SOAR

STATE-OF-THE-ART REPORT (SOAR)
JANUARY 2024

APPLICATIONS OF ARTIFICIAL INTELLIGENCE

CSIAC-[illegible]-2023-499

(AI) FOR PROTECTING SOFTWARE SUPPLY CHAINS

(SSCS) IN THE DEFENSE

INDUSTRIAL BASE (DIB)

By Abdul Rahman

Contract Number: FA8075-21-D-0001

Published By: CSIAC

This Page Intentionally Left Blank

SOAR

STATE-OF-THE-ART REPORT (SOAR)

JANUARY 2024

APPLICATIONS OF ARTIFICIAL INTELLIGENCE (AI) FOR PROTECTING SOFTWARE SUPPLY CHAINS (SSCS) IN THE DEFENSE INDUSTRIAL BASE (DIB)

ABDUL RAHMAN

ABOUT CSIAC

The Cybersecurity & Information Systems Information Analysis Center (CSIAC) is a U.S. Department of Defense (DoD) IAC sponsored by the Defense Technical Information Center (DTIC). CSIAC is operated by SURVICE Engineering Company under contract FA8075-21-D-0001 and is one of the three next-generation IACs transforming the DoD IAC program: CSIAC, Defense Systems Information Analysis Center (DSIAC), and Homeland Defense & Security Information Analysis Center (HDIAC).

CSIAC serves as the U.S. national clearinghouse for worldwide scientific and technical information in four technical focus areas: cybersecurity; knowledge management and information sharing; modeling and simulation; and software data and analysis. As such, CSIAC collects, analyzes, synthesizes, and disseminates related technical information and data for each of these focus areas. These efforts facilitate a collaboration between scientists and engineers in the cybersecurity and information systems community while promoting improved productivity by fully leveraging this same community's respective knowledge base. CSIAC also uses information obtained to generate scientific and technical products, including databases, technology assessments, training materials, and various technical reports.

State-of-the-art reports (SOARs)—one of CSIAC's information products—provide in-depth analysis of current technologies, evaluate and synthesize the latest technical information available, and provide a comprehensive assessment of technologies related to CSIAC's technical focus areas. Specific topic areas are established from collaboration with the greater cybersecurity and information systems community and vetted with DTIC to ensure the value-added contributions to Warfighter needs.

CSIAC's mailing address:

CSIAC
4695 Millennium Drive
Belcamp, MD 21017-1505
Telephone: (443) 360-4600

REPORT DOCUMENTATION PAGE		*Form Approved* *OMB No. 0704-0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions

for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE January 2024	**2. REPORT TYPE** State-of-the-Art Report	**3. DATES COVERED**
4. TITLE AND SUBTITLE Applications of Artificial Intelligence (AI) for Protecting Software Supply Chains (SSCs) in the Defense Industrial Base (DIB)		**5a. CONTRACT NUMBER** FA8075-21-D-0001
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Abdul Rahman		**5d. PROJECT NUMBER**
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Cybersecurity & Information Systems Information Analysis Center (CSIAC) SURVICE Engineering Company 4695 Millennium Drive Belcamp, MD 21017-1505		**8. PERFORMING ORGANIZATION REPORT NUMBER** CSIAC-BCO-2023-499
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Defense Technical Information Center (DTIC) 8725 John J. Kingman Road Fort Belvoir, VA 22060		**10. SPONSOR/MONITOR'S ACRONYM(S)** DTIC
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.		
13. SUPPLEMENTARY NOTES		

14. ABSTRACT

The application of artificial intelligence (AI) to software supply chains (SSCs) within the defense industrial base (DIB) holds promise to improve cybersecurity posture, ensure stricter compliance with National Institute of Standards and Technology (NIST) controls, and increase user confidence in software built in part upon modules and libraries from outside repositories. AI can provide analysts with suggested frequencies for (re)scanning, supplement threat assessments of infrastructure, automate threat intelligence processing, and expedite cybersecurity risk management. Moreover, the security of SSCs in the DIB can benefit from similar uses of AI as a

recommendation engine for communicating the probability of compromise. For U.S. Department of Defense cybersecurity analysts,

AI-driven automation can provide insight into how closely software capabilities deployed on military and government networks adhere to NIST compliance standards. The ability to reflect the most up-to-date set of vulnerabilities within a system security plan could significantly improve upon the existing practice of relying on manual internal scanning. AI can enable human-in-the-loop workflows to optimize the integration of processed threat intelligence and better identify vulnerabilities per software and/or operating system. This report presents and discusses how AI can protect SSCs purpose-built for the DIB ecosystem.

15. SUBJECT TERMS

cybersecurity, cyberattack, software supply chain (SSC), code repositories, software vulnerabilities, cybersecurity framework, software bill of materials, artificial intelligence, machine learning, automation, penetration monitoring, defense industrial base, contractor software, software build security, third-party vendor security

16. SECURITY CLASSIFICATION OF: U			**17. LIMITATION OF ABSTRACT** UU	**18. NUMBER OF PAGES** 48	**19a. NAME OF RESPONSIBLE PERSON** Vincent "Ted" Welsh
a. REPORT UNCLASSIFIED	**b. ABSTRACT** UNCLASSIFIED	**c.THIS PAGE** UNCLASSIFIED			**19b.TELEPHONE NUMBER** *(include area code)* 443-360-4600

Standard Form 298 (Rev. 8/98)

Prescribed by ANSIStd. Z39.18

ON THE COVER:

(Source: Shutterstock & freepik)

THE AUTHOR

ABDUL RAHMAN, PH.D.

Dr. Abdul Rahman is a subject matter expert in the design and implementation of cloud analytics and architectures that support situational awareness tools for cybernetwork operations for commercial and government customers. He has over 25 years of information technology experience, including software development, network engineering, systems design, systems architecture, security, and network management. He has published widely on topics in physics, mathematics, and information technology. Dr. Rahman holds Doctor of Philosophy degrees in mathematics and physics.

ABSTRACT

The application of artificial intelligence (AI) to software supply chains (SSCs) within the defense industrial base (DIB) holds promise to improve cybersecurity posture, ensure stricter compliance with National Institute of Standards and Technology (NIST) controls, and increase user confidence in software built in part upon modules and libraries from outside repositories. AI can provide analysts with suggested frequencies for (re)scanning, supplement threat assessments of infrastructure, automate threat intelligence processing, and expedite cybersecurity risk management. Moreover, the security of SSCs in the DIB can benefit from similar uses of AI as a recommendation engine for communicating the probability of compromise. For U.S. Department of Defense cybersecurity analysts, AI-driven automation can provide insight into how closely software capabilities deployed on military and government networks adhere to NIST compliance standards. The ability to reflect the most up-to-date set of vulnerabilities within a system security plan could significantly improve upon the existing

practice of relying on manual internal scanning. AI can enable human-in-the-loop workflows to optimize the integration of processed threat intelligence and better identify vulnerabilities per software and/or operating system. This report presents and discusses how AI can protect SSCs purpose-built for the DIB ecosystem.

ACKNOWLEDGMENTS

The author would like to thank the staff of the Cybersecurity & Information Systems Information Analysis Center and SURVICE Engineering Company for their guidance and review of this report.

EXECUTIVE SUMMARY

Managing the intricate and diverse supply chain within the U.S. government involves a heavy reliance on an extensive and varied network of suppliers and vendors for software components. This dependence introduces a range of challenges in ensuring the security of these software components. To address these software supply chain (SSC) security challenges effectively, a combination of technical solutions, robust security practices, collaboration among stakeholders, and adherence to industry standards is essential.

Prioritizing SSC security is critical for organizations to mitigate risks and safeguard against potential vulnerabilities and attacks. Unfortunately, federal entities often lack complete visibility into their SSCs, including information about the origin, integrity, and security of both packet and precursor components. This lack of visibility makes it challenging to identify and mitigate risks and vulnerabilities. Furthermore, reliance on third-party vendors introduces additional risks related to the security practices and integrity of provided software components.

To secure SSCs, it is crucial to implement preventive strategies against attacks. This can be achieved by establishing a security baseline and engaging in robust and continuous behavioral monitoring practices. The most sophisticated of these behavior-based methods involves the utilization of artificial intelligence (AI) models to forecast, infer, predict, correlate, and pinpoint likely weaknesses, potential attack vectors, and avenues of approach within SSC-embedded software.

AI-powered systems can continuously monitor SSCs in realtime, identifying suspicious activities and flagging actions that would otherwise allow for unauthorized access.

AI models are particularly well suited for the automation of routine SSC security audits and assessments that are intended to detect potential vulnerabilities, risks, and security control gaps. Such a proactive, real-time approach enables organizations to address potential exploits and vulnerabilities promptly and, if a penetration does occur, to receive immediate alerts to facilitate swift responses to security incidents, minimizing damage. Moreover, the integration of AI with security coding workflows can streamline the autocompletion and updating of required compliance practices, thereby enhancing overall code quality, defect reduction, and efficiency.

CONTENTS

CONTENTS, *continued*

FIGURES

TABLES

SECTION 01 INTRODUCTION

Once used by the U.S. military in only its most high-tech systems, software is now omnipresent across the defense establishment. As the Defense Innovation Board noted in 2019,software drives “almost everything”that the U.S. Department of Defense (DoD)“operates and uses,”from discrete weapons systems to the overarching networks that provide command, control, and communications capabilities for commanders [1]. While protecting DoD systems from traditional cyberbased attacks will remain an enduring challenge, threats to the security of the software supply chains (SSCs) that develop and produce critical products have recently risen in prominence as a preferred threat vector for penetrating and compromising information systems. By one estimate, the number of SSC attacks against commercial and public entities in the United States increased by more than 700% between 2019 and 2023 [2]. SSC attacks have become such an acute threat that the real-time tracking of SSC incidents has become a niche subsection of the cybersecurity solutions market [3].

1.1 DEFINING SSC ATTACKS

As its name suggests, an SSC refers both to the process of developing code-based packages across multiple parties and the outcome of chained-development activities into usable software products. SSCs encompass software modules, libraries, registries, and components, as well as all the hardware, operating systems, and cloud services that may be used during the coding and development process. As one leading

software developer Red Hat has pointed out, an SSC is most properly considered to include even the people who write the code [4]. Current software development practices are relatively open, especially when compared with traditional coding methods, which remained in use well into the early 2000s. Instead of single entities developing software—entirely in house and by writing all code from scratch—current practices intentionally draw upon broad software communities. Developers leverage code sourced from external (but interconnected) libraries and modules that may serve different purposes for an application (e.g., encryption, authentication, and networking) [4].

Although this type of community development delivers key efficiencies to software production, it also presents bad actors with a wide range of potential threat vectors. Admitting dependencies through SSC development can introduce exploitable software code that is vulnerable to numerous, and cascading, vulnerabilities into the postbuilt product code baseline (see Figure 1-1).

An SSC attack might seek to exploit open-source or shared tools, or to illicitly access a single developer's proprietary build infrastructures [5]. Whatever the vector, an SSC attack consists of at least two elements: (1) a malign actor compromising at least one supplier within an SSC and (2) that vulnerability then being used to harm other supplier(s) or the final product/customer. While it is possible that an SSC can be penetrated in part due to the actions of an insider, leading defense intelligence authorities like the U.S. National Counterintelligence and

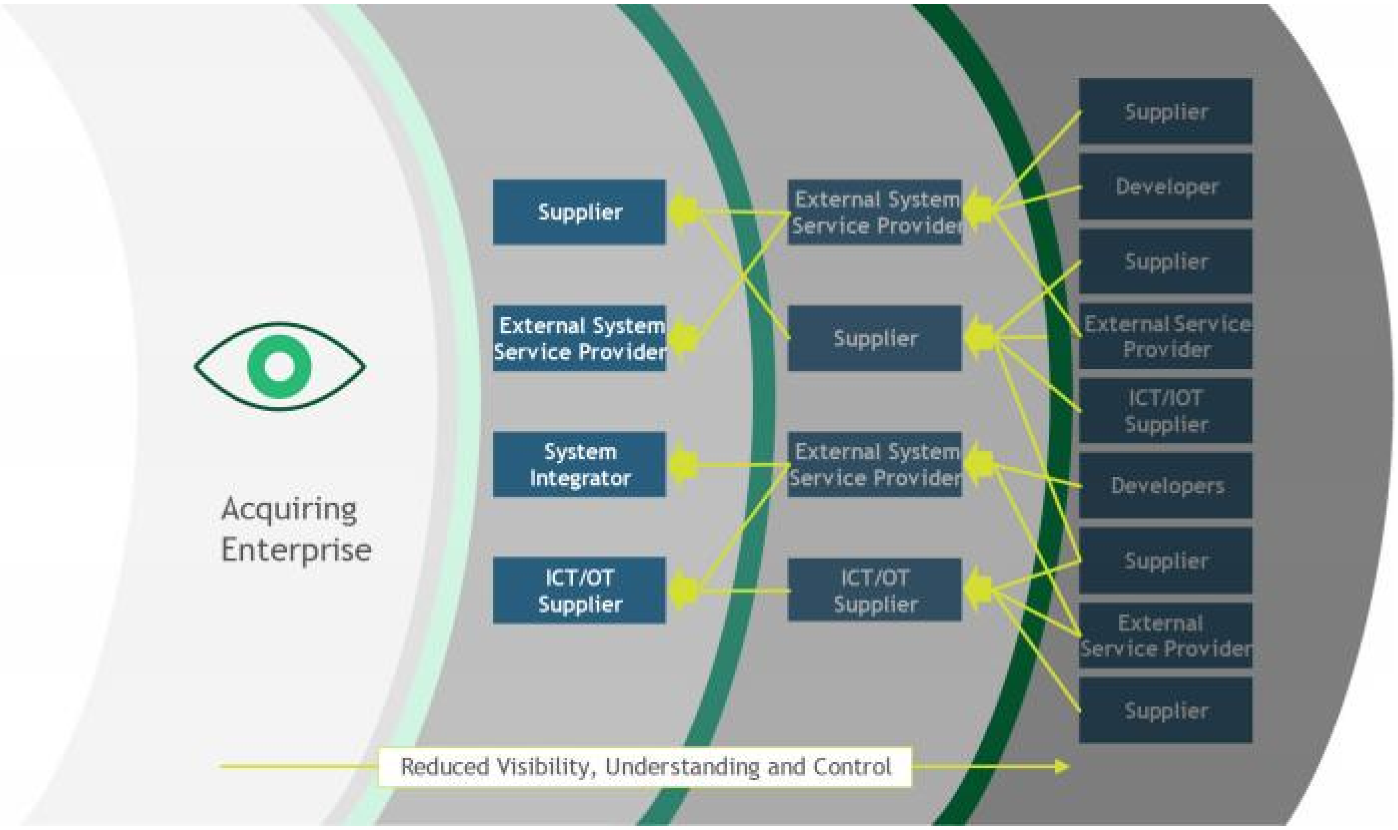

Acquiring Enterprise
Supplier
External System Service Provider
System Integrator
ICT/OT Supplier
External System Service Provider
Supplier
External System Service Provider
ICT/OT Supplier
Supplier
Developer
Supplier
External Service Provider
ICT/IOT Supplier
Developers
Supplier
External Service Provider
Supplier
Reduced Visibility, Understanding and Control

Figure 1-1. An Enterprise's Visibility, Understanding, and Control of Its SSC Decrease With Each Layer of the Broader Development Community's Involvement *(Source: Boyens etal. [6])*.

Security Center see cyberbased (or software enabled) SSC attacks as the more common and, thus, greater threat at present [5].

The documented ability to exploit vulnerabilities in an SSC has existed since at least the 1980s, when the "Ken Thompson hack" or "trusting trust attack" demonstrated the ability to compromise source code while leaving behind almost no trace of alteration [7]. Since then, the massive expansion of software production and the ubiquitous use of connected information systems across all sectors of the economy have made SSC exploits a prime vector for malign actors. For example, SSC attacks often target popular package managers (e.g., node package manager [npm] for Javascript node.js) and their user communities. These communities have experienced incredible growth over the past decade—the number of public repositories hosted in the GitHub platform grew from 46,000

in early 2009 to more than 200 million by 2022 [5]. Accordingly, adversarial nation-states, terrorists, and other transnational criminal organizations recognize that SSC attacks can cause widespread and cascading harmful effects, all while requiring relatively few resources to execute [8].

A number of headline penetrations in recent years have raised the profile of SSC attacks for malign actors. In 2017, the “NotPetya”SSC cyberattack—the most damaging such attack then to date—infected a line of accounting and tax reporting software used by the Ukrainian government before spreading to several large multinational firms. The malware that Russian-sponsored hackers inserted disrupted email systems at a major food manufacturer and disabled multiple logistics systems for an international shipping company. In doing so, NotPetya even crippled one pharmaceutical firm’s ability to supply vaccines to the U.S. Centers for Disease Control and Prevention [9]. By 2020, the "SolarWinds" cyberattack, which originated from the Russian Foreign Intelligence Service, similarly penetrated a wide array of networked systems, primarily within the U.S. federal government. After being injected with backdoor code, a routine software update package for a technology administration suite was widely downloaded; worse, the compromise went undetected for nearly 12 months [10].

1.2 SSCS AND THE DEFENSE INDUSTRIAL BASE

The DoD acquires software products and systems, professional services, and the supporting hardware and computing power needed for operation much in the same way it obtains crates of 5.56-mm rifle ammunition—mostly purchasing them from private firms and other public or nonprofit suppliers. Generally known as the Defense Industrial Base (DIB), this collection of organizations, facilities, and resources provides the DoD with hundreds of billions of dollars of products and services each year and represents the nation's enduring industrial and economic might [11]. The broad magnitude and scope of the DoD's acquisition activities means that more than 1 million workers and around 60,000 firms can be considered part of the DIB [11]. While many of these firms do not directly shape or influence the development of software products that enter militarily-relevant SSCs, every single entity (even those that only produce hardware, like 5.56-mm cartridges) uses software platforms that are vulnerable to penetration.

The DIB's immense scope and wide reach into suppliers and subcontractors make the defense of its SSCs an immense task. Two longstanding vulnerabilities further complicate this challenge:

1. The production of microelectronics, once common in the United States, has been mostly offshored to international producers, limiting government security oversight. (Enactment

of the $54-billion federal "Creating Helpful Incentives to Produce Semiconductors (CHIPS) Act of 2022" is aimed at reversing this trend [12].)

2. "The growing complexity" of the electronics, platforms, and architectures that DIB-produced and DoD-operated systems depend upon makes SSC security an utterly overwhelming task. Both a "lack of traceability" and the need for persistent, "continuous monitoring" by the DoD of vendors and components in the DIB are key limiters in comprehensively securing SSCs within the national security and homeland defense space [13].

Along with the centrality of software to DoD operations, these two vulnerabilities have made penetration of SSCs within or adjacent to the DIB, as well as the intelligence community at large, a key objective for adversarial action [14]. In the past 5 years, military analysts have witnessed an uptick in attempts to penetrate defense-related SSCs, with a particular eye toward gaining direct control over DoD systems and other critical infrastructure to disable them in the event of armed conflict. In September 2019, hackers attacked the SSCs of four subcontractors working for Airbus, a major aeronautics firm that supplies the DoD with sensing systems as well as airframes [15]. In May 2023, a multi-agency joint advisory warned that a hacking group sponsored by the People's Republic of China, known as Volt Typhoon, had penetrated electrical systems in the homeland and in the U.S. territory of Guam—a key strategic site for operations in the U.S. Indo-Pacific Command [16]. Further complicating the daunting task of SSC security is the hodgepodge of systems, software vintages, and architectures that the DoD employs; each service branch largely operates its systems and networks separately from the others. Unifying a software security posture across the department has been likened to "assembling a puzzle with pieces from different sets" [17].

1.3 SECURING SSC

Both the DoD and the broader federal national security enterprise have responded to assess the vulnerability of their systems to SSC exploits and secure the broader software development and production communities that support government operations. For example, in July 2023, new administrative policies promulgated in the U.S. “National Cybersecurity Strategy Implementation Plan”tightened the technical requirements that suppliers and contractors must meet in following cybersecurity supply chain risk management (C-SCRM) best practices [18].

Operating in compliance with these best practices is a critical step to building trust in international software suppliers, as compliance makes the digital ecosystem more“transparent, secure, resilient, and trustworthy” [10].

Two months later, the DoD followed up the whole-of-government strategy with its own DoD-specific cyberstrategy [19]. The document recognizes, at a high strategic level, the importance of protecting the DIB from malicious cyberattacks and recommends a number of procedural changes, like the alignment of DIB contract incentives with DoD-specific cybersecurity requirements. Moreover, the strategy points toward the usefulness of ongoing research and development activities that might increase DoD capabilities for“rapid information-sharing and analysis” in the“identification, protection, detection, response, and recovery of critical DIB elements” [19]. The Office of the DoD Chief Information Officer is also working to finalize an enterprise-wide strategy for cyber supply chain risk management to guide protective actions for SSCs across the DoD [20].

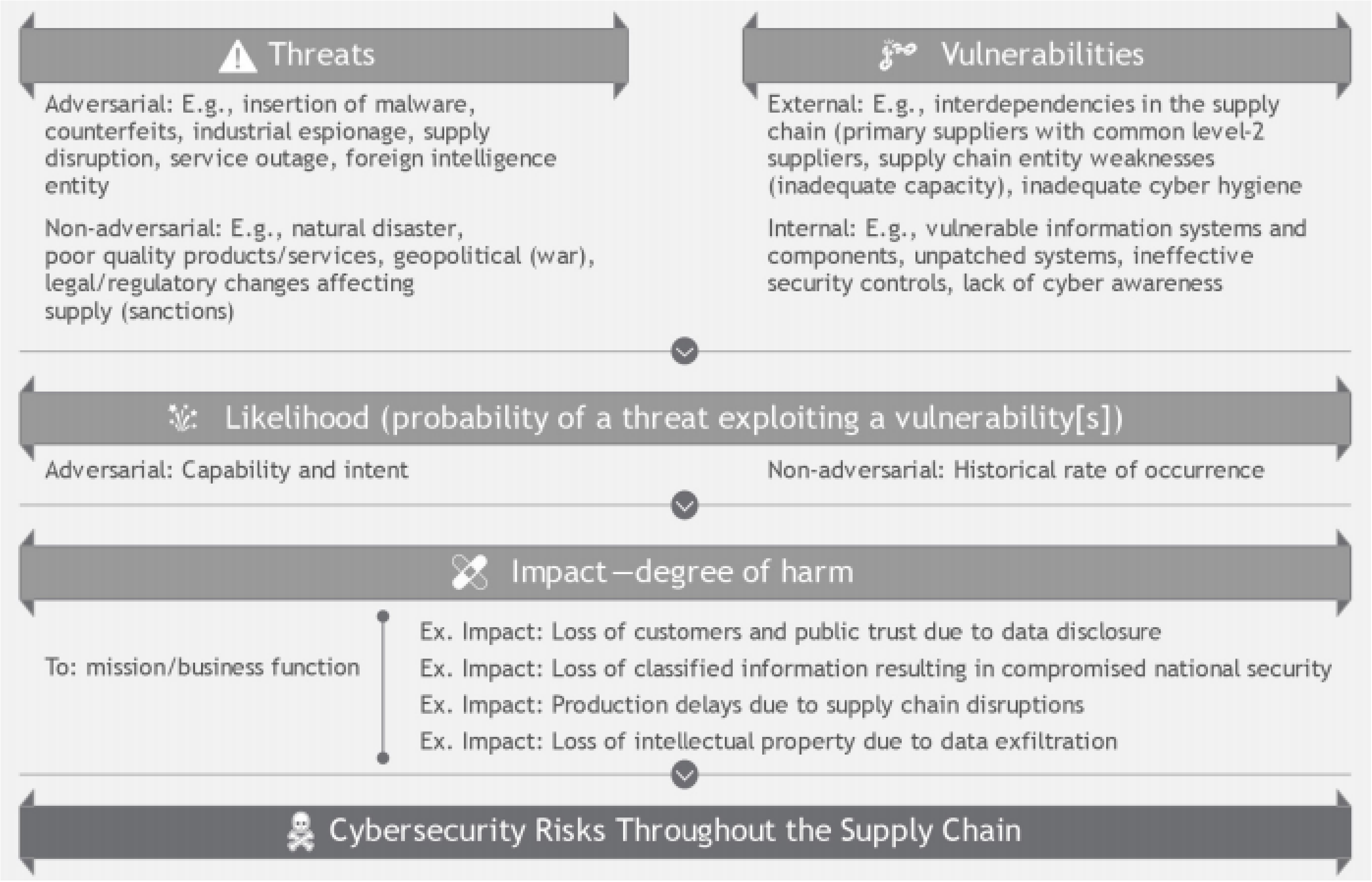
The majority of technical guidance for securing SSCs across the firms and organizations that make up the DIB is generated by the National Institute for Standards and Technology (NIST). A longstanding federal entity originally involved in the standardization of weights, measures,

and metrology measurements, NIST released its landmark cybersecurity framework (CSF) as Version 1.0 in 2014 [21]. The framework quickly found widespread adoption among commercial firms and government information technology (IT) departments and has been updated and expanded several times since [22].

At its core, the CSF details a set of best-practice cybersecurity activities, standardized tools, and references and further describes the"desired outcomes" of the application of the framework across an organization. While NIST is not a traditional regulatory agency, use of the CSF has since become mandatory for federal agencies [23]. Other NIST guidance, including the "Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities" (NIST Special Publication [SP] 800-218) [24] and the "Cybersecurity Supply Chain Risk Management Practices for Systems and Organizations" (NIST SP 800-161r1) [6], provides additional discussion of vulnerabilities and SSC security controls at both a technical and conceptual level (see Figure 1-2).

1.4 REPORT OVERVIEW

While guidance documents for the organizational practice of C-SCRM are very useful, they might also best be characterized as broad and nonspecific [25]. Moreover, as the volume of data and code that inhabit a given SSC continues to grow, entities like firms within the DIB would benefit greatly from next-generation analytical tools to identify potential SSC vulnerabilities and then secure them. Accordingly, this state-of-the-art report discusses the requirements, progress, and latest trends in using artificial intelligence (AI) tools and techniques to secure the defense-critical SSC. Detection of SSC attacks can be accomplished through building AI models deployed against collected distributed datasets designed, developed, trained, and tested over useful features. The combination of AI-enabled analytics with broader security



Threats

Adversarial: E.g., insertion of malware, counterfeits, industrial espionage, supply disruption, service outage, foreign intelligence entity

Non-adversarial: E.g., natural disaster, poor quality products/services, geopolitical (war), legal/regulatory changes affecting supply (sanctions)

Vulnerabilities

External: E.g., interdependencies in the supply chain (primary suppliers with common level-2 suppliers, supply chain entity weaknesses (inadequate capacity), inadequate cyber hygiene

Internal: E.g., vulnerable information systems and components, unpatched systems, ineffective security controls, lack of cyber awareness

Likelihood (probability of a threat exploiting a vulnerability[s])

Adversarial: Capability and intent

Non-adversarial: Historical rate of occurrence

Impact—degree of harm

To: mission/business function

Ex. Impact: Loss of customers and public trust due to data disclosure

Ex. Impact: Loss of classified information resulting in compromised national security

Ex. Impact: Production delays due to supply chain disruptions

Ex. Impact: Loss of intellectual property due to data exfiltration

Cybersecurity Risks Throughout the Supply Chain

Figure 1-2. Cybersecurity Risks Throughout the Supply Chain *(Source: Boyens etal. [6])*.

approaches like the current version of NIST CSF 1.1 [26] (Version 2.0 of the CSF [27] is underdevelopment) can generate a truly comprehensive method of securing SSCs.

This report discusses data management strategies and feature development as the two core prerequisites for robustAI model development. Section 2 summarizes data management strategies to describe the most salient aspects needed for robustAI model development aligned to SSC security. Section 3, in surveying feature engineering and development, addresses the required understanding of SSC frameworks and their attributes upon which AI models will be trained. Section 4 explores how AI models can enhance software code reliability, integrate with blockchain technology, and improve SSC vulnerability analysis and detection. Overall, this

report discusses the performance of AI models across all phases of SSC analytical processing, where it may lead to faster predictions and enhanced integration with security operations workflows.

SECTION 02 DATA MANAGEMENT STRATEGIES

The development of AI-enabled models is predicated first upon the use of robust and best-practice-compliant data management practices. The processes of data collection, aggregation, storage, and organization are key enablers of engineering (or developing) features targeted at the phenomena that will provide the largest benefits to early detection of SSC compromises. For instance, the use of packages from public component registries, if not carefully monitored, can introduce significant vulnerabilities to an SSC. Data provided by Sonatype, an SSC management company, reveal that the count of malicious packages identified across diverse open-source ecosystems in 2023 has tripled compared to the previous year [28]. That increase, in turn, comes on the heels of a staggering 650% year-on-year increase in security attacks exploiting vulnerabilities in open-source software's supply chain in 2021 [29].

2.1 OPEN-SOURCE PACKAGES

This rapid rate of expansion is truly remarkable, emphasizing the supply chain's emergence as one of the fastest-growing avenues for malevolent code execution. The widespread use of open-source packages in particular threatens to introduce vulnerabilities (or compromises) into a singleSSC or multiple-linked, interdependent SSCs, with harmful ramifications that can cascade both upstream and downstream of a penetration [30] (see Figure 2-1). Without greater vision into the full reach of an SSC, benevolent actors are limited in the measures

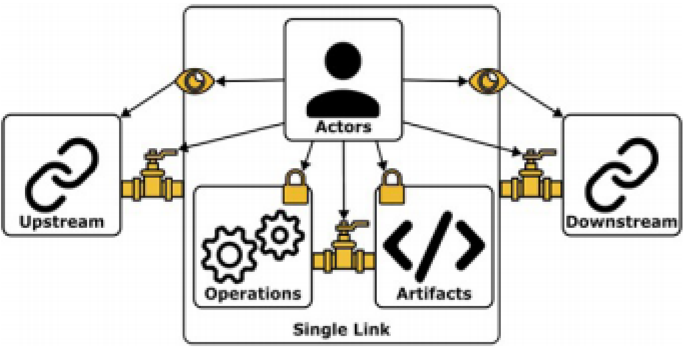


Figure 2-1. An SSC With Focus on a Single Link; Systemwide Security Depends on Upstream/Downstream Transparency, Link Validity, and Logical Separation Between Components and Links *(Source: Okafor et al. [30]).*

available to them to mitigate risk or employ countermeasures in a timely fashion.

Virtually all modern software relies heavily on prior innovations distributed freely and made accessible by the world's most skilled experts. This invaluable foundation is offered to developers at no cost. As a result, it is often estimated that as much as 90% of the code utilized in software production systems is derived from open-source origins. However, a substantial number of open-source programming language repositories are maintained by the open-source community in a voluntary, part-time, and often haphazard manner [28]. While efforts have been made to prevent the hijacking of existing developer accounts for the dissemination of malicious components (such as the introduction of mandatory multifactor authentication), this does not fully deter attacks involving the upload of rogue packages from new accounts.

Few, if any, automated detection techniques are currently in place (much less actively used in practice), and the volunteer-based vulnerability removal procedures used by many community repositories are slow, cumbersome, and grossly inefficient when facing code intentionally designed to be malicious from the outset. Sonatype emphasizes the fact that packages harboring malicious code are often treated similarly to packages with new security vulnerabilities. This practice can allow malicious packages to persist longer than necessary, exposing developers to risks [28].

The potential of generative AI in software development is undeniably promising, but it does come with its set of challenges, both real and perceived. Significantly, a full 61% of the developers polled by Sonatype in 2023 view generic AI technology as“overhyped,”while only 37% of IT security leads feel the same. While a majority of respondents currently utilize AI to varying degrees, that use is not always driven by personal preference. An astonishing 75% of both groups acknowledge feeling pressure from their organization’s leadership to embrace and deploy AI technologies, as leadership typically stresses AI’s productivity-enhancing capabilities over its associated potential security concerns [28]. However, it is likely that applying specific targeted AI models to the task of SSC vulnerability monitoring will minimize this skepticism, as AI moves from a nebulous technological concept to a series of discrete, defined, and useful software tools.

To proactively address the issue of open-source compromises, robustAI models can be implemented to support the prediction of package vulnerabilities that are susceptible to high-risk supply chain attacks. In 2022, Zahanet al. [29] focused on assisting software developers and security experts in assessing signals of weakness in the npm supply chain to prevent future attacks by conducting empirical investigations into npm package

metadata. The authors scrutinized the metadata of 1.63 million packages, applying 6 indicators of compromise (IoC) of SSC security vulnerabilities. These include an expired maintainer domain, installation scripts, unmaintained packages, too many maintainers, too many contributors, and overloaded maintainers [29].

These IoCs can be used both to structure SSC data and formulate feature engineering approaches for AI models equipped to detect SSC attacks.

One of the case studies used by the authors [29] identified more than 10 malicious packages using the installation script indicator. Furthermore, they discovered over 2,800 maintainer email addresses that were associated with expired domains—a vulnerability that could potentially enable an attacker to hijack over 8,000 packages by way of compromising npm accounts. The software development community provided positive feedback for the use of these IoCs as"weak link signals"or indicators. A survey completed by 470 npm package developers found greater than 50% support of responses for the use of 3 of the 6 IoCs: an expired maintainer domain, installation scripts, and unmaintained packages [29].

2.2 ATTACK SURFACE MANAGEMENT AND THREAT MODELING

Software package vulnerabilities are a significant contributor to the overall risk associated with software security. Eliminating all vulnerabilities is both impossible and impractical, as they can potentially lead to security risks in the SSC. Nevertheless, effective strategies exist for reducing and managing these risks. Two of the most effective strategies for managing supply chain security risks are known as"attack surface management" and"threat modeling."

The task of controlling attack surfaces involves assessing and managing the system entry points that attackers could exploit to compromise a system. Doing so helps to identify vulnerabilities in either the system’s design or implementation that might be particularly susceptible to malign action [31]. Threat modeling, on the other hand, is the process of analyzing and understanding the characteristics and scope of potential threats to a system—a key input to which is an assessment of its prime attack surfaces (see Figure 2-2) [31]. Both approaches are valuable to perform throughout the entire software lifecycle, from development and deployment to ongoing maintenance.

From an attack surface perspective, open-source code compromises transpire when malicious actors infiltrate publicly accessible code repositories and insert harmful code for public consumption. Unsuspecting developers—in their understandable search for freely available code snippets to fulfill specific functions—unwittingly incorporate these tainted elements into their third-party code.

One salient example dates back to 2018 and involved the detection of malevolent Python libraries on the official Python Package Index. Employing what is known as“typosquatting” tactics, the attacker fashioned libraries with names like“diango,”“djago,”and“dajngo,” mimicking the common and much sought-after Python library correctly spelled as“django.” To aid in the persistence of their propagation across linked SSCs, these deceptive libraries replicated the genuine code and functionality of their genuine counterpart but harbored additional features, such as the capability to establish boot persistence and create a reverse shell on remote workstations. Notably, open-source code compromises can also affect privately owned or enterprise software, since developers of proprietary code frequently incorporate open-source elements into their products [32]—sometimes even if their organization’s security policy prohibits it.

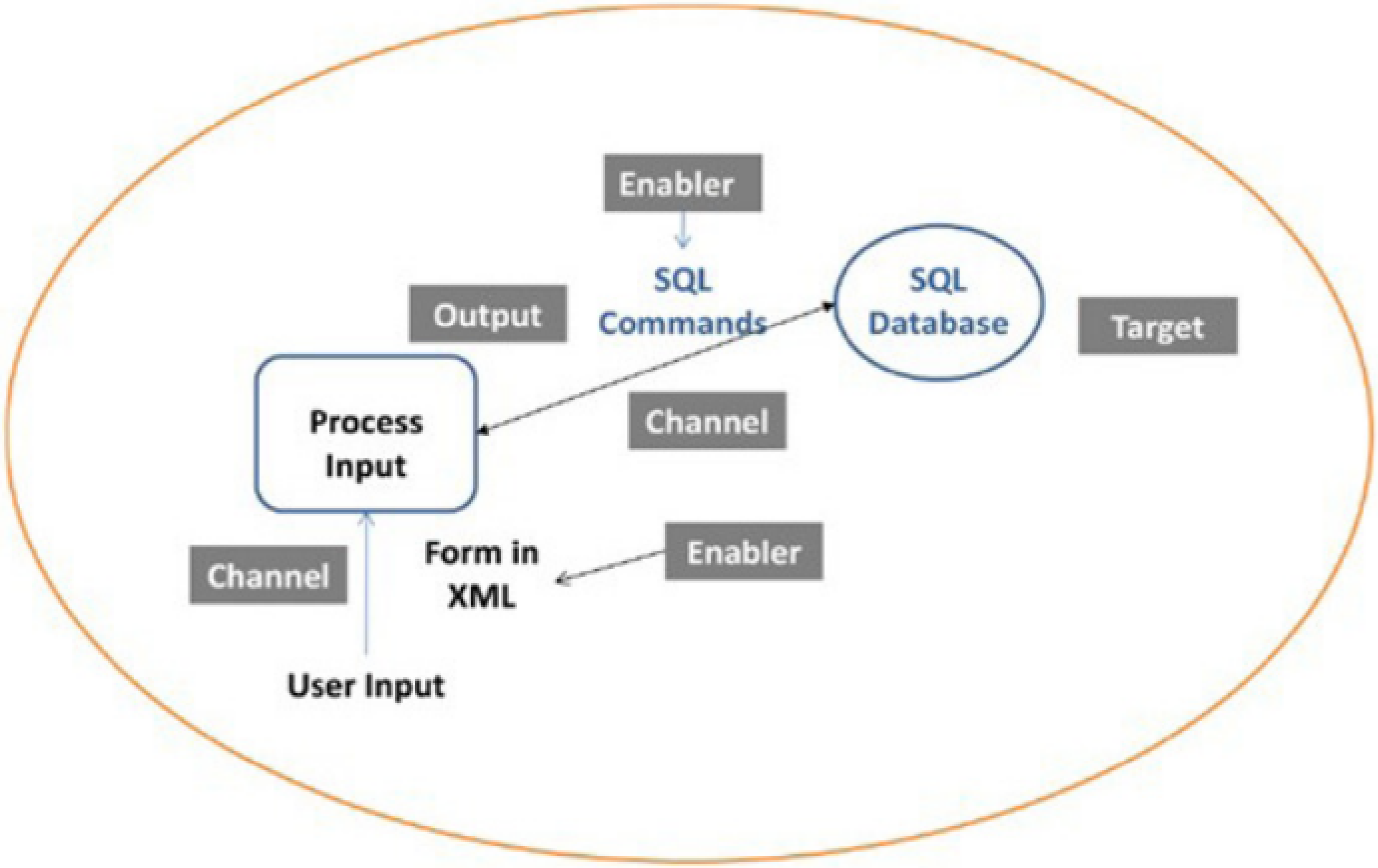


Figure 2-2. Data Flow Diagram of an Example Attack Surface *(Source: Ellisonetal. [31])*.

Because an SSC’s attack surface can admit a diverse and wide range of vulnerabilities, the ramifications of an SSC compromise can be dire. Initially, threat actors seek to exploit the“gaps” in a compromised software vendor to secure privileged and persistent access to a victim’s network. By attacking an outside or third-party software vendor as part of their effort to target another organization, bad actors circumvent outer security measures like border routers and firewalls, thereby gaining an initial foothold. In the case of network access loss, threat actors can often simply re-enter the system through the compromised vendor.

While the process of gaining initial access is generally indiscriminate, threat actors often exercise discretion in selecting targets for subsequent actions. These follow-on actions exhibit considerable variability; however, they frequently commence with the insertion of tailored malware packages into a chosen target. Depending on the threat actor’s intent and capabilities, this added malware may enable the attacker to conduct a variety of malicious activities, to include data or financial theft; surveillance of organizations or individuals; network or system disruption; or, in extreme cases, even physical harm or loss of life.

Those who work to defend friendly networks are limited when attempting to promptly mitigate the repercussions of an SSC compromise. This stems from the fact that organizations seldom have full control over their entire SSC, lacking the authority to compel each participant in the supply chain to swiftly undertake mitigation measures. Recognizing the challenge of mitigating postattack consequences, it is imperative for network defenders to proactively adopt and adhere to industry best practices.

Implementation of these practices can only improve or enhance an organization’s capacity to prevent, mitigate, and respond to such attacks.

Examining the attack surface and following known risk assessment methodologies (like threat modeling) are essential practices for mitigating SSC security risks. Nevertheless, it is crucial to acknowledge that these analyses are not static entities. Attackers have the capability—and are highly motivated—to introduce novel techniques that may infect software or code snippets that had previously been considered secure. Consequently, the assessment of the attack surface and its corresponding threat models should undergo periodic reviews via human-in-the-loop workflows and/or automated processes. The frequency of these reviews should be particularly heightened when dealing with emerging technologies (including the use of AI by third-party code-development processes elsewhere) and could align with the training, testing, and deployment of AI development lifecycles.

By their nature, new technologies may possess undocumented vulnerabilities (e.g., zero-days) because they lack an extensive history of known exploits, which would otherwise be used to inform threat modeling and other security risk assessment techniques. More frequent reviews are thus necessary to adapt to this evolving threat landscape. One such response might increase the frequency of internal system/enterprise scanning to detect abnormal behavior. In this use case, AI models can be developed and trained to specifically alert to such anomalies.

Both the frequent recalibration of the scope of security assessments and the behavior-based AI models can significantly aid the collection of essential information for organizational leaders to prioritize the means for their SSC security. Note that the security risks addressed by threat modeling and attack surface analyses differ significantly from those addressed by more traditional infrastructure security mechanisms, such as firewalls, authentication methods, and access control mechanisms. These infrastructure mechanisms primarily focus on preventing unauthorized access