSA and distribute a malicious payload through hosts managed by the VSA software, thus extending the reach of the attack. Kaseya shut down its VSA cloud and Software-as-a-Service (SaaS) servers and issued a security advisory to its customers, including those with on-premises deployments of VSA.

The VSA tool is a remote management and maintenance suite used by managed service providers (MSPs) to manage their clients. The authentication bypass gave the attackers the ability to upload their payload to the VSA server, which they then executed via SQL injection. This in turn pushed a REvil ransomware payload down to the systems managed by the compromised VSA server and began to execute the ransomware portion of the attack.[13]

The REvil ransomware gang publicly claimed to have encrypted more than one million systems during the incident. They initially asked for a $70 million ransom to release a universal decryptor to unlock all affected systems. On July 5, Kaseya said that between 800 and 1,500 downstream customers were impacted in the attack.

Government Response:

After a July 2021 phone call between U.S. President Biden and Russian President Putin, President Biden told the press, "I made it very clear to him that the United States expects when a ransomware operation is coming from his soil even though it's not sponsored by the state, we expect them to act if we give them enough information to act on who that is."[14]

On 13 July 2021, REvil websites and other infrastructure vanished from the Internet. On July 23, 2021, Kaseya announced it had received a decryptor tool for the REvil-encrypted files and was helping victims restore their files. The FBI provided Kaseya a decryptor for the ransomware but needed three weeks to test it. The FBI tested it to ensure the threat actors were not deploying additional backdoors through the key.

On November 8, 2021, the U.S. Department of Justice (DOJ) unsealed indictments against Ukrainian national Yaroslav Vasinskyi and Russian national Yevgeniy Polyanin. Both were charged with ransomware attacks against multiple victims including Kaseya. Vasinskyi was arrested in Poland in October 2021 and arraigned in Federal Court in Texas on March 9, 2022. Polyanin was charged with conducting ransomware attacks against multiple victims including

[12] Statement from CISA Director Easterly on "LOG4J" Vulnerability, CISA, published December 11, 2021.
[13] The Kaseya/REvil Attack Explained, Bugcrowd, published July 7, 2021.
[14] Biden urges Putin to 'take action to disrupt' Russia-based hackers behind ransomware attacks, July 9, 2021. https://www.nytimes.com/2021/07/09/us/politics/putin-biden-ransomware-hackers.html

Texas businesses and government entities. DOJ worked with a number of governments and law enforcement agencies and announced the seizure of $6.1 million tied to ransomware payments.[15]

5 Analysis of Current Industry and Governmental Efforts

5.1 Executive Order 14028

U.S. Presidential Executive Order 14028, "Improving the Nation's Cybersecurity", issued May 12, 2021, introduces the requirement for SBOMs as a prerequisite for US Government software purchases. The EO directs several federal departments and agencies to take specific actions, including directing the Secretary of Commerce to provide guidance about the minimum elements of a SBOM and other related parameters.

Apart from the SBOM requirement, section 4e of EO 14028 also calls for software producers to indicate conformity with secure software development practices by providing artifacts to federal agency purchasers and/or attesting to conformity.[16]

5.1.1 NTIA – Minimum Elements for a Software Bill of Materials

The National Telecommunications and Information Administration (NTIA) is part of the U.S. Department of Commerce. Under the direction of the Secretary of Commerce, NTIA issued "The Minimum Elements for a Software Bill of Materials (SBOM)."[17] The main purpose of a SBOM is to provide an understanding of what software components are present in supplied software. Service provider systems are potentially made up of many individual systems from different suppliers, and therefore need a common format and development process for the SBOM. The NTIA document states that the primary security use case for SBOM today is to identify known vulnerabilities and risks in the software supply chain.

This guidance recommends the following minimum attributes and fields to ensure that a software component is completely identified:

- **Supplier Name**: The name of an entity that creates, defines, and identifies components.
- **Component Name**: Designation assigned to a unit of software defined by the original supplier.
- **Version of the Component**: Identifier used by the supplier to specify a change in software from a previously identified version.
- **Other Unique Identifiers**: Other identifiers that are used to identify a component or serve as a look-up key in relevant databases.
- **Dependency Relationship**: Characterizing the relationship that an upstream component X is included in software Y.

---

[15] FBI decision to withhold Kaseya ransomware decryption keys stirs debate, ZDNet, published September 24, 2021.
[16] NIST, *Software Supply Chain Security Guidance Under Executive Order (EO) 14028 Section 4e* (Feb. 4, 2022). https://www.nist.gov/system/files/documents/2022/02/04/software-supply-chain-security-guidance-under-EO-14028-section-4e.pdf
[17] U.S. Department of Commerce, National Telecommunications and Information Administration, *The Minimum Elements for a Software Bill of Materials (SBOM)* (July 12, 2021). https://www.ntia.doc.gov/files/ntia/publications/sbom_minimum_elements_report.pdf

- **Author of SBOM Data**: The name of the entity that creates the SBOM data for this component.
- **Timestamp**: Record of the date and time of the SBOM data assembly.

The following fields are recommended but not among the essential minimum fields:

- **Component Hash**: Hashes are encouraged in SBOMs but are not one of the minimum fields. Adding a cryptographic hash of the component is the most precise way to identify a component, effectively acting as a unique identifier. However, correct hashing can be complex and unintuitive. For example, when hashing binary code, it is often difficult to match hashes due to compiler differences and parameter settings. Binary composition analysis tools, for example, may be used to verify or generate SBOMs, but hashes generated by these tools for statically linked libraries have limited use.

Use of incomplete or tampered SBOM information is counterproductive and can have severe consequences. An SBOM consumer may be concerned with verifying the source of the SBOM data and its integrity,[18] as well as its veracity.[19]

5.2 Director of NIST – Publish Guidance of Practices that Enhance Software Supply Chain Security

Executive Order 14028, Section 4 directed NIST to identify existing (or develop new) standards, tools, best practices, and other guidelines to enhance software supply chain security. The target of this directive was for federal agencies, but some practices are relevant for broader consideration as well.

On May 11, 2022, NIST published guidance on software supply chain security.[20] NIST evaluated the existing standards, tools, and then generated some recommended practices. Specifically, NIST published an updated definition of critical software, software supply chain security guidance and recommended some minimum standards for vendor or developer verification of software.

Key Suggestions:

- **Critical Software** - while the definition of "critical software" may vary from that defined for federal agencies, it is paramount that software supply chain security should be initially focused on "critical software".
- **Cybersecurity Posture** - security measures for critical software do not end with utilizing the recommended software security development practices. There needs to be a recognition that breaches are inevitable. Strengthening user access and data protection mechanisms bolsters security. It is also important to have strong incident detection, response, and recovery capabilities, including maintaining an accurate software inventory to quickly respond to zero-

[18] Ibid., p16.
[19] NTIA *Software Consumers Playbook: SBOM Acquisition, Management, and Use* (November 11, 2021). https://www.ntia.gov/files/ntia/publications/software_suppliers_sbom_production_and_provision_-_final.pdf
[20] NIST Guidance on Software Supply Chain Security. https://www.nist.gov/system/files/documents/2022/05/11/Guidance%20on%20Software%20Supply%20Chain%20Security_EO14028%20Sections%204c_4d%5B71%5D.pdf

day vulnerabilities.

- **Application Security Validation and Testing** - guidelines for the minimum standard required for vendor testing of software applies to software source code written by the vendor as well as that in libraries and packages with the understanding that libraries and packages cannot be tested as thoroughly as source code. The minimum standard outlines several tests that should be feasible for all vendors: threat modeling, SAST, DAST, and SCA. Additional testing that could be required for larger vendors includes penetration testing, security auditing and ethical hacking.
- **SBOM Automation and Cyber SCRM** - the preliminary guidance provided by NIST relied primarily on SP 800-161r1[21] – Cybersecurity Supply Chain Risk Management [C-SCRM] Practices for Systems and Organization for current standards, tools, and recommended practices. The evolving standards, tools, and recommended practices focus primarily on vulnerability management. The SBOM provides a software inventory of libraries and packages. As new vulnerabilities are identified, the SBOM would be used to identify where those vulnerabilities exist in vendor delivered software. Automating the SBOM ingestion process allows for quick identification of where software vulnerabilities are present. The SBOM should be used as a companion to the C-SCRM practices, not as a replacement.
- **Secure Software Development Lifecycle and Operations** - NIST has incorporated practices that enhance the security of the software supply chain into the NIST SP 800-218 document. This NIST document focuses on secure software development practices that look to reduce vulnerabilities, reduce the impact of undiscovered vulnerabilities, and address the root cause of the vulnerabilities.

5.3 U.S. Department of Commerce and Department of Homeland Security - Assessment of the Critical Supply Chain Supporting U.S. Information and Communications Technology Industry

In February 2022, the Commerce and Homeland Security Departments jointly published an extensive assessment on supply chains supporting the Nation's information and communications technology industries (ICT).[22]

The executive summary of the assessment highlights the Nation's dependency on foreign suppliers such as China for critical electronic products and their assemblies. The widespread use of open source software and the outsourcing by OEMs for firmware development represents a risk to the supply chain due to a lack of transparency into the suppliers' programming and cybersecurity standards. The dependency on foreign suppliers has opened U.S. based software vendors and cloud service providers to additional external risks of intellectual property theft which could be used maliciously by bad actors.

The assessment makes eight key recommendations to strengthen the U.S. ICT supply chain

[21] NIST's Cybersecurity Supply Chain Risk Management Practices for Systems and Organizations Special Publication 800-161r1. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-161r1.pdf
[22] US Department of Commerce and US Department of Homeland Security, *Assessment of the Critical Supply Chains Supporting the U.S Information and Communications Technology Industry* (Feb. 24, 2022), pp. 4-5. Available at: Assessment of the Critical Supply Chains Supporting the U.S. ICT Industry | Homeland Security (dhs.gov)

resiliency. This work group has identified three recommendations specific to software supply chain and those are listed below:

- **Build Resilience through Secure and Transparent Supply Chains:** Promote supply chain risk management practices through procurement and monitoring efforts such as implementing an Assured Supplier Program for PCBs for Federal Government and establishing a Critical Supply Chain Resilience Program at the Department of Commerce.
- **Collaborate with International Partners to Improve Supply Chain Security and Resiliency:** Improve international engagements through existing fora to advance shared interests in the ICT industry. These interests include bolstering supply chain security and diversity for critical products, strengthening trade enforcement, and enhancing participation in international standards development.
- **Engage with Industry Stakeholders on Resiliency Efforts:** Strengthen public-private engagements to promote awareness and adoption of risk mitigation techniques and best practices for securing the ICT supply chain.

5.4 GSMA Network Equipment Security Assurance Scheme

The Network Equipment Security Assurance Scheme (NESAS)[23] is widely known as a security assurance framework amongst mobile network operators and vendors in the mobile industry. Originally introduced in 2020, NESAS is jointly defined by global standards organizations The 3rd Generation Partnership Project (3GPP) and Global System for Mobile Communication Association (GSMA) with the goal to provide a common security assurance framework for mobile networks. While the demands for securing the supply chain and security assurance increases, GSMA NESAS promotes supply chain security by encouraging mobile products and software vendors to follow secure development and product lifecycle processes that are aligned with industry standards and best practices and making sure that the security capabilities in their products are evaluated against the 3GPP Security Assurance Specifications (SCAS) specifications.

The NESAS evaluation consists of two stages:

- **Stage 1** - an independent security audit on the equipment vendor's product development lifecycle processes including secure coding, security testing, software delivery security, and so forth. GSMA has published a series of specifications and guideline documents that are used by the auditor.
- **Stage 2** - an independent network equipment security evaluation on the vendor's product based on 3GPP-defined SCAS which are the security test cases that will be executed against the vendor's product for compliance evaluation.

Stage 1 - NESAS Security Audit on the Product Development Lifecycle Processes

In the security auditing process, independent security auditors who are accredited by GSMA perform the security audit of the vendor's product development process, software development

[23] GSMA Network Equipment Security Assurance Scheme (NESAS). https://www.gsma.com/security/network-equipment-security-assurance-scheme/

process, and product lifecycle assessments are conducted against security requirements that cover the following areas:

- Security by design
- Version control systems
- Change tracking
- Source code review
- Security testing
- Staff education
- Vulnerability remedy processes
- Vulnerability remedy independence
- Information security management
- Automated build process
- Build environment control
- Vulnerability information management
- Software integrity protection
- Unique software release identifier
- Security fix communication
- Documentation accuracy
- Security point of contact
- Source code governance
- Continual improvement
- Security documentation

Stage 2 – 3GPP SCAS Evaluation of Network Equipment

The 3GPP SCAS specifications[24] define what security properties need to be checked in product implementations of different network nodes.

3GPP has defined a baseline SCAS (3GPP TS 33.117) that applies to all products and then individual SCAS for each 5G network function defined by 3GPP. To complete a SCAS test against a particular network function, the baseline SCAS and the specific network function SCAS would need to be executed.

For NESAS, an independent security test laboratory, accredited according to ISO 17025,[25] evaluate the respective vendor product(s) by executing security testing that confirm compliance with the requirements in the 3GPP SCAS specifications. Once the vendor's product passes and/or meets the SCAS specifications, the product is declared compliant to the specifications. This means that an appointed independent third-party actor has verified security properties of a specific product towards those certain SCAS specification.

5.5 ATIS Standard: 5G Network Assured Supply Chain

The ATIS 5G Network Assured Supply Chain Standard[26] provides requirements necessary to operationalize a set of agreeable levels of supply chain assurances associated with the deployment and operation of 5G networks. This work is based on a flexible reference model and component flow through the complex 5G supply chain to identify a complete set of controls that can mitigate the identified threats and associated attacks given a specific level of assurance. Attack classes are identified by using defined attributes. These attributes represent a defining quality of an asset (hardware component, module, system, software) and consequently reflects

[24] 3GPP Security Assurance Specifications (SCAS). https://www.3gpp.org/DynaReport/33-series.htm
[25] ISO/IEC 17025 Testing and Calibration Laboratories. https://www.iso.org/ISO-IEC-17025-testing-and-calibration-laboratories.html
[26] ATIS Standard: 5G Network Assured Supply Chain. https://access.atis.org/apps/group_public/download.php/66150/ATIS-I-0000090.pdf

the asset's attackable characteristics.

Designating specific system components as "critical" as part of a 5G cybersecurity risk management effort is essential for managing supply chain risks within available resource constraints. Network operators and enterprises must select, shape, and scale their risk mitigation strategy according to business, operational, and security needs. In doing so, they must identify and prioritize a subset of "critical components" that warrants "extra attention" in the supply chain assurance assessment, testing, and monitoring activities.

The approach taken in this document is to leverage, where possible, techniques that can link back to a component's source to verify the authenticity and integrity of that component. SBOM and Hardware Root of Trust (HRoT) represent two methods that can effectively accomplish this goal. In addition, the application of security best practices helps secure each of the supply chain lifecycle functions identified.

The entity responsible for attesting the level of supply chain assurance for a network can use this specification with suppliers by providing:

1. An assurance level, as defined in the standard, that the supplier must comply with.
2. A list of the identified critical components that apply to the supplier.
3. The standard itself, which includes the set of requirements that the supplier must comply to as part of the purchase agreement, along with any desired exceptions and/or additions.

Key Suggestions:

- A supply chain security strategy should be founded on a robust risk assessment based on the application, the needs of the end system, and associated applications that identifies as a critical component(s) which may require additional attention through the supply chain in order to meet an acceptable level of assurance.
- Acceptable levels of assurance should be well defined.
- Robust software vulnerability tracking systems that employ SBOM should be used to better manage software integrity.
- Hardware Root of Trust mechanisms can be used to securely authenticate the underlying platforms used for software and can further be used to attest to the authenticity of the software stack running on this hardware.
- Security Best Practices should be employed across all component life cycle management functions including the overall purchasing and control processes used to manage component production and suppl

5.6 5G Americas' 2021 Security for 5G White Paper

5G Americas is an industry trade organization composed of leading North, Central, and South America based telecommunications service providers and global manufacturers. Some of their Member organizations are T-Mobile, AT&T, VMware, Ericsson, Nokia, Cisco, Telefonica, Shaw, Samsung, Qualcomm, and Mavenir. 5G Americas is partnered with numerous Standards Development Organizations (SDO) such as 3GPP, GSMA, ITU, ATIS, ETSI, and others. 5G Americas leverages their memberships, member's subject matter experts, periodical whitepapers, SDO relationships, and events to influence the LTE and 5G industries.

5G Americas "Security for 5G" white paper highlights a few examples of supply chain attacks including the potential impacts on the water and energy sectors' critical infrastructure. Attacks on the Nation's critical infrastructure could have significant repercussions to the safety and health of the general public. The paper's supply chain recommendations are categorized in the following domains:

- Trusted Suppliers
- Open Source Software Security
- Secure Software Development Lifecycle
- DevSecOps
- Software Bill of Materials (SBOM).

Key Suggestions:

- **Software Composition Analysis:** for open source software, the paper provides guidance on using a Software Composition Analysis (SCA) tool within the developer's integrated development environment (IDE) so that a developer can receive a real-time alert when embedding known vulnerable free and/or open source software including any supporting mitigation recommendations. The paper highlights the importance on leveraging static and dynamic application security testing which combined can assist in identifying vulnerabilities in the code base and in run-time execution.
- **Secure Software Development Lifecycle:** the paper encourages organizations to develop a secure SDLC and DevSecOps programs and includes references to BSA | The Software Alliance (BSA), Open Web Application Security Project (OWASP), SAFECode, and provides some pre-release insights into the NIST's Secure Software Development Framework (SSDF). With the 5G core network being virtualized by using network function virtualization (NFV) on commercial off-the-shelf (COTS) hardware, the paper makes recommendations to build a secured automated orchestration, continuous integration/continuous development (CI/CD) processes and leverage behavioral analytics to help detect compromised software.
- **Software Bill of Materials:** regarding SBOM, the paper highlights the importance of knowing the meta-data structures about the software and its components. The paper suggests the minimum data fields required for a SBOM along with suggestions on SBOM automation that can be used to generate and consume SBOMs (e.g., Software Package Data eXchange (SPDX), CycloneDX, or Software Identification (SWID) tags) to perform dependency checks and vulnerability scans on the identified software. The paper points out that the Department of Commerce recommends that the customer define the cryptographic hashing and/or digital signature of the SBOM in the contractual agreement with the software vendor. From a global perspective, there is no global naming/identity authority for published software, which creates challenges for identification of supplies, upstream vendors, and Free and Open Source Software (FOSS).

5.7 BSA Framework for Secure Software

BSA developed The BSA Framework for Secure Software (BSA Framework) to bring together industry best practices in a detailed, holistic manner that can improve software security regardless of the development environment or the purpose of the software. The BSA Framework offers an outcome-focused, standards-based risk management tool to help stakeholders in the

software industry – developers, vendors, customers, policymakers, and others – communicate and evaluate security outcomes associated with specific software products and services.

Specifically, the BSA Framework helps:

- Software development organizations describe the current state and target state of software security in individual software security products and services.
- Software development organizations identify opportunities for improvement in development and lifecycle management processes and assess progress toward target states.
- Software developers, vendors, and customers communicate internally and externally about software security; and

Software customers evaluate and compare the security of individual software products and services.

5.8 NSTAC Report to the President – Software Assurance

The President's National Security Telecommunications Advisory Committee (NSTAC) published a Software Assurance in the Information and Communications Technology and Services Supply Chain report.[27] High profile cyber attacks that impacted critical infrastructure provided the motivation for the NSTAC study and the resulting report. These same high profile cyber attacks were the driving force behind the Executive Order (EO 14028) that was published soon after the beginning of the NSTAC study. Phase one of the multi-phase report formulated findings and recommendations on various aspects of software assurance and supply chain applicable to a wide range of software systems.

The subcommittee's findings and recommendations fall into three main areas of focus:

- Software assurance
- Stakeholders
- External influencing factors

Key Callouts:

- **Software Assurance:** the subcommittee discovered Software Assurance was found to not have a single software security assurance approach, that Supply Chain Risk Management (SCRM) needs adaptation of its approaches to software, and FOSS is not less secure than commercially developed software, although in need of incentives to emphasize security. The subcommittee recommended that the federal government and industry must collaborate on broad, actionable, and well established SCRM practices; NIST should convene a public-private effort to improve harmonization among standards in security assurance; the federal government should invest in research and development in software assurance to keep up with rapidly expanding technologies; and the federal government and private sector should improve security and assurance processes for FOSS.
- **Stakeholders:** stakeholders are an important part of the software assurance in ICT and the

[27] NSTAC Software Assurance Report, November 2, 2021. https://www.cisa.gov/sites/default/files/publications/NSTAC%20Report%20to%20the%20President%20on%20Software%20Assurance.pdf

services supply chain. The subcommittee found that stakeholders (developers, procurement teams, and administrators) have different requirements that are sometimes in tension. Stakeholders have additional concerns, for example, with providing evidence of software security assurance or with guidelines for software supply chain assurance not keeping up with cloud-based releases and more frequent releases and updates of third-party code modules. The study recommended incentivizing engagement of all groups of stakeholders in software assurance programs at all levels; incentivizing easy-to-adopt software assurance practices; reform and update U.S. government acquisition regulations to drive better SCRM practices; and improve software administrator information sharing practices to increase awareness of and mitigation of risks.

- **External Influencing Factors:** the subcommittee found that the global nature of software development and supply chain make it challenging for "one-size-fits-all" approaches. The subcommittee also agreed that security assurance practices are not taught early in the education system or not taught broadly enough. Recommendations include a task force to define viable incentives convened by the Federal Government, including public-private representation; harmonize and improve the content for teaching software assurance security among engineering students and training programs; and encourage teaching security concepts early in K-12 education.

5.9 Broader SBOM related Industry Initiatives

In 2018, the NTIA convened a multistakeholder process on promoting greater software component transparency[28] culminating in a consensus on the importance of a Software Bill of Materials (SBOM) as a key enabler for providing such transparency. An SBOM is still a nascent practice that has recently been gaining greater attention both in the US and globally, especially since the issuance of EO 14028 mandating all US government procured software have an SBOM.

SBOM information enables various uses cases such as vulnerability management, license compliance management, asset management, and high assurance to name a few. For instance, if a new vulnerability is discovered in a particular software component version, organizations will want to quickly understand if and where they are potentially impacted so they can take remediation action. An SBOM enables this by identifying if such a vulnerable software component version is used and in which product(s) it is used. With a high assurance use case the focus will be on by whom and where the software component was created which can also be identified from a correctly populated SBOM.

In addition to EO 14028, various entities have started to list SBOMs in their requirements including the O-RAN Alliance,[29] ATIS 5G Supply Chain,[30] and TIA SCS 9001.[31] OpenChain, which maintains the International Standard for open source license compliance, has established the Telco Work Group which is focused on developing a telecommunications standard for

[28] https://www.ntia.doc.gov/federal-register-notice/2018/notice-071918-meeting-multistakeholder-process-promoting-software
[29] https://orandownloadsweb.azurewebsites.net/specifications
[30] https://www.atis.org/initiatives/5g-supply-chain-working-group/
[31] https://tiaonline.org/what-we-do/scs-9001-supply-chain-security-standard/

SBOM.[32]

While SBOM momentum is growing, a Linux Foundation report published in January 2022 pointed to concerns that still need to be addressed and overcome. The report titled "The State of Software Bill of Materials (SBOM) and Cybersecurity Readiness"[33] is based on a survey conducted in the second half of 2021 which highlighted the following concerns from respondents:

- Is the industry is committed to requiring SBOMs, or whether it is optional.
- Vendors/end users are unsure about the value of providing SBOMs to their customers.
- Uncertainty as to whether there are tools available that automate the consumption/production of SBOMs.
- Consensus on what an SBOM should contain.

The NTIA has concluded its multistakeholder process on software transparency, however CISA[34] will advance the SBOM work by facilitating community engagement, development, and progress, with a focus on scaling and operationalization. To this end, CISA has facilitated a number of community driven initiatives such as the SBOM-A-RAMA[35] in December 2021, a series of Listening Sessions in July 2022, and periodic work stream meetings starting in August 2022 (on SBOM Sharing, SBOM Adoption, SBOM Tooling, and SBOM Cloud)[36] to further the understanding of SBOM creation, use, and implementation across the broader technology ecosystem.

5.10 TIA'S SCS 9001 Supply Chain Security Standard

The Telecommunications Industry Association (TIA) is an ANSI accredited standards development organization representing more than 400 global companies in the ICT industry.

TIA has developed the *SCS 9001 Supply Chain Security Management System,* a process-based, certifiable standard that addresses the challenges of ICT cybersecurity and supply chain risk management.[37] TIA reviewed existing industry research, government agency publications and existing standards and concluded that a purpose-built standard to address the growing problem of supply chain security was warranted.

A certification to SCS 9001 includes the collection and reporting of key metrics, which are anonymized and used to create industry benchmarking reports identifying average, best and worst in class and used over time to drive continuous improvement. SCS 9001 is well aligned with works from peer SDOs and government agencies and can be leveraged to operationalize the recommendations of such publications.

Key Callouts:

SCS 9001 is intended to provide a higher level of confidence in a vendor's ability to deliver

[32] https://www.openchainproject.org/news/2022/06/01/telco-wg-meeting-2022-06-2
[33] https://www.linuxfoundation.org/tools/the-state-of-software-bill-of-materials-sbom-and-cybersecurity-readiness/
[34] https://www.cisa.gov/sbom
[35] https://www.cisa.gov/cisa-sbom-rama
[36] https://www.cisa.gov/sbom
[37] https://tiaonline.org/what-we-do/technology-programs/supply-chain-security/

more secure products. It does so by providing a broad assessment of the operational practices of the vendor as well as the delivered products and services.

Examples of operational practices include but are not limited to:

- Principles of corporate trust: ensure that vendors are trustworthy and operate with integrity.
- Operational security: ensure vendors practice a high degree of operational hygiene across all functions.
- Management: ensure management drives organizational commitment in meeting cybersecurity and supply chain risk goals.
- Incident Management Process: ensure processes are in place to identify, mitigate, and restore operations upon a security incident including effective communications to customers.
- Vulnerability Management: ensure processes are in place to identify, manage and report on security vulnerabilities wherever they exist.
- Risk Assessment, Mitigation and Management: ensure vendors identify and reduce risks of all types.
- Business Continuity Planning: ensure vendors have implemented effective business continuity plans.
- Human Resource Management and Training: ensure vendors sufficiently train employees.

Examples of development practices include but are not limited to:

- Provenance: ensure vendors source components from trusted suppliers, can trace all components to their origin, and have protections against tampering and counterfeits.
- SBOMs: provide SBOMs in support of software and firmware deliverables.
- Secure Product Development: implement security considerations across the entire product lifecycle.
- Advanced Tooling: vendors should make requisite investments in advanced tools in support of their software development efforts.
- Outsourced Software Development: ensure that contractors and outsourcers employ equally strong development controls.
- Open Source Software: ensure additional controls are applied to open source software due to the potential for tampering.

Key Suggestions:

The industry continues to be challenged with the need to improve approaches to cyber and supply chain security. The definition of a high-quality product or service should include assurance that the vendor accounts for cybersecurity and supply chain risk management as fundamental requirements throughout the entire product lifecycle. Quality and security are not mutually exclusive, they are intricately linked.

1. Leverage available expertise with a collaboration between government agencies, network operators, suppliers, and standards development organizations in solving this problem.
2. Self-attestations to standards or requirements may not be sufficient. Vendors are under business pressures of profitability, staffing, time to market, and supply chain disruptions. Best practices diminish in time due to human behavior and churning of employees. The

industry could benefit with adoption of independently certified standards with regular surveillance auditing and periodic recertification.

3. The industry should consolidate efforts and align around a number of key global standards and publications to ensure consistency of approach on a worldwide basis and to drive cost and time efficiencies.
4. Governments are setting higher expectations of service providers as evidenced by the baseline cyber and supply chain requirements of the U.S. Broadband Equity, Access, and Deployment (BEAD) program as described in the Notice of Funding Opportunity (NOFO) and legislation such as the U.K. Telecommunications Security Act of 2021. In response, it is suggested that service providers work with their vendors to meet the expectations being set by government.

5.11 Limitations of Industry Accepted Vulnerability Management Processes

When new vulnerabilities become public, both suppliers and consumers often rush to patch based more on the notoriety of a flaw than on the real impact. Consumer demand pushes suppliers to focus on headline-grabbing issues that may or may not impact the product, leading to wasted time and effort. Suppliers often struggle to express to consumers why one vulnerability should be prioritized over another, and consumers struggle to express to internal teams why one issue requires immediate patching while another does not.

Reliance on the Common Vulnerability Scoring System (CVSS) to drive patching decisions fuels many of these problems. The CVSS "is an open framework for communicating the characteristics and severity of software vulnerabilities."[38] It scores vulnerabilities on a scale from 0 to 10, with 10 being the most severe. While the CVSS score provides a valuable starting point, it cannot be used alone without context. Other factors must inform decisions.

Key Suggestions:

- Is a vulnerable component actually in use, or is it simply included in a package? For example, a library may contain a vulnerability related to Universal Serial Bus (USB) drivers, but if the device in question has no USB ports, the issue is not important.
- Is the vulnerable component used in an exploitable way? Just because a system uses a vulnerable library does not mean it can be exploited. For example, a command injection vulnerability may not be exploitable if user input to the library is properly sanitized.
- How complex is the exploit? An exploit requiring high skill or an unusual set of preconditions can be prioritized lower than an exploit requiring low skill.
- Is an exploit publicly available? If threats actors can be observed exploiting a vulnerability in the wild, patching for that issue should take higher precedence.

5.12 Zero Trust Model

In a software supply chain, artifacts travel along a series of repositories (source code or binary artifact) as they are transformed from an initial commit to a running artifact in a production environment. Traditionally, the main defense against attackers was to keep them out by relying on trusted network perimeters like firewalls, internal networks, and physical security. A

[38] https://nvd.nist.gov/vuln-metrics/cvss

compromise to any one of the systems in the supply chain pipeline can result in an attack.

A Zero Trust Supply Chain moves artifact repositories out of the Trusted Compute Base. Individuals and build systems attest to source code and artifacts directly. These attestations form a verifiable chain from its origin (developer or system) to final, deployed production artifact. Artifact metadata (including rich provenance) is digitally signed with PKI support. Signed metadata files, or attestations, are stored and accessible in a global transparency log.

The zero trust architecture (ZTA), as NIST terms it, is still a work in progress. NIST SP 800-207, "Zero Trust Architecture," describes earlier work of moving from a perimeter-based security model to one focused on individual transactions. [39] NIST recognizes earlier work of moving from emphasis on trust based on the location of a network and on static defenses to evaluating trust on a transaction basis. In the Federal sector, zero trust principles and practices have evolved in various programs such as risk management frameworks; trusted Internet connections; identity, credential and access management, and continuous diagnostics and mitigation programs.

Industry commentators are looking to tie SBOM and ZTA through an updated DevSecOps program and possibly limiting access of the individual SW components/code to unnecessary compute components (e.g., kernel, memory, cache) and network interfaces/protocols.

Zero trust is a strategy to prevent cybersecurity breaches by eliminating the concept of automatic trust from an organization's supply chain network. In a Zero Trust framework, users have to request privileged access each time they need access to the system. A SBOM is designed to further enable transparency into software components and their developers. Ultimately, maintaining an SBOM, a formal record of software containing details and supply chain relationships of various components used in building software, is critical for organizations to improve their security models and mitigate supply chain disruption.

The increased transparency that SBOMs enables provides an accelerated assessment of risks, vulnerabilities, and dependencies in software. In the case of a crisis, like the Log4j vulnerability, SBOMs can help organizations identify active issues and minimize huge potential financial risks, damages in reputation and loss of productivity. Additionally, SBOMs help achieve compliance with government regulations and foster trust with customers. Combining ZTA with SBOM procedures and policies will make software products safer throughout each segment of the supply chain lifecycle.[40] However, ZTA does require additional resources to develop, train and implement, especially for smaller enterprises and will, in the opinion of WG5, undoubtedly have a long evolution in being widely adopted.

5.13 Synopsys – 2022 Open Source Security and Risk Analysis Report

The Synopsys Cybersecurity Research Center (CyRC) published the seventh edition of their Open Source Security and Risk Analysis Report[41] in 2022, providing "an in-depth snapshot of the current state of open source security, compliance, licensing, and code quality risk in

[39] NIST SP 800-207, p. 2
[40] Zero Trust & Software Bill of Materials (SBOM): why they're mission critical - Blog - Hikvision (Accessed: Aug 3, 2022).
[41] 2022 Open Source Security and Risk Analysis Report, Synopsys, Inc,

commercial software". The report suggests that it is prudent to assume open source will be part of the software that any business uses, explaining that "open source is the foundation for every application we rely on today." By examining audit findings from over 2,400 commercial codebases across 17 industries using software composition analysis, the report aims to increase understanding of risks associated with open source development.

Key Callouts:

Considering all codebases covered by the Synopsys report:

- 78% of code was open source
- 81% of codebases contained at least one vulnerability
- 88% of codebases audited contained components with no new development in two years,
- 85% of codebases contained open source more than four years out-of-date

As a point of improvement, yet still of concern, is the number of codebases containing at least one high-risk open source vulnerability which decreased to 49% compared with 60% in the prior year. Specifically, of all scanned codebases for telecommunications and wireless companies audited by the report, 95% were found to contain open source. Of those codebases from telecommunications and wireless companies, 41% were found to contain open source vulnerabilities. Regardless of industry, Synopsys reported how open source components were found to make up the majority of codebases, and "much of those codebases were vulnerable to exploit and attack."

The variety of open source code further complicates matters, with millions of GitHub projects being maintained by small teams of less than ten people. In contrast, other popular projects may be maintained by large numbers of developers, even companies that have a vested interest. The statistics presented in the report originally produced by the Linux Foundation further illustrate the situation. Findings from the study indicated that 23% of the top 50 non-node package manager (npm) projects had only one developer accounting for more than 80% of the lines of code, and 94% of the projects had fewer than 10 developers accounting for more than 90% of the lines of code. The findings concur with other points in the report indicating that "almost all the most widely used open source is developed and maintained by only a handful of contributors."

Key Suggestions:

- An important distinction outlined by the Synopsys report is that open source itself does not create business risk, but rather the mismanagement of open source does. One example of mismanagement is the embedded open source software not regularly patched within the codebase.
- The Synopsys report describes how the Log4j incident unveiled the inherent trust that organizations place in open source, with developer teams using open source without requiring the same security reviews as would otherwise be in place for commercial or proprietary software.
- In highlighting the Log4j incident, the report suggests that it became apparent that many organizations are altogether unaware of the usage of open source in their software.
- Mention of SBOM as an initial means of addressing business risk was made, considering all software a business uses, regardless of how it was acquired.
- In the simplest terms, the report states that "it's awfully hard to fix something you don't

know about or can't find." This illustrates how the SBOM plays a critical role in addressing risk, by enabling teams to "chart a path forward."

5.14 Impacts of using Open Source Software in the Supply Chain

While open source has served other industries such as banking for years, service providers have been slower to adopt open source and have been slow to participate in the evolution of open source. However, open source is gaining traction and its adoption requires new skills and participation. As described in an IBM Institute for Business Value report,[42] there is a huge paradigm shift in the idea that "...software is the center of value, rather than equipment". Rather than having a focus on hardware-based solutions, network systems built based on software can be modified with greater ease and at a faster interval. This enhancement of capabilities can ultimately benefit the consumer.

The adoption of open source makes cooperation and participation in the open source community more relevant for a provider. A provider can influence future development, including security and functionality, for those projects of interest. Yet this requires allocating resources having the skills and time to contribute, in turn creating some level of resource or other financial impacts. Open cooperation is an important aspect of open source, as group participation reduces the likelihood that a particular software solution is being maintained by only one individual or entity. More attention from developers can potentially increase the probability that flaws are addressed before being exploited. Logically speaking, the same lines could be drawn to closed source software, where having the eyes on the codebase from a team within a company should equally decrease flaws. It is important to re-emphasize the point that open source itself is not the factor creating business risk, rather it is the mismanagement of open source.[43]

A volunteer presenter from within the industry shared details concerning their organization's handling of open source software. The following key suggestions were compiled from the main points shared by the presenter.

Key Suggestions:

- Strict internal requirements should exist to protect the company and its customers
- Third-party suppliers should be held to the same company standards
- Policies should apply equally to open source software as with proprietary software
- Third-party software should be sourced by a centralized configuration management team
- Centralized configuration management teams should ensure sources are reputable
- A gating subprocess should validate that patches are applied and scans are completed
- A post-scan analysis should be used to reveal issue severity, priority, and applicability

This content remains unattributed due to the sensitivity behind exposing organizational methods, approaches, or architectures.

[42] IBM Institute for Business Value, *Telecom answers the open source call*. https://www.ibm.com/downloads/cas/MWL4K98L
[43] 2022 Open Source Security and Risk Analysis Report, Synopsys, Inc,

5.15 Enabling Platform Software Security

Infrastructure software often runs on server platforms that may be configured in virtualized, cloud native, or bare metal arrangements. These server platforms can take advantage of various technologies to verify the authenticity and integrity of platform software using a chain of trust rooted in an embedded Hardware Root of Trust (HRoT). An HRoT must be inherently trusted, and therefore must be secure by design providing a foundation on which all secure operations of a computing system depend. It contains secured and protected keys and can execute cryptographic functions to enable such operations as a secure boot process, secure platform identification (via unique keys verified via the protected cryptographic functions), and remote software attestation.

NISTIR 8320 – Hardware-Enabled Security: Enabling a Layered Approach to Platform Security for Cloud and Edge Computing Use Cases[44], explains hardware-based security techniques and technologies that can improve server platform security and data protection for cloud data centers and edge computing. Hardware-enabled security can provide a stronger foundation than one enabled by software or firmware alone. For example, this technology can enable real-time attestation of platform software prior to workload placement in virtualized and cloud native deployments. In addition, HRoT presents a smaller attack surface due to the small codebase. Existing security implementations can be enhanced by providing a base-layer, immutable hardware module that chains software and firmware verifications from the hardware all the way to the application space or specified security control.

HRoT capabilities are particularly relevant in securing the software supply chain. HRoT supports the ability to assign unique and cryptographically verifiable identities to servers. In addition, HRoT real time attestation of firmware and software enables secure verification of running software to mitigate a number of supply chain vulnerabilities. For example, workload placement functions can request a real time attestation on the target server before placing a critical workload on that server.

The Trusted Platform Module (TPM)[45] has been used for more than twenty years as a hardware root of trust to enhance the integrity of the platform and improve its resistance to some software attacks. Developed by the Trusted Computing Group (TCG)[46] and transposed into ISO/IEC JTC1 SC27 as ISO/IEC 11889,[47] the TPM is a standard component on most platforms, including in the telecommunications space. More recently, TPMs and Platform Certificates[48] began to be considered as solutions to protect integrity in the supply chain, with pilot implementations for a number of different use cases.

5.16 Supply chain Levels for Software Artifacts, or SLSA (salsa)

SLSA[49] is a set of incrementally adoptable security guidelines, established by industry consensus, focused on supply chain integrity, with a secondary focus on availability. The

[44] https://csrc.nist.gov/publications/detail/nistir/8320/final
[45] https://trustedcomputinggroup.org/resource/tpm-library-specification/
[46] https://trustedcomputinggroup.org/
[47] https://www.iso.org/standard/66510.html
[48] https://trustedcomputinggroup.org/resource/tcg-platform-certificate-profile/
[49] https://github.com/slsa-framework/slsa/blob/main/docs/spec/v0.1/levels.md

standards set by SLSA[50] are guiding principles for both software producers and consumers: producers can follow the guidelines to make their software more secure, and consumers can make decisions based on a software package's security posture.

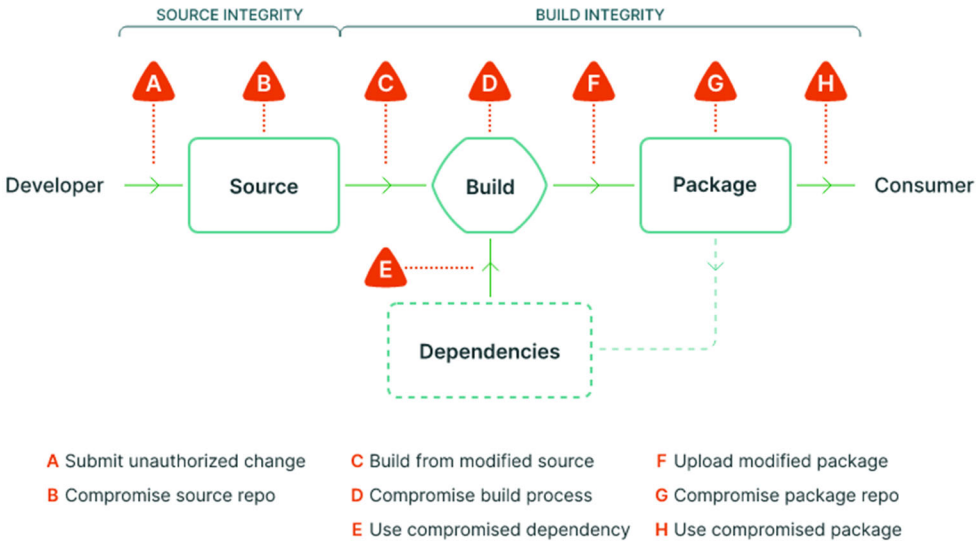

Figure 1 – Supply chain Levels for Software Artifacts[51]

SLSA is organized into a series of levels that provide increasing "integrity" guarantees. Integrity means protection against tampering or unauthorized modification at any stage of the software lifecycle. Within SLSA, integrity is divided into source integrity versus build integrity. SLSA's four levels include Level 1: Documentation of the build process; Level 2: Tamper resistance of the build service; Level 3: Extra resistance to specific threats; and Level 4: Highest levels of confidence and trust. The levels are designed to be incremental and actionable, and to protect against specific integrity attacks. SLSA represents the ideal end state, and the lower levels represent milestones with corresponding integrity guarantees.

High profile attacks or exploits such as SolarWinds, Codecov, or Linux Hypocrite Commits demonstrate these kinds of supply chain integrity vulnerabilities may go unnoticed or be underdeveloped, and quickly become extremely public, disruptive, and costly in today's environment. SLSA is designed with these examples in mind to make sure they are common knowledge and easier to protect against.

5.17 Vulnerability-Exploitability eXchange (VEX)

Since 1999, the public and private sector have collaborated to identify, define, and catalog publicly disclosed cybersecurity vulnerabilities.[52] Most recently, NTIA and CISA have facilitated public and private sector collaboration on the development of the so-called Vulnerability-Exploitability eXchange (VEX), a tool complementary to an SBOM to provide users (e.g., operators, developers, and services providers) additional information on whether a

[50] SLSA is currently in alpha, as indicated by its project status on https://slsa.dev/
[51] https://github.com/slsa-framework/slsa/blob/main/docs/spec/v0.1/levels.md
[52] For example, CVE Program, https://cve.mitre.org/about/cve_and_nvd_relationship.html

product is impacted by a specific vulnerability in an included component and, if affected, whether there are actions recommended to remediate it.[53]

VEX has been implemented as a profile in the Common Security Advisory Framework (CSAF), which is a standard for machine readable security advisories developed by the OASIS Open CSAF Technical Committee, as well as integral part of the OWASP CycloneDX specification. VEX can also provide rich information on vulnerabilities, such as remediation, workarounds, restart/downtime required, scores, and risks that can be provided by vendors, systems integrators, and operators.

The goal of VEX is to allow a software supplier or other parties to assert the status of specific vulnerabilities in a particular product. VEX documents allow both suppliers and consumers to focus on vulnerabilities that pose the most immediate risk, while not investing time in searching for or patching vulnerabilities that are not exploitable and therefore have no impact.[54] To this end, a VEX indicates a status per vulnerability.[55]

- **NOT AFFECTED** – No remediation is required regarding this vulnerability.
- **AFFECTED** – Actions are recommended to remediate or address this vulnerability.
- **FIXED** – These product versions contain a fix for the vulnerability.
- **UNDER INVESTIGATION** – It is not yet known whether these product versions are affected by the vulnerability. An update will be provided in a later release.

Additionally, when the product is indicated as NOT AFFECTED, VEX permits the document to include a justification statement of why the VEX document creator chose to assert that the product's status is NOT AFFECTED. Status justifications range from indicating the product is not affected by the vulnerability because the component is not included in the product to the vulnerable code can never be executed in the context of the application.

While VEX is a recent development, enterprises may already have implemented a capability that facilitates indicating whether products are affected by vulnerabilities or recommend mitigations.

A VEX can advise remediating actions. However, it is important to verify the veracity of the information within the VEX, including any recommended actions. Ideally, a VEX originator should be authenticated and screened, and the VEX itself should be checked for integrity.

Generally, it is recommended to include requirements to provide VEX or VEX-like information in contracts between consumer and developer/supplier.

6 Description, Findings and Recommendations

6.1 Secure Software Supply Chain: Work Group Description

Based on the findings, the work group has identified a description for the secure software supply chain.

Secure Software Supply Chain is a set of practices that enable organizations to adjust the way they securely consume proprietary or open source packages – both first- and third-party – from

[53] https://www.ntia.gov/files/ntia/publications/vex_one-page_summary.pdf
[54] https://www.cisa.gov/sites/default/files/publications/VEX_Status_Justification_Jun22.pdf
[55] https://www.cisa.gov/sites/default/files/publications/VEX_Use_Cases_April2022.pdf

source code to operationalization at a sustained high speed and quality relative to their accepted risk level.

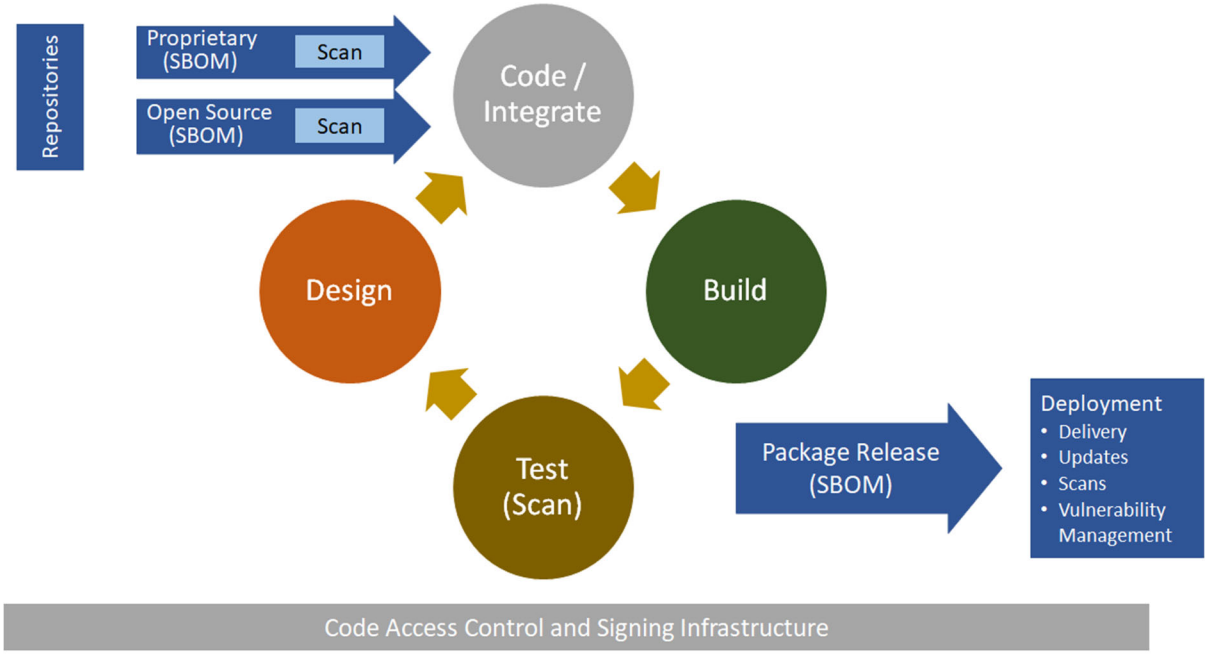

Figure 2 – Secure Software Supply Chain

Figure 2 depicts the iterative process of integrating open source and proprietary software through the build, test, design and integration cycle into a finalized codebase and final release through an SBOM. If third-party, proprietary, or open source components are delivered without SBOM, the organization should generate an SBOM prior to using the component.

This iterative process provides confidence that the code and its dependencies are trustworthy, compliant, up-to-date, and release-ready, as well as ensures regular scans are in place to detect, report, and eliminate vulnerabilities. With a defined set of policies enforced consistently across all systems in the chain, it prevents unauthorized access and prohibits unsigned packages to run.

6.2 Summary of Key Findings

Communications service providers connect millions of households and businesses to jobs, health care, education, entertainment, and one another every day. These connections require the use of software and/or cloud services to perform the functions needed by consumers and businesses. Accordingly, service providers must be able to identify and manage software supply chain risk regardless of their size and whether the provider uses “off the shelf” software, open source software, custom software, cloud services, or any combination of these.

Recent breaches of trusted vendors of software have exposed risks in segments of the supply chain that have resulted in previously trusted systems becoming compromised. These recent breaches have highlighted that the threat is pervasive and extends well beyond the telecommunications network itself to software components and cloud-based services that service providers rely on to manage and operate their networks. Attacks on these operational networks could have a significant impact on emergency 911 calls and national security communications.

6.2.1 Change in Paradigm impacting Current Software Supply Chain

As service providers transform and evolve into the next generation of service offerings, new vulnerabilities are emerging, and the surface area of attack is growing.

Virtualization of software resulting in vertical and horizontal disaggregation

The evolution of cloud services has spawned a growing list of acronyms that are widely used in the marketplace. Prevalent offerings include Software as a service ("SaaS"), Platform as a service ("PaaS"), and Infrastructure as a service ("IaaS"), each with important distinctions in how they are used and secured.[56] Responsibility for the security of information contained in cloud services is typically shared between the cloud service provider and the customer, with the responsibilities shifting according to the type of cloud service used (e.g., SaaS places the highest amount of responsibility for content security on the cloud service provider).[57] Importantly, in each type of cloud service, critical responsibilities remain that must be addressed by those consuming the cloud service, such as service providers. These responsibilities must be clearly identified and understood, both initially and continuously over time. As such, service providers who do not fulfill their end of the shared responsibility may ultimately expose themselves to supply chain security risks.

With the disaggregation of the vertical (e.g., hardware, Communications as a Service ("CaaS"), platform, and application) and horizontal (e.g., cloud native microservices and pod-based architecture) selecting best of breed software has become the norm resulting in multi-vendor deployments; thus, increasing the touch points and sharing responsibility in delivering end-to-end security. This requires clear processes and automated tools to ensure inter-vendor solutions do not have gaps.

Wide adoption of open source software:

Open source is gaining traction and its adoption and has proven instrumental in accelerating software development — providing developers with feature velocity, ease of customization, and quality reusable code. However, hacks of open source software projects are becoming a big concern. Regardless of industry, it has been reported how open source components were found to make up the majority of codebases, and "much of those codebases were vulnerable to exploit and attack."[64] Today's threat actors have no qualms about injecting malicious code upstream as a way to target downstream applications. Developers need to recognize this new reality and rethink security across the software supply chain. While there have been improvements made, specifically of all scanned codebases for telecommunications and wireless companies audited by the report,[58] 95% were found to contain open source. Of those codebases from telecommunications and wireless companies, 41% were found to contain open source

---

[56] Software as a service occurs when a cloud service provider builds, runs, and hosts applications delivered over the internet, which customers pay to access; Platform as a service occurs when a cloud service provider creates an environment, or platform, for customers to build and deliver applications; and Infrastructure as a service takes place when a cloud service provider delivers access to storage, networking, servers, or other computing resources. *See* Cloud Security: A Primer for Policymakers, available at https://carnegieendowment.org/2020/08/31/cloud-security-primer-for-policymakers-pub-82597.

[57] See, e.g., Cybersecurity & Infrastructure Security Agency, Cloud Security Technical Reference Architecture, v. 1.0 (Aug. 2021) at Figure 2.

[58] 2022 Open Source Security and Risk Analysis Report, Synopsys, Inc.

vulnerabilities.

6.2.2 Modernization of the Supply Chain

The Office of the Director of National Intelligence (ODNI) defines supply chain as a network of people, processes, technology, information, and resources that delivers a product or service.[59] Historically, most in the public and private sectors would define a supply chain as something very similar to the ODNI's definition. The SCRM programs were typically focused on risks that could impact or delay the delivery of a hardware product by an agreed to date. For instance, any delays of raw materials, components, and/or subcomponents from third-parties. The manufacturers would provide a hardware bill of materials (HBOM) to the customer which identified all the hardware components in the product. Over time, the SCRM programs started tracking any potential delays with the product's software and/or firmware that could delay the manufacturing and/or final build process which could delay the delivery of the product. The information on the software was primitive at best and it was used as the original software bill of materials (SBOM). For example, a SBOM may have just listed "Product A Software 7.1". This SBOM does not communicate all the individual software components (e.g., FOSS, third-party, etc.) that are used as ingredients for the final software being delivered.

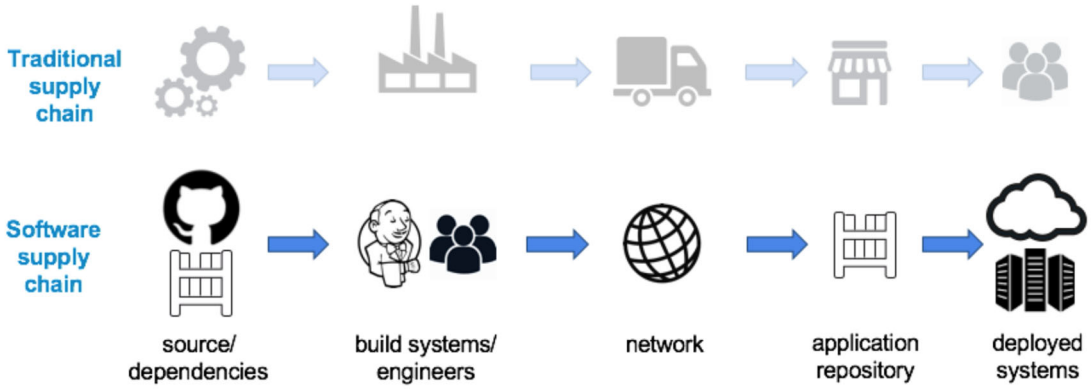

Figure 3 – Software supply chain and traditional industry model similarity[60]

To the ODNI's credit, they have updated their SCRM to include software and firmware into their key information and communications technology (ICT) supply chain. In recent years, much of the public and private sectors have incorporated specific SBOM requirements into their SCRM programs. Today, the SBOM requirements are becoming robust based upon the cyber events identified in Section 4 of this paper as well as other events. Cybersecurity teams within public and private organizations are shaping the SCRM programs to identify cyber risks. As a result of these recent cyber events along with the fact that the industry still cannot deliver an effective SBOM, the Whitehouse issued Executive Order 14028 in May 2021. In response, NIST defined a new Cyber-SCRM (C-SCRM) in May 2022 which covers the entire life cycle of a system (including design, development, distribution, deployment, acquisition, maintenance, and

[59] ODNI - https://www.dni.gov/files/NCSC/documents/supplychain/20200925-NCSC-Supply-Chain-Risk-Management-tri-fold.pdf
[60] https://blog.convisoappsec.com/en/is-your-software-supply-chain-secure/

destruction).[61]

Even with these published supply chain enhancements, there are still gaps in the industry today that need modernizing. To effectively secure the software supply chain and cloud services, the transparency provided through SBOM should continue to reflect relevant and modern components that can be used to enhance cybersecurity. Note that SBOM operationalization is a work in progress and additional work is required. This work group has created a list of SCRM enhancement considerations:

- **Software Bill of Materials (SBOM) Enhancement Considerations:**
  - ***Machine Readable*** - delivered SBOMs need to be machine readable so that the consumer can perform vulnerability and risk assessments on all the software being delivered.
  - ***Hardware and Software Components/Subcomponents*** – all software delivered must be included in the SBOM including firmware, operating system, drivers, etc.
  - ***Embedded FOSS*** - SBOM must include all FOSS that is embedded within the delivered binaries, libraries, packages, etc.
  - ***Software Licensing*** - the software licensing for each of the embedded software components, binaries, packages, etc. must be provided so the consumer can ensure that they are in compliance with the various use licenses.
  - ***Software Versions*** - the software version for each of the embedded software components, binaries, packages, etc. must be provided.
  - ***Access Control*** – for the individual software components, applets, binaries, etc., what access do they have to the kernel, buffers, cache, configurations, memory, storage, transmission media including cryptography in use, APIs, data fields, etc.
  - ***Data Access*** – for the individual software components, applets, binaries, etc., what type of data will this software have access too? For example, any customer information or other sensitive information.
  - ***End of Life/Support*** – for the individual software components, applets, binaries, etc., list any published end of life and/or end of support dates which communicates the end date which software patches will cease to be available.
  - ***Provenance of the Source Code*** – the origin of the software developer(s) should be provided. This includes the employees of the software vendor, contractor(s), 3rd third-party(s), and any FOSS software as well. Any software that is being developed and delivered out of a country of concern should be made known to the software consumer. Providing the provenance may not always be possible as of the date of this report but the industry should move towards this being a requirement in the future.
  - ***SBOMs, Software Repositories and/or Delivery Cryptography Protections*** – the SBOMs including the hosting and transmission of the software components, applets, binaries, etc. must use strong cryptography that does not allow for code manipulation which could inject malware.

Beyond the above noted SBOM enhancement considerations, there are additional SCRM improvement considerations to discuss.

[61] NIST C-SCRM. https://csrc.nist.gov/projects/cyber-supply-chain-risk-management

- **Other SCRM Enhancement Considerations:**
    - ***Cryptographic Agility*** – the hosting and transmission platforms should be capable of rapidly changing the algorithms in use in response to a cryptographic threat. If quantum-safe cryptography[62] is used, it is an option that signing algorithms are based on hybrid cryptography or to use dual signatures. In this case, quantum-safe public-key algorithms are used alongside traditional public key algorithms (e.g., elliptic curves) so that the solution is at least no less secure than existing traditional cryptography. For example, OpenSSH[63] was an early adopter of Post-Quantum Cryptography, and its version 9.0 uses the hybrid Streamlined NTRU Prime and x25519 key exchange method by default. Note that since release of OpenSSH 9.0, NIST has decided not to advance Streamline NTRU Prime past Round 3[64] of the Post-Quantum Cryptography (PQC) standardization process.
    - ***SCRM Attacks/Events Public Communications*** – software vendors and/or consumers should be required to issue public statements to notify the industry when certain SCRM cyber attacks and/or security events occur so that the industry can respond more quickly to disrupt the attack(s)/event(s).
    - ***Cloud Providers and/or Software Vendors with Established CI/CD Pipeline***
        - As the industry continues to evolve and transition from bare metal infrastructure into cloud native environments, the creation and delivery of a SBOM can be more challenging given the frequency that developers and/or cloud providers could deploy updated software images. For the cloud providers and/or software vendors that have established a Continuous Integration / Continuous Deployment (CI/CD) pipeline, they can deploy updated software and applications very efficiently and frequently in a production environment. By having an established CI/CD pipeline, this should not give them a waiver on delivering SBOMs because of the challenges in this domain.
        - **SBOM Creation Automation** – the provider and/or vendor needs to deliver an updated SBOM for every updated software image deployment.
        - **Automated Security Testing** – automated software composition analysis (SCA) and static application security testing (SAST) must be mandated, at a minimum, for all new source code before being compiled and deployed. Dynamic application security testing (DAST) and interactive application security testing (IAST) is recommended as well.
        - **Unpatched Vulnerabilities** – any unpatched vulnerabilities must be communicated to the consumer prior to deployment.
        - **Risk Appetite** – the consumer should be given an option to deploy or not to

[62] NIST, *Post-Quantum Cryptography PQC* January 28, 2020, FAQ. https://csrc.nist.gov/Projects/post-quantum-cryptography/faqs, questions: "is it possible for a hybrid key establishment mode to be performed in a FIPS 140 approved mode of operation?", "is it possible for dual signatures generation or verification to be performed in a FIPS 140 approved mode of operation?", and "does NIST consider the hybrid key establishment modes and dual signatures to be long-term solutions?".
[63] OpenSSH, *OpenSSH 9.0 was released on 2022-04-08* April 8, 2022. https://www.openssh.com/txt/release-9.0
[64] NIST, *Round 4 Submissions* July 5, 2022. https://csrc.nist.gov/Projects/post-quantum-cryptography/round-4-submissions

deploy an updated software image in the CI/CD pipeline if the unpatched vulnerabilities pose too much of a risk to the consumer.

Additionally, use of incomplete or tampered SBOM information is counterproductive and can have severe consequences. This has been highlighted in Section 5.1.1. Integrity and authenticity checking are often supported through signatures and public key infrastructure (PKI). A code signing system that protects sensitive signing keys using hardware protection should be used. Organizations that need to provision and implement digital signatures may specify the use of FIPS-validated cryptography[65]. The veracity of an SBOM can be checked by comparing a received SBOM with one generated by the recipient. In case of discrepancies, it is recommended that the SBOM provider and the recipient seek to resolve the mismatch.

In closing of this section, it is important to note that service providers may source software from different vendors and cloud service providers who develop platforms that ultimately power the telecommunication networks in use today. SBOM guidance and oversight by governmental and industry actors should therefore consider the broad set of software vendors and cloud service providers that have important roles in the supply chain.

6.2.3 Lack of a Complete and Definitive Standard in Software Supply Chain

Across the industry, various Standards Development Organizations, industry forums, and government agencies are attempting to address the software supply chain security concerns. Individually, there are notable recommendations and/or specifications being published that both the public and private sectors should be using as guidance to strengthen their supply chain risk management (SCRM) programs. These emerging artifacts of guidance, specifications, and/or standards have notable differences and conflicts that creates confusion in the industry.

To shed more light on the deltas in the published artifacts, one just needs to compare the outputs from NTIA and TIA on SBOMs. Driven by EO 14028, the NTIA defines a set of baseline data fields that a SBOM should include. TIA's SCS 9001 specifies a baseline SBOM that differs from that of NTIA. The table below highlights the baselines from both entities and provides a visual aid that highlights the deltas between the two artifacts. For the service provider, software vendor, and cloud service provider, there is not a clear and concise definition for the minimum data fields required for a SBOM. This lack of industry standardization will create challenges as the software vendors and cloud service providers attempt to secure contracts with both government agencies and service providers.

As a disclaimer, this work group did not map the minimum data fields between NTIA and TIA. The table below is a raw listing of the high level field names that are captured in the separate artifacts. The table below is a snapshot as of August 31, 2022.

Baseline SBOM Data Fields

[65] The requirement for FIPS 140 validation, as well as timelines for acceptance of FIPS 140-2, and 140-3 can be found at the NIST Cryptographic Module Validation Program (CMVP). https://csrc.nist.gov/projects/cryptographic-module-validation-program

NTIA's Baseline	TIA's SCS 9001 Baseline
Supplier	Supplier Name
Component Name	Component Name or Unique Identifier
Version of the Components	Version
Other Unique Identifiers	
Dependency Relationship	Relationship (including in or derived from)
	Components Relationship
Author of SBOM Data	
Timestamp	
	Mapping to Existing Formats
	Component Hash or Equivalent
	Compatibility Requirements
	Open Source Software Content
	Free Software Content
	Third Party Content

Table 5 - NTIA and TIA Baseline SBOM Data Fields

Post the analysis conducted on the various artifacts in Section 5 "Analysis of Current Industry and Governmental Efforts", there are notable improvements being published and proposed, but there are still areas of opportunities that have not been addressed. As the industry globally attempts to tackle the software supply chain issues, requiring the software vendors and cloud service providers to comply with a set of geopolitical and/or regionalized requirements or specifications is not the best approach for the industry. The industry needs to collaborate to define universal standards and specifications relating to software supply chain security. This will significantly reduce the complexities and provide improved operational efficiencies that should reduce the cost burden on the software vendors and cloud service providers.

To advance the industry forward towards developing a sustainable and repeatable process for software supply chain security, the industry needs to collaborate to evaluate the feasibility of recommendations including:

Foundational Processes

- **SBOM Minimum Data Fields** – the industry needs to codify the minimum set of data fields that are required and optional so that the industry can standardize their SBOM data globally.
- **Software Identification Tags** – the industry needs to coalesce to one or two at most methods to assign the firmware, software, and the individual components. Using a centrally defined body to assign unique identifier tags to individual software components is an option. Preferably, the software identification tags should identify the software developer(s) as well. There are a number of software identifier proposals and solutions that the industry is

coalescing around, but the industry needs to gravitate to as few of these as possible.[66]

- **SBOM Automation** – the SBOMs should be in a machine readable format so that the software vendors, cloud service providers, and service providers can use automation to process the SBOMs in real-time. This includes the creation of APIs that allow for scalable SBOM data queries and validations. This is important for environments that have CI/CD pipelines and frequent software deployments into production environments.

Risk Management Processes

- **Software Provenance Authority** – software supplier(s) must be able to provide the country of origin for all software that is included in any firmware and software that is delivered to software consumer(s). The application of provenance validations must be securely and uniformly implemented.
- **Software Author Identity** – the actual software developer that wishes to develop software should go through an identity verification process that can be authenticated. This process would probably receive considerable backlash from an industry perspective, but it could go a long way towards mitigating bad actors from publishing known vulnerable software in public repositories and it could open up new opportunities for the freelance software developers.
- **Supplier Identity** – the software supplier(s) identities for all embedded software should be provided.
- **Software Supplier and Author Reputation** – the reputation of the software author should be provided so that the software vendors, cloud service providers, and service providers can do a risk analysis on the software prior to it being deployed into a production environment.

Vulnerability Management Processes

- **Software Chain of Custody** – the software consumer(s) should be able to identify the software producer(s), both at the individual developer and at the organization level, that contributed to the final software package that is delivered. This chain of custody should include the countries of origin(s) for the software producer(s).
- **Software Vulnerability and Exposure Disclosures** – there needs to be a method to publish any new vulnerabilities and exposures based upon a unique software identification tag or other unique identifier so that the software vendors, cloud service providers, and service providers can quickly identify if that software is in use within a software package or in a production environment. This could be something similar to CISA/NTIA's Vulnerability-Exploitability eXchange (VEX).[67,68]
- **Unpatched Vulnerabilities** - after the software vendor completes industry approved security testing (e.g., SAST, DAST, IAST, etc.) on the source code and compiled binaries, the software vendor needs to report any unpatched vulnerabilities in the final delivered software so the consumer can determine their risk appetite prior to implementation.
- **SBOM Immutability** – the software vendors, cloud service providers, and service providers

---

[66] Survey of Existing SBOM Formats and Standards, 2021. https://www.ntia.gov/files/ntia/publications/sbom_formats_survey-version-2021.pdf
[67] NTIA, *Vulnerability-Exploitability eXchange (VEX) – An Overview* (September 27, 2021). https://ntia.gov/files/ntia/publications/vex_one-page_summary.pdf
[68] CISA, *Software Bill of Materials*, https://www.cisa.gov/sbom

need a method of confirming that the software has not been tampered with in the supply chain. This could be done via digital signatures or certificates that are signed by approved authorities.

6.2.4 Cybersecurity Operations

As noted in the earlier sections referencing recent software supply chain cyber attacks, runtime security could have been a security capability that may have alerted the service provider, software vendor, and/or cloud service provider to the attack(s). Runtime Application Self-Protection (RASP)[69] is an emerging security capability that enhances the security monitoring on the operating platforms and applications. Additionally, cloud workload protection platform (CWPP) is a solution applicable to cloud-based services.[70] CWPP uses a cloud-based software defined network (SDN) with artificial intelligence (AI) /machine learning (ML) based risk engine to analyze context-based signals and derive a risk score. The context-based signals are output of sensors e.g., OS-level events such as those related to processes, files, network, and memory. When a risk score changes, the AI can be configured to stop adversaries and take various actions in real-time as opposed to relying on the traditional revocation of access credentials triggered by security events.

Commercial RASP solutions are typically constructed to function with web-based APIs and require a learning period. These commercial solutions are generally acceptable for web applications but telecommunications systems, including service providers, require additional intelligence due to industry specific protocols and/or interfaces such as those defined by 3GPP. Performance impacts as well as the actions to be taken when a potential threat is detected are some of the additional key aspects to be taken into consideration when assessing RASP for telecommunication environments. The software vendors and/or cloud service providers that deliver these telecommunications specific platforms know their source code, how it executes, performs, and functions. As the various SDOs and Industry Forums have invested time to define, develop, and publish security requirements, it seems practical that the software producers should invest time and energy to develop runtime security capabilities within their software releases so that the software consumers can more intelligently monitor the platforms for misbehaving software code in their networks.

Understandably, this is a challenging ask, but this work group suggests that broader discussions within the industry should be conducted to possibly engage in some studies to determine the feasibility of such an ask.

6.2.5 Common Best Practices for Software Supply Chain

Some common software best practices are listed below. While they are not directly impacting software supply chain, they are contributing towards secure delivery of software.

- Developer Training: Ensure software development teams are competent. Conduct regular security awareness training.

---

[69] Gartner, Runtime Application Self-protection (RASP). https://www.gartner.com/en/information-technology/glossary/runtime-application-self-protection-rasp
[70] Gartner, What are Cloud Workload Protection Platforms? https://www.gartner.com/reviews/market/cloud-workload-protection-platforms

- Software Composition Analysis (SCA): Leverage SCA tools to determine the contents, origins and versions of all components integrated into a software deliverable. Ensure licenses are compatible. Verify that the most recent versions are being used to ensure no previously known vulnerabilities are introduced.
- Security Scanning: Continue to run vulnerability scans post release. New vulnerabilities are found all the time with improvements in scanners and their databases.
- Outsourced Software Development: Ensure that outsourced software vendors embrace the same security standards as native development teams.
- Credential Management: Implement processes for proper credential management. Permit access to systems only as needed. Ensure credentials are retired upon termination of employment.

6.3 Recommendations

In order to work effectively, both suppliers and consumers of such software should adopt a repeatable, defensible vulnerability management process. The industry continues to be challenged with the need to improve approaches to cyber and supply chain security. At the same time, not every flaw can or should be treated with the same level of urgency and the security teams should adopt a risk-based approach. One such approach is Stakeholder-Specific Vulnerability Categorization[71] (SSVC). SSVC guides organizations through the process of developing a decision tree appropriate for that organization's business model and risk tolerances. It provides a repeatable, transparent process for evaluating a vulnerability, prioritizing/deprioritizing patching, and explaining the decision to both internal and external stakeholders. Example inputs include the CVSS score, context in which the vulnerable software is used, complexity of the attack, and availability of publicly exploitable code. Example outputs include decisions such as emergency patching, patching within normal development cycles, and declining to patch. Another such approach is provided by VEX (see section 5.17).

Maturity levels and expected outcomes, however, will not be the same for every provider or for providers of the same size as the level of security should meet the objectives and capabilities of each provider rather than applying the same supply chain security objectives to every provider. Maintaining a scalable risk management approach will result in more effective supply chain security than a checklist and is consistent with other federal cybersecurity guidelines such as the NIST Cybersecurity Framework[72] ("NIST CSF").

The standards set by Supply chain Levels for Software Artifacts [73] (SLSA) are guiding principles for both software producers and consumers: producers can follow the guidelines to make their software more secure, and consumers can make decisions based on a software package's security posture. SLSA's four levels are designed to be incremental and actionable, and to protect against specific integrity attacks. SLSA 4 represents the ideal end state, and the lower levels represent milestones with corresponding integrity guarantees. Even in the alpha stage, it is being adopted by several software vendors.

[71] https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=636379
[72] NIST Cybersecurity Framework. https://www.nist.gov/cyberframework
[73] https://github.com/slsa-framework/slsa/blob/main/docs/spec/v0.1/levels.md

6.3.1 General Guidance for Organizations regarding SBOMs.

The following outlines some general guidance that organizations producing, consuming, and sharing software could be taking to prepare themselves to build SBOM support into their software supply chain processes.

1) **Expected SBOM Mandates** - organizations providing solutions directly and/or indirectly to the US Government should be aware that SBOM's may become mandatory requirements going forward and as such should prepare now to address this ask.
2) **SBOM Generation and Consumption** - steps organizations may take to prepare for SBOM support:
   a) Build awareness - interested parties should familiarize themselves with SBOM's in general, a good starting point being the NTIA and CISA SBOM websites and resources
   b) Understand what SBOM support means for your organization in terms of:
      i. Contractual Aspects, what are your obligations and what are your requirements towards suppliers/partners e.g., how to share, the frequency and depth?
      ii. How to implement SBOM support to Produce, Consume and Share (e.g., data formats, depths, tools, and processes)?
      iii. How to improve tools, processes, and fill requirement gaps, for example, review existing CISA Listening session and webinars and participate in and contribute to future SBOM initiatives?

6.3.2 Common Software Supply Chain Security Recommendations

The following table identifies key software supply chain security recommendations based on commonly known vulnerabilities and issues relating to the recent cyber attacks referenced earlier in this report as rationale. The rationale reflects potential vulnerabilities that can be mitigated through the stated recommendations. Note that in some cases, context based cyber risk analysis is needed to provide the best set of security recommendations to be applied.

Open source is not intrinsically less secure than that of closed source software, but it requires, in some cases, a different set of incentives to enhance security. The distributed and international nature of some open source communities, differences in expertise among the participants, considerable complexity of some open source projects, and other factors can lead to challenges that various initiatives, including those undertaken by the Linux Foundation, are working to address. While this work group was not specifically charted to provide a security report on open source software, this report does provide several open source security recommendations but the topic itself should be researched independently in a future CSRIC sessions.

Recommendations for Service Providers, Software Vendors and Cloud Service Providers			
Tier (Best Practice Tier 1 = Most Critical)	**Categories/Topic Groupings**	**Vulnerabilities**	**Recommendations**
1	**Secure Development**	**Compromised Source Code Compiler** (build server)	• Apply secure software development best practices as defined by NIST Computer Resource Center Special Publication (SP) 800-218 Secure Software Development Framework (SSDF)[74] additional contextual examples include: • Patch Operating System and Application Software Frequently. • Employ security hardening guidelines and audits. • Leverage Single Sign-On (SSO) and MFA for access. • Eliminate all local identities and restrict access to only the staff that requires access. • Develop advanced security configuration monitoring capabilities. • Monitor accesses and configuration changes as part of an insider threat program. • Enable runtime security capabilities to alert to anomalous behavior. • Perform a cyber kill chain analysis to determine weaknesses in the defense in depth strategy.

[74] NIST SP 800-218 – Secure Software Development Framework (SSDF), https://csrc.nist.gov/publications/detail/sp/800-218/final

1		**Implanted Malicious Source Code**	• Integrate a Software Composition Analysis (SCA) tool into the software development tool chain. • Provide (standardized when available) SBOMs, automate SBOM consumption, and execute vulnerability scan on SBOM components. • Execute static application security testing (SAST), dynamic application security testing (DAST), and interactive application security testing (IAST) frequently. • Update the SCA, SAST, DAST, and IAST vulnerability lists frequently. • Conduct regularly scheduled vulnerability scans and penetration testing. • Develop a security auditing process for the software. • For critical applications, engage ethical hackers and/or establish bug bounty programs. • Independently audit internal and/or vendor software development programs to certify compliance with industry best practices and/or certification programs.
1		**Compromised Internally Developed, Vendor Provided and/or FOSS Software** (malicious code, backdoor, zero day, etc.)	
1	**Free and Open Source Software** (FOSS)	**Malicious or Unvetted FOSS Downloaded from an Untrusted Public Development Platform**	• Implement an internal security vetted repository and open source management process to acquire FOSS packages from trusted sources identified and continually validate sources as reputable. • Frequently update internal repository with patched code and notify developer(s) of the FOSS updates. • Implement governance program to audit, identify, and force patching of FOSS software.
1		**FOSS Vulnerabilities not Expediently Reported Publicly**	• Promote industry collaboration and information sharing to communicate identified compromised code sets publicly prior to waiting on a software patch to be released.

			• Encourage the industry, industry associations, government, and educational institutions to work towards creating a trusted and centralized code evaluation ecosystem. • Software security policies, where practical, should apply equally to open source software as they do with proprietary software within a given organization.
2		**Lack of Secure Coding Practices in FOSS**	• Increase industry participation and investment in the evolution of the open source community, including influence over future development, security, and functionality of projects, particularly those of the highest visibility. • In general, OpenSSF's Security Score Cards[75] can be considered when selecting open source. Scorecards is an automated tool that assesses a number of important heuristics ("checks") associated with software security and assigns each check a score.
1	**Cyber and Supply Chain Risk Management and Processes**	**Disaggregation of Software increases Attack Surface**	• Create and follow a Cybersecurity Risk Management Plan as well as a Supply Chain Risk Management Plan • For critical assets, compel vendors to undergo an industry accepted and independent security audit of their product/software development lifecycles. • Develop robust security testing programs including vulnerability scanning, penetration testing, independent security testing, ethical hackers, and/or bug bounties within controlled test environments. • Explore cloud security posture management (CSPM), SaaS (Software as a Service), Security Posture Management (SSPM), and cloud workload protection

[75] OpenSSF *Security Scorecards - Security health metrics for Open Source*. https://github.com/ossf/scorecard

			(CWP) functionality offered/available (see CISA's Cloud Security Technical Reference Architecture, v2.0, p. 7,8). • Develop highly trained staff with deep knowledge and core competencies on the third-party domains including holistic multi-cloud security monitoring. • Follow the Secure Development recommendations as part of the Defense in Depth cyber strategy.
1		**Vulnerabilities introduced in a Service Provider's Network through a Software Vendor and/or Cloud Service Provider**	• Work with industry, cloud service providers, and software vendors to provide a Software Bill of Materials that can be used to identify whether listed components have known vulnerabilities. • Establish a secure remote connection for all software and cloud service vendors. • Verify the identity, integrity and authenticity of the software and patches through signatures and public key infrastructure (PKI) using hardware protection and authentication. • Use of Hardware Root of Trust (HRoT) based attestation of software can ensure the integrity of loaded software. • Ensure all third-party entities of a software vendor (upstream and downstream) are held to the same security standards of the vendor itself • Leverage CISA's Cloud Security Technical Reference Architecture, v2.0, p. 26 includes Common Requirement for CI/CD Pipelines, as part of Security Testing. • Establish a mechanism linked to the internal gating process to ensure that patches are applied, scans have been completed, and provide an opportunity for

			resolution or mitigation of any defects or vulnerabilities. In terms of scanning, an automated post analysis can be used to reveal context surrounding severity and applicability.
		Vulnerable Software from Untrusted Suppliers	• Develop a robust supply chain strategy and cyber risk assessment program that prohibits untrusted software vendors and/or cloud service providers. • Require all untrusted suppliers to undergo the cyber risk assessment including all relevant security analysis, audits, testing, etc., and deliver the SBOMs.
2		**Ineffective or Non-Existent Cyber Threat Intelligence Program**	• Due to the reports on the rise in nation state sponsored cyber attacks[76], subscribe to industry threat briefings and maintain awareness of national security threat list and the TTPs used to adjust SCRM and Secure Development programs.
1	**Cybersecurity Operations**	**Undetected or Unknown Malware Operating in the Network**	• In addition to Perimeter Security, Zero Trust Architecture principles should be evaluated and adopted where deemed practical/meaningful. • Well defined network segregation techniques to isolate affected software and thus make it difficult in many cases for malicious software to contact command and control servers as well as limit lateral movement within the system. • Segregate management networks such that these networks do not have direct access to the Internet.

[76] The Rise of Nation State Cyber Attacks and the Threat to Business. https://www.intelligencefusion.co.uk/insights/resources/article/nation-state-cyber-attacks/

			• Run time security techniques should be studied to assess their applicability and potential effectiveness in telecommunication environments: for example (RASP) capabilities used in enterprise environments may potentially be used to detect and mitigate these attacks. • Use of Hardware Root of Trust (HRoT) based attestation of software can ensure the integrity of loaded software.
3		**Compromised Cloud Container Software Images**	• Adopt a run time security solution that could potentially detect, mitigate, and/or isolate security anomalies and threats inside the cloud software (e.g., CWPP).
3		**Lack of Account Take Over (ATO) Detection**	• Implement ATO detection on high-value assets and/or applications.
2		**Inadequate Software and Software Components Inventories/Tracking**	• Where possible, all software in use to include (standardized when available) SBOMs, automate SBOM consumption, execute vulnerability scan, and inventory all of the SBOM components – reference NIST SP 800-40r4, p. 10.
1		**Vulnerable Unpatched Software Operating in the Network** (Including embedded FOSS)	• Monitor and track all end-of-life announcements for all vendor provided software. • Complete a security risk assessment for any software that is approaching or reached end-of-life to determine risk of continuing to operate unpatched software in context of use. • Vendors should develop proper communications in their SDLC process to notify consumers of approaching end-of-life dates for any software including FOSS.

			• Vendors should provide mitigation plans for end-of-life software, including FOSS, when possible. • Vendors should regularly evaluate any embedded FOSS software libraries, binaries, etc. for available software updates/patches and develop a plan to patch.
2		**Overprivileged Accounts Compromised**	• Implement secure access capabilities for any employees, contractors, or vendors with access to the service provider's system, including applying access of least privilege.
2	**Standardization Opportunities**	**Uncertainty of SBOM Format, Use, and Deployment**	• Promote the creation of a complete and definitive SBOM standard usable across the industry to address but not limited to: o SBOM Minimum Data Fields o Software Identification Tags o SBOM Automation o Software Provenance Authority o Software Author Identity o Software Supplier Identity o Software Supplier/Author Reputation o Software Chain of Custody o SBOM Immutability
2		**Multiplicity of standards and best practices provides uncertainty as to which should be applied.**	• Encourage collaboration in the industry to coalesce on as few standards and best practices as possible. • Near term, use context specific risk analysis processes to identify appropriate standards and best practices to apply • Additional research is needed to identify how best to manage vulnerabilities: - Vulnerability and Exposure Disclosures - Unpatched Vulnerabilities

6.3.3 Additional Recommendations for the Commission

The application and enforcement of supply chain security varies across small and large providers. In review of today's challenges, small providers are the most challenged given that they have limited personnel and financial resources to devote to supply chain security. Additionally, most security tools are not designed for small providers, whether due to cost or complexity. To add to the challenge, the few tools that have been developed to help small providers do not account for a larger-than-usual small provider network size, while enterprise tools are often unaffordable to these providers. These challenges can lead to small providers outsourcing at least a portion of the company's security needs. Risk and protection, however, can never be entirely outsourced as the provider ultimately must have the ability to act quickly on any supply chain attack.

In addition to the recommendations highlighted in this report, the working group proposes some additional measures.

- The Commission can assist providers of all sizes with software and cloud services supply chain security by offering resources to increase providers' awareness of software and cloud services supply chain risks and describing methods of enhancing software supply chain security. Accordingly, the resources must be capable of being understood and implemented without the need for in-house cybersecurity expertise. Additionally, small providers especially – and in turn the businesses and individuals they serve - would benefit from a three-pronged approach: (1) education, (2) financial assistance, and (3) developer incentives.

  Prioritizing investments in supply chain cybersecurity as a method of identifying which solutions are likely to provide the largest benefit for the cost, consistent with the NIST CSF and the NIST Cybersecurity Supply Chain Risk Management Practices for Systems and Organizations[77] is also important for all service providers, and especially for small providers given their limited financial and staff resources.

- The Commission could engage with CISA and NIST to support an effort to create universal standards and specifications for software supply chain security. These should be applicable regardless of the contextual view of the industry segment. The Commission could foster collaboration amongst the various federal agencies and the industry.

- The Commission could engage with CISA and NIST on the standardization of the SBOM formats, uses, and deployments since these are critical to addressing the software supply chain security vulnerabilities globally. The software and hardware vendors develop and sell products and services to a global community and thus these need to be addressed very broadly in the industry.

- Also, while this work group was not specifically charted to provide a security report on open

---

[77] NIST Special Publication NIST SP 800-161r1 (May 2022). https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-161r1.pdf.

source software, this report does provide several open source security recommendations but the topic itself should be researched independently in a future CSRIC session. Both federal agencies and the industry have published conflicting reports relating to the overall security risks associated with open source software. Most of these have reported that open source software does not pose any more risks than proprietary software. While at the same time, the industry, per Synopsys' 2022 report as stated earlier in this report, identifies that open source is not being patched and managed effectively. The cybersecurity industry acknowledges that software must be patched regularly as part of a mature cyber hygiene program. So, these findings and facts are in conflict and therefore it is recommended that a broader study on open source be done in order to objectively understand the potential risks that open source software poses.

7 Appendix A – Glossary[78]

Term	Description
Bare Metal Server	A physical computer server that is used by one consumer, or tenant, only. Each server offered for rental is a distinct physical piece of hardware that is a functional server on its own. They are not virtual servers running in multiple pieces of shared hardware. In terms of virtualization, a bare metal server makes resources more readily available to one "tenant", network latency is minimized for better performance, and the tenant enjoys root access. Bare metal is highly customizable, and the tenant may optimize the server based upon their individual needs.
Best Practice	A method or technique that users generally accept as superior because it produces results that are superior to those achieved by other methods or techniques.
Cloud Computing	The on-demand availability of computer system resources, especially data storage (cloud storage) and computing power, without direct active management by the user. Large clouds often have functions distributed over multiple locations, each location being a data center. Cloud computing relies on sharing of resources to achieve coherence and typically using a "pay-as-you-go" model which can help in reducing capital expenses but may also lead to unexpected operating expenses for unaware users.
CVSS	Common Vulnerability Scoring System – a free and open industry standard for assessing the severity of computer system security vulnerabilities. CVSS attempts to assign severity scores to

[78] Unless otherwise noted, term descriptions are sourced from Wikipedia.

	vulnerabilities, allowing responders to prioritize responses and resources according to threat. Scores are calculated based on a formula that depends on several metrics that approximate ease and impact of an exploit. Scores range from 0 to 10, with 10 being the most severe. While many utilize only the CVSS Base score for determining severity, temporal and environmental scores also exist, to factor in availability of mitigations and how widespread vulnerable systems are within an organization, respectively.
CWPP	Cloud Workload Protection Platform – a cloud-based software defined network (SDN) with artificial intelligence/machine learning based risk engine to analyze context-based signals and derive a risk score. The context-based signals are output of sensors e.g., OS-level events such as those related to processes, files, network, and memory. When a risk score changes, the artificial intelligence can be configured to stop adversaries and take various actions in real-time as opposed to relying on the traditional revocation of access credentials triggered by security events.[1]
DAST	Dynamic Application Security Testing – a non-functional testing process where one can assess an application using certain techniques and the end result of such testing process covers security weaknesses and vulnerabilities present in an application. This testing process can be carried out either in manual way or by using automated tools.
DevSecOps	Development, Security, and Operations – the integration of security at every phase of the software development lifecycle, from initial design through integration, testing, deployment, and software delivery.[2]
Directory Service	The collection of software and processes that store information about an enterprise, subscribers, or both. An example of a directory service is the Domain Name System (DNS), the hierarchical and decentralized naming system used to identify computers reachable through the Internet or other Internet Protocol (IP) networks.
DNS Server	A computer that stores the mappings of computer host names and other forms of domain name to IP addresses. A DNS client sends questions to a DNS server about these mappings (e.g., what is the IP address of test.example.com?). The mapping of host names enables users to locate computers on a network, using host names rather than complex numerical IP addresses. Whereas the DNS server stores only two types of information - names and IP addresses - an LDAP (see below) directory service can store information on many other kinds of real-world and conceptual objects.

FOSS	Free and Open Source Software – computer software that is released under a license in which the copyright holder grants users the rights to use, study, change, and distribute the software and its source code to anyone and for any purpose. Open source software may be developed in a collaborative public manner.
Hardware Root of Trust	The foundation on which all secure operations of a computing system depend. It contains the keys used for cryptographic functions and enables a secure boot process. It is inherently trusted, and therefore must be secure by design. The most secure implementation of a root of trust is in hardware making it immune from malware attacks. As such, it can be a stand-alone security module or implemented as security module within a processor or system on chip (SoC).[3]
Hyperscale computing	The ability of a computer architecture to scale appropriately as increased demand is added to the system. This typically involves the ability to seamlessly provide and add compute, memory, networking, and storage resources to a given node or set of nodes that make up a larger computing, distributed computing, or grid computing environment. Hyperscale is necessary to build a robust and scalable distributed system.
JDNI	Java Naming and Directory Interface – a Java application programming interface for directory services that allows Java software clients to discover and look up data and resources (in the form of Java objects) via a name.
LDAP	Lightweight Directory Access Protocol – an open, vendor-neutral, industry standard application protocol for accessing and maintaining distributed directory information services over an Internet Protocol (IP) network. Directory services play an important role in developing intranet and Internet applications by allowing the sharing of information about users, systems, networks, services, and applications throughout the network.
MSP	Managed Service Provider – a third-party company that remotely manages a customer's information technology (IT) infrastructure and end-user systems. Small and medium-sized businesses (SMBs), nonprofits and government agencies hire MSPs to perform a defined set of day-to-day management services. These services may include network and infrastructure management, security, and monitoring.[4]
NESAS	Network Equipment Security Assurance Scheme – a universal and global security assurance framework designed to raise confidence and trust in mobile network equipment. The purpose of the scheme

	is to audit and test network equipment vendors, and their products, against a security baseline so they can demonstrate to network operators that they are conforming to the desired standard. The scheme has been defined by industry experts through GSMA and 3GPP.[5]
Proprietary Software	Computer software for which the software's publisher or another person reserves some licensing rights to use, modify, share modifications, or share the software, restricting user freedom with the software they lease. It is the opposite of open source or free software. Non-free software sometimes includes patent rights.
RASP	Runtime Application Self-Protection – a security technology that uses runtime instrumentation to detect and block computer attacks by taking advantage of information from inside the running software.
RCE	Remote Code Execution – the process by which an agent can exploit a network vulnerability to run arbitrary code on a targeted machine or system. For example, in an RCE attack, hackers exploit a remote code execution vulnerability to run malware. RCE can prompt the targeted device to perform code execution, running their own programming in its place, and thus enabling the hacker to gain full access, steal data, carry out a full distributed denial of service (DDoS) attack, destroy files and infrastructure, or engage in illegal activity.[6]
SaaS	Software as a Service – a software licensing and delivery model in which software is licensed on a subscription basis and is centrally hosted. SaaS is also known as "on-demand software" and Web-based/Web-hosted software.
SAST	Static Application Security Testing – a method to secure software by reviewing the source code of the software to identify sources of vulnerabilities.
SBOM	Software Bill of Materials – a list of all open source and third-party components present in a codebase. An SBOM may lists the licenses that govern those components, the versions of the components used in the codebase, and their patch status, which allows security teams to quickly identify any associated security or license risks.[7]
SCA	Software Composition Analysis – the use of one or more tools for scanning a codebase to identify what code (e.g., closed source software, free and open source software, libraries, and packages) is included. These tools may also check for reported vulnerabilities

	pertaining to the code included.[8]
SDLC	Software Development Lifecycle – the complete process of developing a software solution with different stages and steps to bring the software from ideation to building, deployment, and maintenance.[9]
SLSA	Supply-chain Levels for Software Artifacts – a set of incrementally adoptable security guidelines, established by industry consensus, focused on supply chain integrity, with a secondary focus on availability. SLSA standards act as guiding principles, for both software producers and consumers: producers can follow the guidelines to make their software more secure, and consumers can make decisions based on a software package's security posture.
Small Provider	For purposes of this report, small providers are defined as those with 250,000 or fewer broadband subscribers. This definition is consistent with prior Commission action to adopt tailored approaches for small entities.[10]
SSSC	Secure Software Supply Chain – a set of practices that enable organizations to adjust the way they securely consume proprietary or open source packages – both first- and third-party – from source code to operationalize at a sustained high speed and quality relative to their accepted risk level.
SSVC	Stakeholder Specific Vulnerability Categorization – a method to guide organizations through the process of developing a decision tree appropriate for that organization's business model and risk tolerances. It provides a repeatable, transparent process for evaluating a vulnerability, prioritizing, or deprioritizing patching, and explaining the decision to both internal and external stakeholders. Example inputs include the CVSS score, context in which the vulnerable software is used, complexity of the attack, and availability of publicly exploitable code. Example outputs include decisions such as emergency patching, patching within normal development cycles, and declining to patch. Another such approach is provided by VEX.[11]
VEX	Vulnerability-Exploitability eXchange – a tool complementary to an SBOM to provide users (e.g., operators, developers, and service providers) additional information on whether a product is impacted by a specific vulnerability in an included component and, if affected, whether there are actions recommended to remediate it.[12]
Virtualization	Emulation of a physical computer system.

Zero Trust	A security model, also known as zero trust architecture (ZNA), zero trust network architecture or zero trust network access (ZTNA), and sometimes known as perimeterless security, describes an approach to the design and implementation of IT systems. The main concept behind the zero trust security model is "never trust, always verify," which means that devices should not be trusted by default, even if they are connected to a permissioned network such as a corporate LAN and even if they were previously verified.

[1] Gartner, What are Cloud Workload Protection Platforms?, https://www.gartner.com/reviews/market/cloud-workload-protection-platforms.
[2] https://www.ibm.com/cloud/learn/devsecops.
[3] https://www.rambus.com/blogs/hardware-root-of-trust/.
[4] https://www.techtarget.com/searchitchannel/definition/managed-service-provider.
[5] https://www.gsma.com/security/network-equipment-security-assurance-scheme/.
[6] https://www.n-able.com/blog/remote-code-execution; accessed July 8, 2022.
[7] https://www.synopsys.com/blogs/software-security/software-bill-of-materials-bom/
[8] (https://nvlpubs.nist.gov/nistpubs/ir/2021/NIST.IR.8397.pdf).
[9] https://geekflare.com/software-development-life-cycle-sdlc-guide/.
[10] *See* Small Business Exemption from Open Internet Enhanced Transparency Requirements, GN Docket No. 14-28, Order, FCC 17-17 (rel. Mar. 2, 2017).
[11] https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=636379.
[12] https://www.ntia.gov/files/ntia/publications/vex_one-page_summary.pdf.