Security Transparency Assurance Technology for Analysis and Visualization of Software Components

Takayuki Uehara, Yo Kanemoto, and Hiroto Nomura

Abstract

Cyberattacks targeting the software supply chain—the process of developing, providing, using, and updating software—has been increasing. This article discusses trends in supply chain security risks and research and development of security transparency assurance technology to identify information on software components to address such risks.

Keywords: security, supply chain, SBOM

1. Introduction

As economic activities and society evolve, the risks to devices and software are also changing. The most notable change is the increased security risk to the software supply chain. In the United States, an executive order was issued to address this risk, and it is necessary to understand the software component using a software bill of materials (SBOM). In this article, we discuss trends in supply chain security risks and the efforts at NTT laboratories to mitigate such risks.

The 2020 White Paper on Information and Communications in Japan reveals that information and communication technology (ICT) has reached the stage of "social and economic infrastructure." In the past, there were high expectations for ICT to enhance productivity and efficiency. We have transitioned from a long-term development style to a continuous integration approach, which involves releasing functions in rapid succession. To support this accelerated development, we use open source software (OSS) libraries for logging and web rendering as well as for databases and frameworks. The role of software has also become increasingly fragmented and complex, including tasks such as updates and plug-ins (**Fig. 1**).

2. Risk of software supply chain

The process of developing, providing, using, and updating software is known as the software supply chain. This supply chain has become a new target for cyberattacks. For instance, a vulnerability may be discovered in a software library, and an attack exploiting this vulnerability can result in an information leak. Malware may also be injected to a software component through a file updater. It is worth noting that the supply chain, which is responsible for maintaining the safety of the main software body, may paradoxically become a threat that causes significant incidents.

In response to numerous security incidents affecting the United States, the "Executive Order on Improving the Nation's Cybersecurity" was issued in May 2021. As a result, the US NTIA (National Telecommunications and Information Administration) mandates that vendors disclose the SBOM to purchasers.

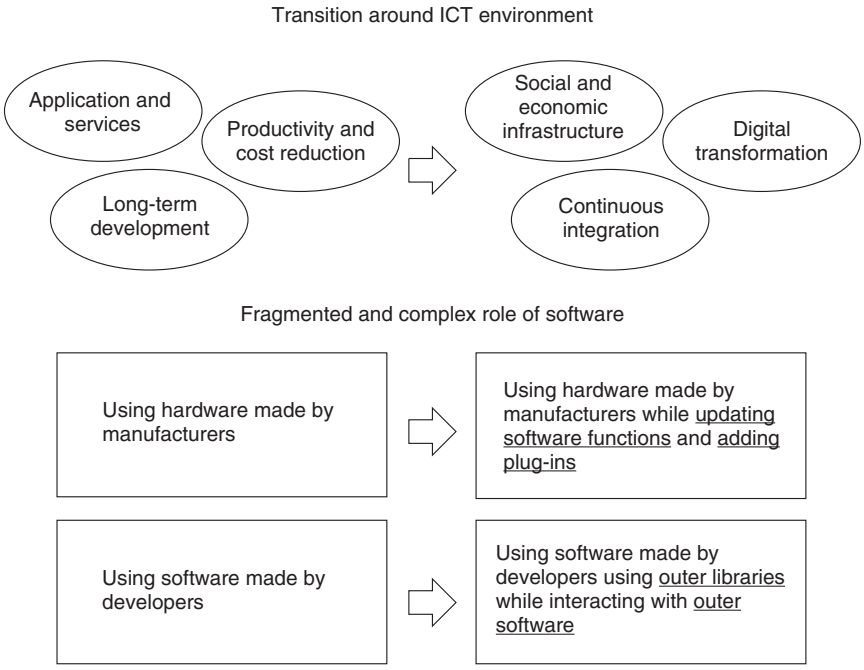


Fig. 1. ICT transition and role of software.

3. Risk countermeasure as key concept of transparency

An SBOM is an inventory of the software included in devices and systems, and data formats such as SPDX (Software Package Data Exchange) and CycloneDX have been proposed. By identifying software components, it becomes possible to promptly investigate the impact of a discovered vulnerability. Additionally, by generating SBOMs at each phase between procurement and use, it is possible to identify instances of unauthorized software infiltration.

In this manner, SBOM-induced transparency is a highly effective risk countermeasure, but there are several issues that must be addressed to further enhance its effectiveness, including:

(1) Software dependencies
(2) Undisclosed software
(3) Managing and using massive amounts of information

We introduce the following efforts of NTT laboratories to address these issues.

4. Software dependencies

Although an SBOM can list the software included in devices and systems, in many cases, only a portion of the software components can be clearly identified. Many pieces of software have dependencies, which reuse code libraries or packages. These dependencies are managed by a system called a package manager. Therefore, it is possible to determine which other software the software depends on from the management information of the package manager. For example, Python has a mechanism called Package Installer for Python (PIP), which manages information about packages to be used. By providing PIP management information from the developer, it becomes clear which software certain software depends on. Thus, while there are explicit dependencies indicated by management information, there are also implicit dependencies (**Fig. 2**).

The most well-known implicit dependency is code cloning. Code cloning refers to matching or similar software source code. A code clone is created by referencing the source code of other software to implement similar functions and copying the source code. In other words, software users may be using the same code as other software without realizing it. Similarly, some code examples are displayed on code question and answer (QA) sites, and some software may use them as is. In other words, it is possible to consider the software code to depend on the code examples on the QA site.

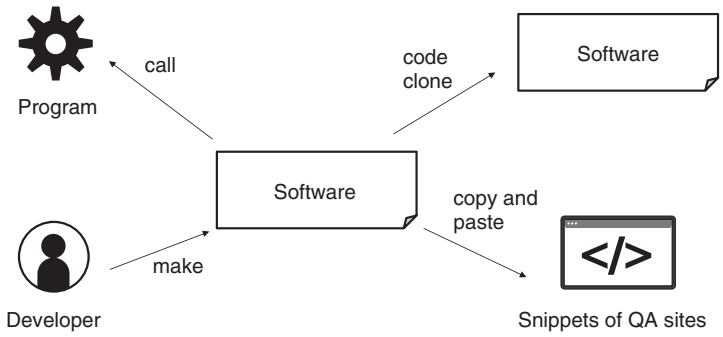


Fig. 2. Various implicit dependencies of software.

Hypothetically, what would happen if a vulnerability (security flaw) is found in the code of the software that the software depends on? Vulnerabilities may exist in all software that has code similar to the software in question. If the software is launched by another program, that program may also be impacted by the vulnerability, potentially affecting the operation of the system on which the software is running.

There may be dependency risks of software developers and their development environments. For an OSS, there are an indeterminate number of developers, and there may be malicious developers or developers among them who are using a development environment that has been compromised without realizing it. As a result, using software created by other developers without sufficient confirmation can pose a security risk.

In this manner, software may have implicit dependencies that are not depicted in the explicit dependencies depicted using an SBOM. We are engaged in research and development of technology to enhance the transparency of software and devices using software by understanding such implicit dependencies.

5. Undisclosed software

Disclosing software components may have drawbacks such as providing hints to attackers. As a result, some developers may want to conceal some or all of the component information. How can transparency be ensured in devices and systems even in such cases?

When a user purchases a device or system, they first verify that it operates as described in the catalog or manual. While using the device, they may detect communication that is not documented in the manual. For instance, some network devices obtain the latest versions of software and data via the Internet. In many cases, this communication is not documented in the manual but a specification for the device.

Therefore, we are conducting research and development focused on the potential for estimating the software components that contribute to operation by using such specifications and data observed from the outside of the device or system. For example, by using logs of communications and operations, it is possible to supplement the component information visualized using an SBOM. Our goal is to create a scenario in which the effectiveness of security risk countermeasures is maximized.

6. Managing and using massive amounts of information

Thus far, we have discussed efforts to increase the quantity of information related to software component information for addressing supply chain security risks. While the diversification of and increase in information increases the likelihood of its utilization for security measures, it may also make it challenging to identify the necessary information and use it appropriately. Additionally, the quantity of information related to component information (e.g., vulnerability information) will grow, and the amount of information to be managed and addressed will become enormous.

We are also researching the state of new security countermeasure work (security operation) and the technology to implement it after achieving security transparency through the enrichment of component information.

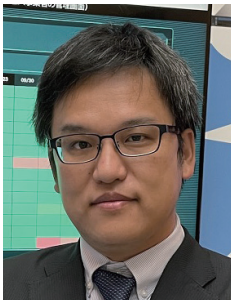
In the Cyber Security Framework (CSF) defined by the NIST (National Institute of Standards and Technology) in the United States, security measures are

divided into five functions: identify, protect, detect, respond, and recover. Enriching component information to achieve security transparency will contribute to strengthening "identify." Therefore, by enhancing "identify," we are researching integrated management and utilization technology for visualization data that efficiently enhances the effectiveness of the other four functions in CSF (efficient vulnerability management using component information, rapid response, highly accurate anomaly detection/cause estimation, automatic countermeasures, etc.).

7. Conclusion

We introduced the research and development of supply chain security risk management based on the key concept of security transparency technology. This technology is an essential security technology for IOWN (the Innovative Optical and Wireless Network), where various players openly co-create, and we aim to support the core infrastructure of society and the economy.

Takayuki Uehara
Senior Research Engineer, Social Innovation Project, NTT Social Informatics Laboratories.
He received an M.E. in science and technology from Chiba University in 1999. Since joining NTT EAST the same year, he has been engaged in research and development on IP telephony and cybersecurity. His research interests include network security and software security.



Hiroto Nomura
Chief Research Engineer, Social Innovation Project, NTT Social Informatics Laboratories.
He received an M.E. in information science from Nara Institute of Science and Technology in 2008. Since joining NTT the same year, he has been engaged in research and development on cybersecurity. His research interests include network security, operational technology security, and software security.

Yo Kanemoto
Chief Research Engineer, Social Innovation Project, NTT Social Informatics Laboratories.
He received an M.E. in information science from Nagoya University in 2013 and Ph.D. in informatics from Kyoto University in 2020. Since joining NTT in 2013, he has been engaged in research and development on cybersecurity. His research interests include network security and software security.