

A close-up photograph of a wooden gavel lying diagonally across the frame. The gavel has a dark, polished head and a lighter-colored wooden handle. It rests on a dark, textured surface that shows concentric circular ripples, suggesting it's sitting on water or a liquid. The lighting is dramatic, highlighting the grain of the wood and the texture of the surface.

# Supply Chain Security Frameworks

Press Space for next page →

# 背景

- ソフトウェアサプライチェーン/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

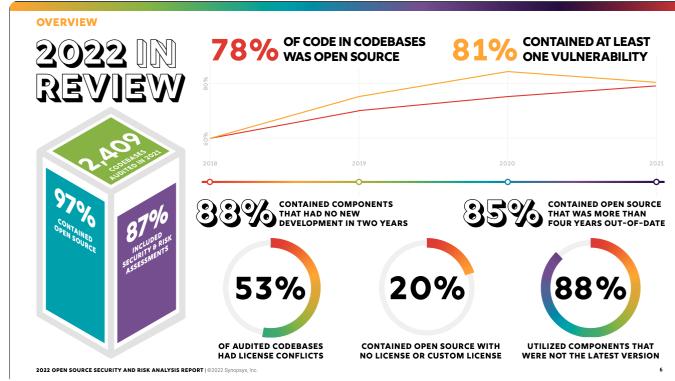
# 背景

# 背景

第三者の成果物への依存度が高まっている  
ある調査で2,409のコードベースの内容を調べると...

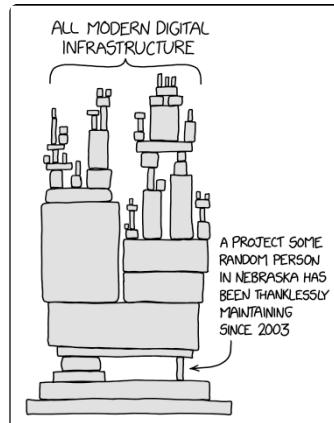
- 97% がオープンソースを含んでいた
- コードベース内の 78% のコードがOSSだった

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

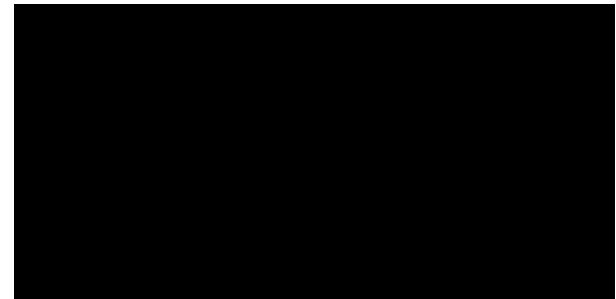


セキュリティの強度はその最も弱い部分で決定する。

*“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



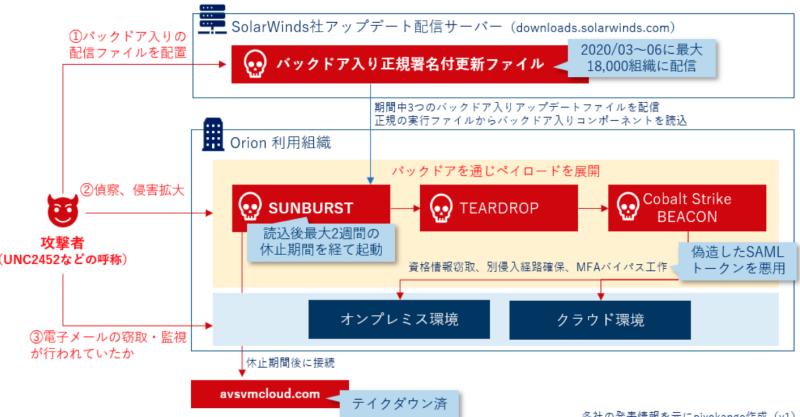
npm install...

# 背景

サプライチェーンセキュリティが脅かされた事例

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

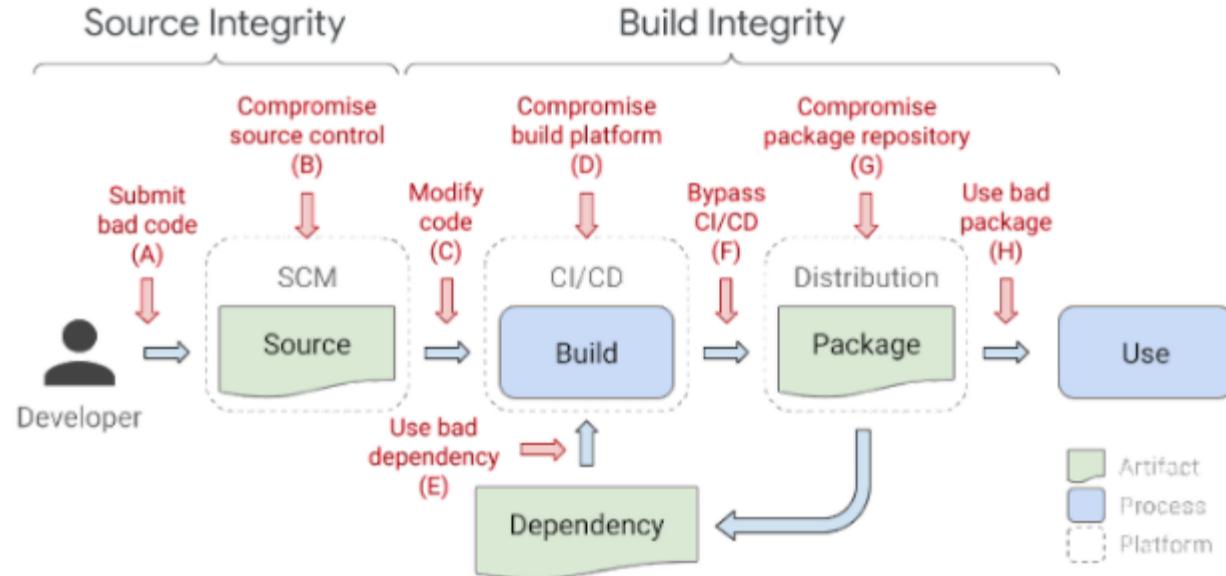
- `Orion`はネットワーク監視ソフトウェア
- Orionから配布された正規のパッチファイルにバックドア機能を持つDLLが含まれていた
  - ビルドサーバーに忍び込まれ、正規のパッチファイルだと認識されていた
- 2020年3~6月の間に最大で**18,000組織**に配信
  - **米国政府機関**をはじめ、Microsoft, Nvidia, Intelなど



参考: SolarWindsのサプライチェーン攻撃についてまとめてみた \- piyolog

# 背景

## Supply Chain Attacking Surface



ref: [Protect your open source project from supply chain attacks | Google Open Source Blog](#)

# Executive Order 14028

## 米国政府の対応

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

\* 余談: `SBOM` に関して述べているのは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<br>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て、特定のインシデント情報の共有を義務づける。   |
| 2<br>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<br>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（ <u>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</u> ）を確立し、特に <u>重要なソフトウェアに対して一定の対策を義務づける</u> 。   |
| 4<br>サイバー安全審査委員会の創設 (Section 5)             | ● 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リオット制度を開始する。  |
| 5<br>インシデント対応のための標準フレイムックの策定 (Section 6, 7) | ● 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具体的な提言を行う権限を与える。   |
| 6<br>調査及び修復能力の向上 (Section 8)                | ● 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フレイムックを策定する。<br>● 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。<br>● 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

6

参考: 最近の産業サイバーセキュリティに関する動向について\-\ 経済産業省

# Supply Chain Security Frameworks

# Supply Chain Security Frameworks

Abbr	Name	Org	Published	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完全性(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

未検出/未対処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*

Table 3: The Feature Software Development Framework (FSDF) Version 1.0

# SSDF (SP 800-218)

NIST - Secure Software Development Framework

v1.1 での改訂箇所(追加セクション)

## 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分離/保護

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

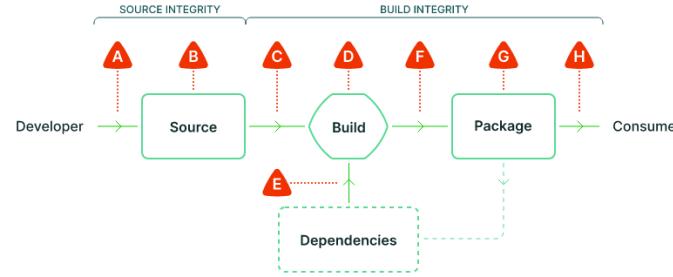
EO 14028 Subsection	SSDF Practices and Tasks
<a href="#">4e(i)(A)</a>	PO.5.1
<a href="#">4e(i)(B)</a>	PO.5.1
<a href="#">4e(i)(C)</a>	PO.5.1, PO.5.2
<a href="#">4e(i)(D)</a>	PO.5.1
<a href="#">4e(i)(E)</a>	PO.5.2
<a href="#">4e(i)(F)</a>	PO.3.2, PO.3.3, PO.5.1, PO.5.2
<a href="#">4e(ii)</a>	PO.3.2, PO.3.3, PO.5.1, PO.5.2
<a href="#">4e(iii)</a>	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
<a href="#">4e(iv)</a>	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
<a href="#">4e(v)</a>	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
<a href="#">4e(vi)</a>	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
<a href="#">4e(vii)</a>	PS.3.2
<a href="#">4e(viii)</a>	RV.1.1, RV.1.2, RV.1.3, RV.2.1, RV.2.2, RV.3.3
<a href="#">4e(ix)</a>	All practices and tasks consistent with a risk-based approach
<a href="#">4e(x)</a>	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完全性(Integrity)を保証するためのEnd-to-endのフレームワーク。  
3つの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
<b>Source Integrity</b>	4
<b>Build Integrity</b>	8
<b>Dependencies</b>	5
(Common)	3



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

## Summary table

Requirement	SLSA 1	SLSA 2	SLSA 3	SLSA 4
Source - <a href="#">Version controlled</a>	✓	✓	✓	
Source - <a href="#">Verified history</a>		✓	✓	
Source - <a href="#">Retained indefinitely</a>			18 mo.	✓
Source - <a href="#">Two-person reviewed</a>				✓
Build - <a href="#">Scripted build</a>	✓	✓	✓	✓
Build - <a href="#">Build service</a>	✓	✓	✓	
Build - <a href="#">Build as code</a>		✓	✓	
Build - <a href="#">Ephemeral environment</a>		✓	✓	

# 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 developer's 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 artifacts 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

Signed source code<sup>18</sup> (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

# CNCF

## CNCF - Software Supply Chain Best Practices

全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、マージ前の検証(Tests, AST等)、Code Ownerの設定など

**Materials** Checksum検証、脆弱性/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最小化、短命の証明書(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、成果物の暗号化(ocirypto)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側の視点に立ったフレームワーク。  
既知の攻撃手法の分析から導出した8つのプラクティスと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
<p> Minimum OSS Governance Program</p> <ul style="list-style-type: none"><li>• Use package managers</li><li>• Local copy of artifact</li><li>• Scan with known vulns</li><li>• Scan for software licenses</li><li>• Inventory OSS</li><li>• Manual OSS updates</li></ul>	<p> Secure Consumption and Improved MTTR</p> <ul style="list-style-type: none"><li>• Scan for end life</li><li>• Have an incident response plan</li><li>• Auto OSS updates</li><li>• Alert on vulns at PR time</li><li>• Audit that consumption is through the approved ingestion method</li><li>• Validate integrity of OSS</li><li>• Secure package source file configuration</li></ul>	<p> Malware Defense and Zero-Day Detection</p> <ul style="list-style-type: none"><li>• Deny list capability</li><li>• Clone OSS source</li><li>• Scan for malware</li><li>• Proactive security reviews</li><li>• Enforce OSS provenance</li><li>• Enforce consumption from curated feed</li></ul>	<p> Advanced Threat Defense</p> <ul style="list-style-type: none"><li>• Validate the SBOMs of OSS consumed</li><li>• Rebuild OSS on trusted infrastructure</li><li>• Digitally sign rebuilt OSS</li><li>• Generate SBOM for rebuilt OSS</li><li>• Digitally sign protected SBOMs</li><li>• Implement fixes</li></ul>

参考: 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