Supply Chain Security Frameworks

Press Space for next page →

背景

- ☒☒☒☒☒☒☒☒☒☒☒☒☒☒/SBOM関連の話題を見かけることが多い
- 巨大テック企業がサプライチェーン関連のフレームワークの整備を行なっているのを見かける

目的

- ソフトウェアサプライチェーンセキュリティ関連のフレームワークを見てみる
  - その中で言及されている技術を眺める
  - どのフレームワークがどの領域をターゲットにしているのかをなんとなく理解する

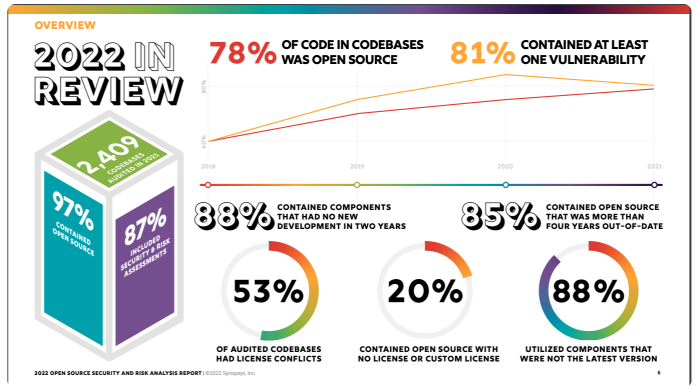
背景

□□□□□□□□□□□□□□□□□□□□

ある調査⊠2,409⊠⊠⊠⊠⊠⊠⊠⊠内容を調⊠⊠⊠...

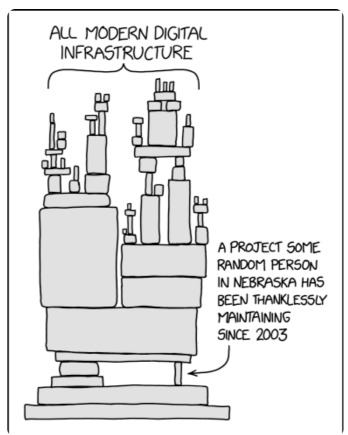
- **97%** ⊠⊠⊠⊠⊠⊠⊠⊠含んでいた
- コードベース内⊠ **78%** ⊠⊠⊠⊠OSSだった

ref: 2022 OPEN SOURCE SECURITY AND RISK ANALYSIS REPORT - Synopsys



セキュリティの強度はその最も弱い部分で決定する。

*“The chain is only as strong as its **weakest link**, for if that fails the chain fails and the object that it has been holding up falls to the ground.” - Thomas Reid (1785)*



refs: 2347: Dependency - explain xkcd

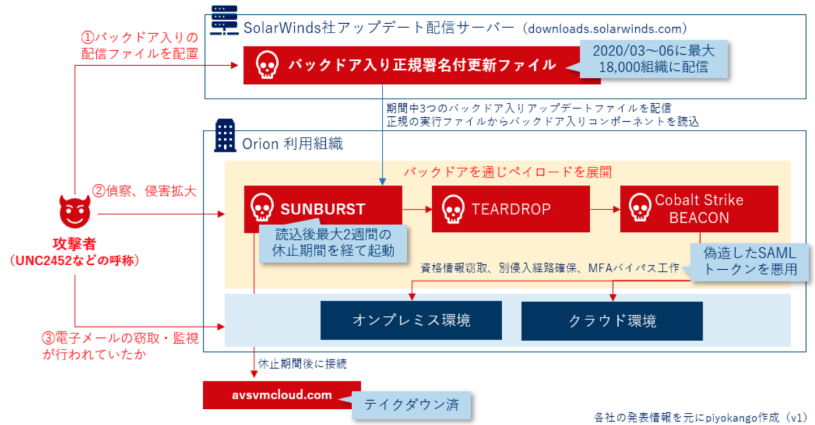
npm install...

背景

□□□□□□□□□□□□□□□□□□□□□□

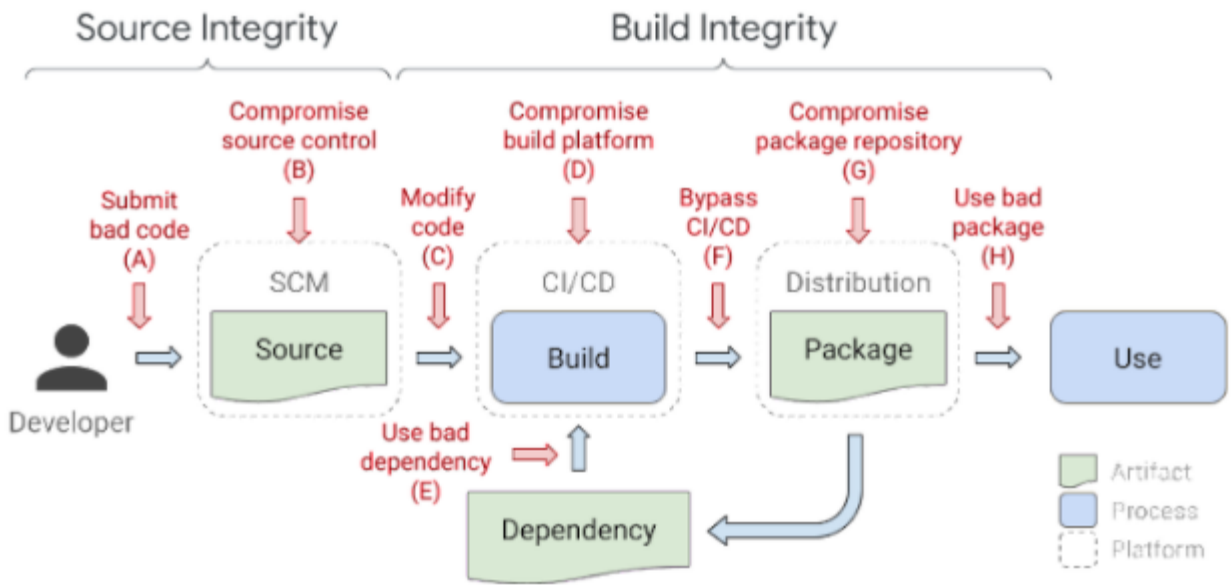
Solar Winds☒ Orion Platform のサプライチェーン攻撃

- `Orion`はネットワーク監視ソフトウェア
- Orionから配布された正規のパッチファイルにバックドア機能を持つDLLが含まれていた
    - ビルドサーバーに忍び込まれ、正規のパッチファイルだと認識されていた
- 2020☒3~6☒の間に最大で**18,000**組織に配信
    - 米国政府機関をはじめ、Microsoft, Nvidia, Intelなど



☒☒: SolarWindsのサプライチェーン攻撃☒☒☒☒☒☒☒☒☒☒ \- piyolog

Supply Chain Attacking Surface



ref: Protect your open source project from supply chain attacks | Google Open Source Blog

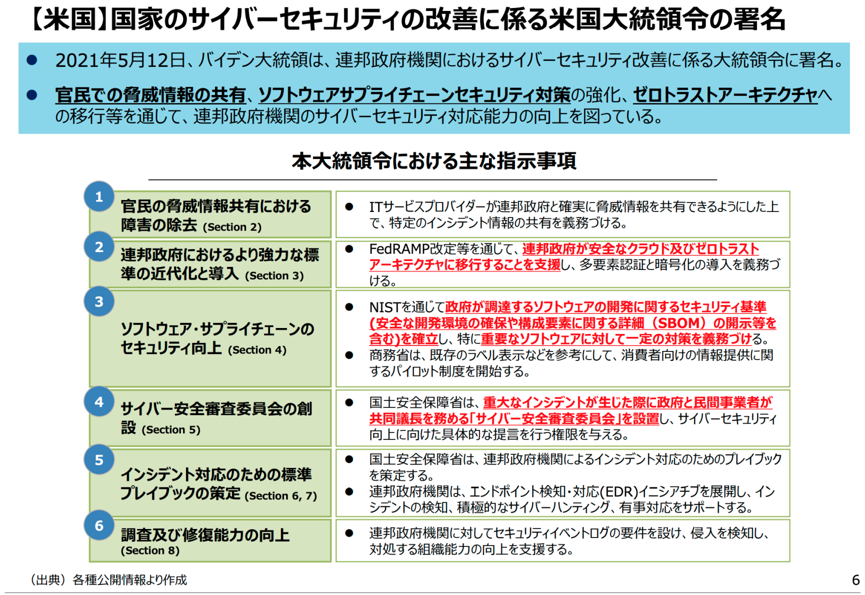
Executive Order 14028

□□□□□□□

官民一体となってサイバーセキュリティを向上するため、注力すべき事項とガイドラインの整備を各連邦政府に指示

※ ☒☒: `SBOM` に関して述☒☒☒☒☒☒2箇所のみ

> Sec. 4. Enhancing Software Supply Chain Security.
>
> …
>
> (e) … (NIST) issue guidance identifying practices that enhance the security of the software supply chain …
>
> __(vii) providing a purchaser a Software Bill of Materials (SBOM) for each product directly or by publishing it on a public website;
>
> …
>
> (f) Within 60 days of the date of this order, … and the Administrator of the National Telecommunications and Information Administration, shall publish minimum elements for an SBOM.



【米国】国家のサイバーセキュリティの改善に係る米国大統領令の署名

- 2021年5月12日、バイデン大統領は、連邦政府機関におけるサイバーセキュリティ改善に係る大統領令に署名。
- **官民での脅威情報の共有**、**ソフトウェアサプライチェーンセキュリティ対策**の強化、**ゼロトラストアーキテクチャ**への移行等を通じて、連邦政府機関のサイバーセキュリティ対応能力の向上を図っている。

本大統領令における主な指示事項

1 官民の脅威情報共有における障害の除去 (Section 2)	● ITサービスプロバイダーが連邦政府と確実に脅威情報を共有できるようにした上で、特定のインシデント情報の共有を義務づける。
2 連邦政府におけるより強力な標準の近代化と導入 (Section 3)	● FedRAMP改定等を通じて、**連邦政府が安全なクラウド及びゼロトラストアーキテクチャに移行することを支援**し、多要素認証と暗号化の導入を義務づける。
3 ソフトウェア・サプライチェーンのセキュリティ向上 (Section 4)	● NISTを通じて**政府が調達するソフトウェアの開発に関するセキュリティ基準(安全な開発環境の確保や構成要素に関する詳細（SBOM）の開示等を含む)を確立**し、特に**重要なソフトウェアに対して一定の対策を義務づけ**る。 ● 商務省は、既存のラベル表示などを参考にして、消費者向けの情報提供に関するパイロット制度を開始する。
4 サイバー安全審査委員会の創設 (Section 5)	● 国土安全保障省は、**重大なインシデントが生じた際に政府と民間事業者が共同議長を務める「サイバー安全審査委員会」を設置**し、サイバーセキュリティ向上に向けた具体的な提言を行う権限を与える。
5 インシデント対応のための標準プレイブックの策定 (Section 6, 7)	● 国土安全保障省は、連邦政府機関によるインシデント対応のためのプレイブックを策定する。 ● 連邦政府機関は、エンドポイント検知・対応(EDR)イニシアチブを展開し、インシデントの検知、積極的なサイバーハンティング、有事対応をサポートする。
6 調査及び修復能力の向上 (Section 8)	● 連邦政府機関に対してセキュリティイベントログの要件を設け、侵入を検知し、対処する組織能力の向上を支援する。

（出典）各種公開情報より作成

☒☒: ☒☒の産業サイバーセキュリティに関する動向☒☒☒☒ \- 経済産業省

Supply Chain Security Frameworks

Supply Chain Security Frameworks

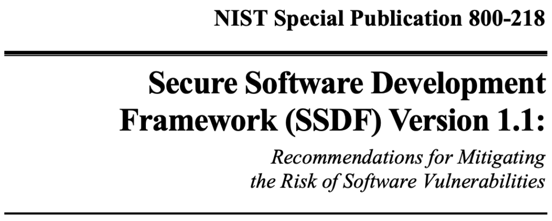
Abbr	Name	Org	Publishd	Descriptions
SSDF	Secure Software Development Framework v1.1	NIST	2022/02/03	ソフトウェアの脆弱性のリスクを軽減するための推奨事項。EO14028を受けて改訂。
SLSA	Supply chain Levels for Software Artifacts	Google	2021/06/21	ソフトウェアアーティファクトの☒☒☒(Integrity)を保証するためのEnd-to-Endのフレームワーク
-	Software Supply Chain Best Practices	CNCF	2021/05/14	セキュアな開発を行うためのプラクティス集
S2C2F	Secure Supply Chain Consumption Framework	OpenSSF (Microsoft)	2022/08/01	OSSのセキュアな利用方法に関するフレームワーク

SSDF (SP 800-218)

NIST - Secure Software Development Framework

☒☒☒/☒☒☒の脆弱性が悪用された場合の潜在的な影響を軽減するための、
Software producerに向けたガイドライン

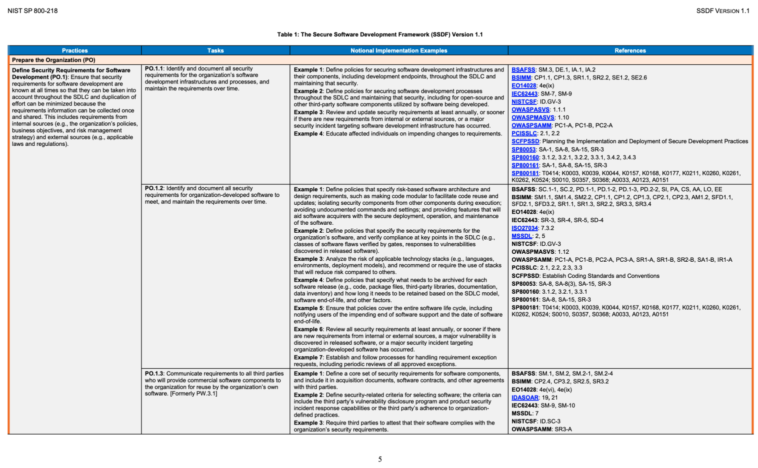
Group	Tasks
Prepare the Organization (PO)	13
Protect the Software (PS)	4
Produce Well-Secured Software (PW)	16
Respond to Vulnerabilities (RV)	9



NIST Special Publication 800-218

Secure Software Development
Framework (SSDF) Version 1.1:

Recommendations for Mitigating
the Risk of Software Vulnerabilities



NIST SP 800-218 SSDF VERSION 1.1

Table 1: The Secure Software Development Framework (SSDF) Version 1.1

Practices	Tasks	Notional Implementation Examples	References
Prepare the Organization (PO)			
Define Security Requirements for Software Development (PO.1): Ensure that security requirements for software development are known at all times so that they can be taken into account throughout the SDLC and duplication of effort can be minimized because the requirements information can be collected once and shared. This includes requirements from internal sources (e.g., the organization's policies, business objectives, and risk management strategy) and external sources (e.g., applicable laws and regulations).	**PO.1.1**: Identify and document all security requirements for the organization's software development infrastructures and processes, and maintain the requirements over time.	**Example 1**: Define policies for securing software development infrastructures and their components, including development endpoints, throughout the SDLC and maintaining that security. **Example 2**: Define policies for securing software development processes throughout the SDLC and maintaining that security, including for open-source and other third-party software components utilized by software being developed. **Example 3**: Review and update security requirements at least annually, or sooner if there are new requirements from internal or external sources, or a major security incident targeting software development infrastructure has occurred. **Example 4**: Educate affected individuals on impending changes to requirements.	BSAFSS: SM.3, DE.1, IA.1, IA.2 BSIMM: CP1.1, CP1.3, SR1.1, SR2.2, SE1.2, SE2.6 EO14028: 4e(ix) IEC62443: SM-7, SM-9 NISTCSF: ID.GV-3 OWASPASVS: 1.1.1 OWASPMASVS: 1.10 OWASPSAMM: PC1-A, PC1-B, PC2-A PCISSLC: 2.1, 2.2 SCFPSSD: Planning the Implementation and Deployment of Secure Development Practices SP80053: SA-1, SA-8, SA-15, SR-3 SP800160: 3.1.2, 3.2.1, 3.2.2, 3.3.1, 3.4.2, 3.4.3 SP800161: SA-1, SA-8, SA-15, SR-3 SP800181: T0414; K0003, K0039, K0044, K0157, K0168, K0177, K0211, K0260, K0261, K0262, K0524; S0010, S0357, S0368; A0033, A0123, A0151
	PO.1.2: Identify and document all security requirements for organization-developed software to meet, and maintain the requirements over time.	**Example 1**: Define policies that specify risk-based software architecture and design requirements, such as making code modular to facilitate code reuse and updates; isolating security components from other components during execution; avoiding undocumented commands and settings; and providing features that will aid software acquirers with the secure deployment, operation, and maintenance of the software. **Example 2**: Define policies that specify the security requirements for the organization's software, and verify compliance at key points in the SDLC (e.g., classes of software flaws verified by gates, responses to vulnerabilities discovered in released software). **Example 3**: Analyze the risk of applicable technology stacks (e.g., languages, environments, deployment models), and recommend or require the use of stacks that will reduce risk compared to others. **Example 4**: Define policies that specify what needs to be archived for each software release (e.g., code, package files, third-party libraries, documentation, data inventory) and how long it needs to be retained based on the SDLC model, software end-of-life, and other factors. **Example 5**: Ensure that policies cover the entire software life cycle, including notifying users of the impending end of software support and the date of software end-of-life. **Example 6**: Review all security requirements at least annually, or sooner if there are new requirements from internal or external sources, a major vulnerability is discovered in released software, or a major security incident targeting organization-developed software has occurred. **Example 7**: Establish and follow processes for handling requirement exception requests, including periodic reviews of all approved exceptions.	**BSAFSS**: SC.1-1, SC.2, PD.1-1, PD.1-2, PD.1-3, PD.2-2, SI, PA, CS, AA, LO, EE **BSIMM**: SM1.1, SM1.4, SM2.2, CP1.1, CP1.2, CP1.3, CP2.1, CP2.3, AM1.2, SFD1.1, SFD2.1, SFD3.2, SR1.1, SR1.3, SR2.2, SR3.3, SR3.4 **EO14028**: 4e(ix) **IEC62443**: SR-3, SR-4, SR-5, SD-4 ISO27034: 7.3.2 MSSDL: 2, 5 **NISTCSF**: ID.GV-3 **OWASPMASVS**: 1.12 **OWASPSAMM**: PC1-A, PC1-B, PC2-A, PC3-A, SR1-A, SR1-B, SR2-B, SA1-B, IR1-A **PCISSLC**: 2.1, 2.2, 2.3, 3.3 **SCFPSSD**: Establish Coding Standards and Conventions **SP80053**: SA-8, SA-8(3), SA-15, SR-3 **SP800160**: 3.1.2, 3.2.1, 3.3.1 **SP800161**: SA-8, SA-15, SR-3 **SP800181**: T0414; K0003, K0039, K0044, K0157, K0168, K0177, K0211, K0260, K0261, K0262, K0524; S0010, S0357, S0368; A0033, A0123, A0151
	PO.1.3: Communicate requirements to all third parties who will provide commercial software components to the organization for reuse by the organization's own software. [Formerly PW.3.1]	**Example 1**: Define a core set of security requirements for software components, and include it in acquisition documents, software contracts, and other agreements with third parties. **Example 2**: Define security-related criteria for selecting software; the criteria can include the third party's vulnerability disclosure program and product security incident response capabilities or the third party's adherence to organization-defined practices. **Example 3**: Require third parties to attest that their software complies with the organization's security requirements.	**BSAFSS**: SM.1, SM.2, SM.2-1, SM.2-4 **BSIMM**: CP2.4, CP3.2, SR2.5, SR3.2 **EO14028**: 4e(vi), 4e(ix) IDASOAR: 19, 21 **IEC62443**: SM-9, SM-10 **MSSDL**: 7 **NISTCSF**: ID.SC-3 **OWASPSAMM**: SR3-A

SSDF (SP 800-218)

NIST - Secure Software Development Framework

v1.1 での□□□□(□□□□□□□)

Section	Title
PO.1.2	セキュリティ要求事項の文書化
PO.5.1	ソフトウェア開発における環境の□□/□□
PO.5.2	開発関連作業用のエンドポイントの堅牢化
PS.3.2	ソフトウェアの全コンポーネントの構成情報の収集、保護、維持、共有
PW.1.2	セキュリティ要求や設計判断の記録

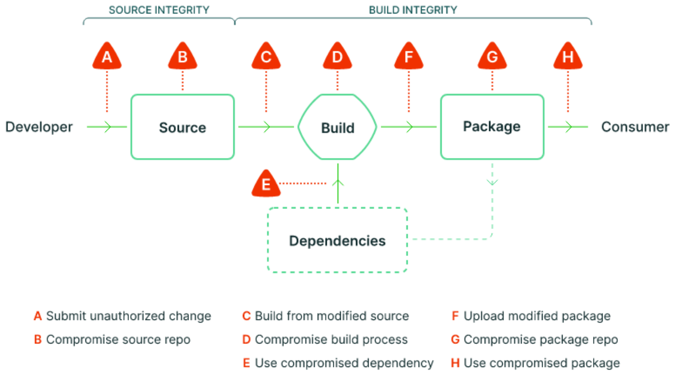
Appendix A—The SSDF and Executive Order 14028

The President's Executive Order (EO) on "Improving the Nation's Cybersecurity (14028)" issued on May 12, 2021 [EO14028], charged multiple agencies – including NIST – with enhancing cybersecurity through a variety of initiatives related to the security and integrity of the software supply chain.

Section 4 of the EO directed NIST to solicit input from the private sector, academia, government agencies, and others and to identify existing or develop new standards, tools, best practices, and other guidelines to enhance software supply chain security. Table 2 maps the subsections from Section 4e of the EO to SSDF practices and tasks that can help address each subsection as part of a risk-based approach.

Table 2: SSDF Practices Corresponding to EO 14028 Subsections

EO 14028 Subsection	SSDF Practices and Tasks
4e(i)(A)	PO.5.1
4e(i)(B)	PO.5.1
4e(i)(C)	PO.5.1, PO.5.2
4e(i)(D)	PO.5.1
4e(i)(E)	PO.5.2
4e(i)(F)	PO.3.2, PO.3.3, PO.5.1, PO.5.2
4e(ii)	PO.3.2, PO.3.3, PO.5.1, PO.5.2
4e(iii)	PO.3.1, PO.3.2, PO.5.1, PO.5.2, PS.1.1, PS.2.1, PS.3.1, PW.4.1, PW.4.4
4e(iv)	PO.4.1, PO.4.2, PS.1.1, PW.2.1, PW.4.4, PW.5.1, PW.6.1, PW.6.2, PW.7.1, PW.7.2, PW.8.2, PW.9.1, PW.9.2, RV.1.1, RV.1.2, RV.2.1, RV.2.2, RV.3.3
4e(v)	PO.3.2, PO.3.3, PO.4.1, PO.4.2, PO.5.1, PO.5.2, PW.1.2, PW.2.1, PW.7.2, PW.8.2, RV.2.2
4e(vi)	PO.1.3, PO.3.2, PO.5.1, PO.5.2, PS.3.1, PS.3.2, PW.4.1, PW.4.4, RV.1.1, RV.1.2
4e(vii)	PS.3.2
4e(viii)	RV.1.1, RV.1.2, RV.1.3, RV.2.1, RV.2.2, RV.3.3
4e(ix)	All practices and tasks consistent with a risk-based approach
4e(x)	PS.2.1, PS.3.1, PS.3.2, PW.4.1, PW.4.4



SLSA

Google - Supply chain Levels for Software Artifacts

ソフトウェアアーティファクトの☒☒☒(Integrity)を保証するためのEnd-to-endのフレームワーク。
3☒☒Trust Boundariesを設け、それぞれの領域の対応状況によりレベルを設定する。

Boundary Name	Requirements
Source Integrity	4
Build Integrity	8
Dependencies	5
(Common)	3

A Submit unauthorized change
B Compromise source repo
C Build from modified source
D Compromise build process
E Use compromised dependency
F Upload modified package
G Compromise package repo
H Use compromised package

Summary table

Requirement	SLSA 1	SLSA 2	SLSA 3	SLSA 4
Source - **Version controlled**		✓	✓	✓
Source - **Verified history**			✓	✓
Source - **Retained indefinitely**			18 mo.	✓
Source - **Two-person reviewed**				✓
Build - **Scripted build**	✓	✓	✓	✓
Build - **Build service**		✓	✓	✓
Build - **Build as code**			✓	✓
Build - **Ephemeral environment**			✓	✓

☒☒: SLSA • Introduction

CNCF

CNCF - Software Supply Chain Best Practices

サプライチェーンが侵害された事例を収集・分析し、攻撃のリスクを低減するための推奨事項や利用可能なツールを開発ステージごとに紹介。

- **Securing the Source Code**
- **Securing the Materials**
- **Securing the Build Pipelines**
- **Securing the Artefacts**
- **Securing Deployments**

Securing the Source Code

The foundational construct of any software supply chain is the source code. The initial step in securing a supply chain is establishing and ensuring the integrity of the source code. “Integrity” in this context means that the source code and tags found in the repository have not changed or altered from the code created by the developer. This includes potential malicious changes introduced by a local compromise on the developers machine. To maintain integrity, organizations must take steps to verify the source of the code added to their first party products and libraries.

It is fundamental to the supply chain that the development activity employs software development best practices. Agile methodologies of “continuous improvement” have been embraced in the industry and enabled through CI/CD pipelines and automated testing. These pipelines must be properly configured to access source code on-demand in order to build, deploy, and release artefacts at the cadence the organization needs.

Identity and Access management (IAM) is the biggest attack vector, regardless of platform or vendor, and it is critical to carefully manage IAM policies to provide both developers and agents secure access to source code. Pipeline agents and human developers must have their access and privileges calibrated to their roles within the organization and be given secure means to authenticate to those roles.

Verification:

Require signed commits

Assurance categories: Moderate to high and risk categories: Moderate to high

Signing source code[18] (commits and tags) ensures integrity and non-repudiation of source code by enabling out-of-band verification of the code. An attack on the source code management

☒☒: Microsoft contributes S2C2F to OpenSSF to improve supply chain security

CNCF

CNCF - Software Supply Chain Best Practices

☒49プラクティスで実践的なものが多く、かつSLSAよりも包括的。

Comparing SLSA to the CNCF Software Supply Chain Best Practices Recommendations

Target	Practices
Source Code	署名付きコミット、マージ前の☒☒(Tests, AST☒)☒Code Ownerの設定など
Materials	Checksum☒☒☒☒☒☒/ライセンススキャン、SBOMの生成など
Pipelines	再現可能なビルドを意識する、ネットワーク接続の☒☒☒☒ ☒☒の☒☒☒(SPIFEE)など
Artefacts	ビルドプロセスに対する署名及び検証、秘密鍵の更新方法に確率、成果物の☒☒☒(ocicrypto)など
Deployments	成果物に対するメタデータの検証、TUF(The Update Framework)の使用、など

S2C2F

OSSF - Secure Supply Chain Consumption Framework

OSSのConsumer側の視点に立☒☒☒☒☒☒☒☒☒☒　既知の攻撃手法の分析から導出☒☒8☒☒☒☒☒☒☒☒☒4段階の成熟モデルから成る。

- **Ingest It:** 外部OSSリソースを取り込む
- **Scan It:** 脆弱性等をスキャンする
- **Inventory It:** どこで使われているか管理する
- **Update It:** OSSの更新をすぐに適用可能にする
- **Enforce It:** OSSの利用方法を定める
- **Rebuild It:** 独自ビルドを実行可能にする
- **Fix It:** 脆弱性を修正できるようにする

Level 1	Level 2	Level 3	Level 4
Minimum OSS Governance Program	**Secure Consumption and Improved MTTR**	**Malware Defense and Zero-Day Detection**	**Advanced Threat Defense**
• Use package managers • Local copy of artifact • Scan with known vulns • Scan for software licenses • Inventory OSS • Manual OSS updates	• Scan for end life • Have an incident response plan • Auto OSS updates • Alert on vulns at PR time • Audit that consumption is through the approved ingestion method • Validate integrity of OSS • Secure package source file configuration	• Deny list capability • Clone OSS source • Scan for malware • Proactive security reviews • Enforce OSS provenance • Enforce consumption from curated feed	• Validate the SBOMs of OSS consumed • Rebuild OSS on trusted infrastructure • Digitally sign rebuilt OSS • Generate SBOM for rebuilt OSS • Digitally sign protected SBOMs • Implement fixes

☒☒: Microsoft contributes S2C2F to OpenSSF to improve supply chain security

Summary

- 各フレームワークに対する印象
  - **SSDF**: 組織論から運用まで、包括的に扱っている
  - **SLSA**: 簡易的だが導入としては良さそう
  - **CNCF**: 実践的なBest Practice集
  - **S2C2F**: 扱う領域がだいぶ絞られるため用途が限定的
- 開発者の立場だと**SLSA**で成熟度を見つつ、**CNCF**を参考にしながら成熟度を上げていく方針を取りそう
- 会社でやるなら**SSDF**なんじゃないですかね