

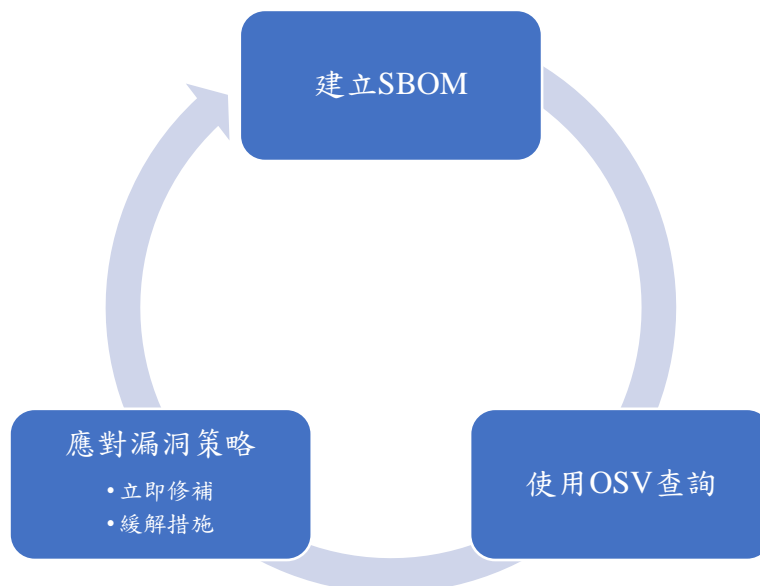
SBOM 開源工具使用說明

SBOM (Software Bill of Materials): SBOM 提供軟體中包含的所有函式庫與組件的清單，還會列出版本號、原始碼來源和關聯資訊。SBOM 的目的是讓使用者或開發者瞭解軟體的組成，這樣在發生安全事件或需要更新時，能夠快速識別和應對。

OSV (Open Source Vulnerability): 是一種公開的軟體漏洞資料庫。透過產生的 SBOM 資訊搭配 OSV 漏洞資料庫，開發人員可以快速查詢其軟體組件是否存在已知的安全問題。

以下是使用 SBOM 與 OSV 的基本流程：

1. **建立 SBOM:** 於系統維運時，定期利用 SBOM 工具建立一份 SBOM 文件，將所有的組件、函式庫和依賴性記錄下來。
2. **使用 OSV 查詢:** 定期將產生的 SBOM 資訊，利用 OSV 資料庫查詢，確認組件是否有已知的安全漏洞。一旦發現，可以迅速決策，例如進行更新或替換該組件。
3. **應對漏洞策略:** 如果在 OSV 中發現了任何漏洞，應該立即進行修補或緩解措施。這可能包括更新軟體組件、應用修補程式，或在使用中考慮其他安全措施。



此操作文件使用的 SBOM 工具有以下兩種：

- Microsoft sbom-tool
- CycloneDX Generator

並搭配 Google osv-scanner 掃描產生的 SBOM 檔案找出元件漏洞，進行修補或緩解措施。

依照相容性測試建議 Windows 環境可以使用 Microsoft sbom-tool，Linux 環境可以使用 Microsoft sbom-tool 與 CycloneDX Generator

本操作文件將使用 Windows 10 與 Ubuntu Desktop 22.04 LTS 進行以下測試

掃描 Github 開源專案 blaze 版本 2.1.2

網址：<https://github.com/blenderskool/blaze/tree/v2.1.2>

GitHub - blenderskool/blaze at v2.1.2

blenderskool / blaze Public

Sponsor Notifications Fork 258 Star 2k

Code Issues 10 Pull requests Actions Projects Security Insights

v2.1.2 Go to file Code About

blenderskool Merge pull request #113 from blender... on Apr 30, 2021 377

.github	Update link	2 years ago
api	Update width of release badge	2 years ago
client	Update packages	2 years ago
common	Move to common directory	3 years ago
nginx	Add reverse proxy for requests to blaze ser...	2 years ago
server	version 2.1.2	2 years ago
.dockerignore	Add api directory	2 years ago

About

⚡ File sharing progressive web app built using WebTorrent and WebSockets

blaze.now.sh/

node preact frontend backend webrtc websockets sharing webtorrent file-transfer collaborate hacktoberfest pwa-app

Readme

Microsoft sbom-tool 開源工具說明

Microsoft sbom-tool 開源專案，工具支援目前主流的 Package Managers，依照 Package Managers 與語言關係對應整理如下表：

語言	支援的 Package Managers
go	GoMod
Rust	Cargo
.NET	NuGet
Java	Maven、Gradle
Node.js	NPM、Yarn
Python	PIP、Poety
Ruby	Gems
Objective-C Swift	CocoaPods

資料彙整：<https://github.com/microsoft/component-detection>

● Windows10 環境操作步驟

步驟一、安裝 winget

藉由微軟商店安裝 winget

相關說明：<https://learn.microsoft.com/zh-tw/windows/package-manager/winget/>

安裝方法：微軟商店安裝”應用程式安裝程式”詳見圖 1

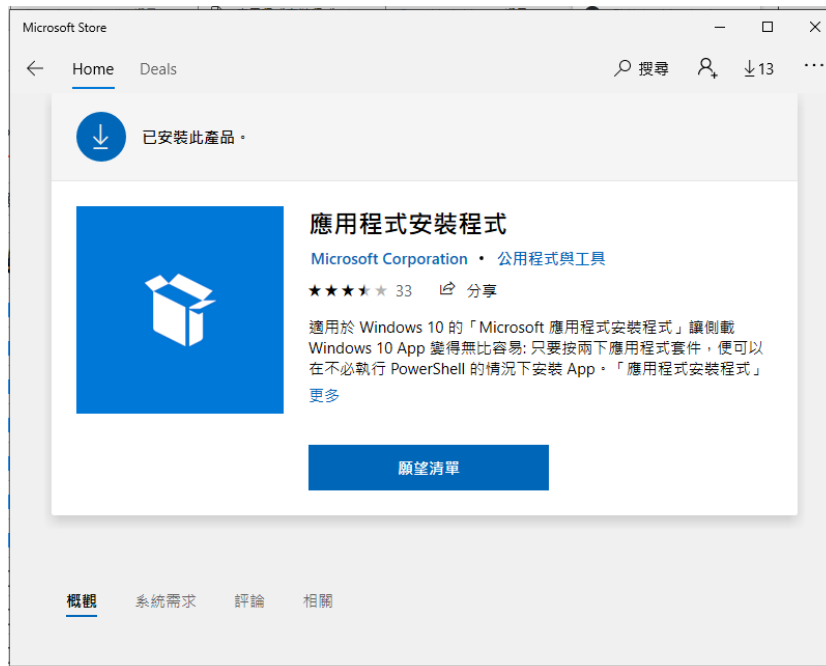


圖 1 微軟商店安裝 winget

步驟二、安裝 Microsoft sbom-tool 工具

開啟 Windows PowerShell 並輸入安裝指令

(安裝完成後請登出使用者再登入，以完成環境變數設定)

➤ `winget install Microsoft.SbomTool`

輸出畫面詳見圖 2

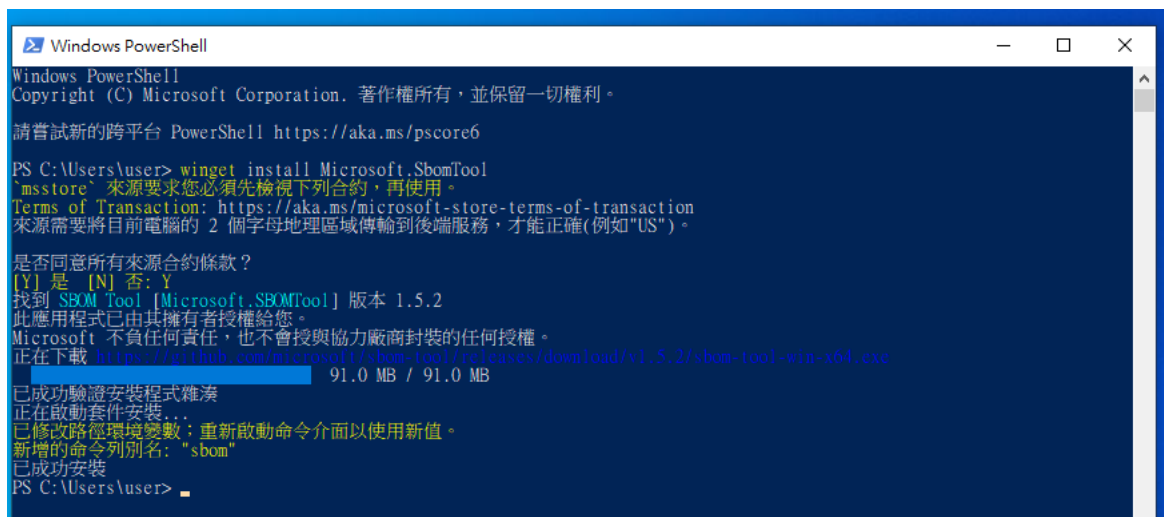


圖 2 安裝 Microsoft sbom-tool 工具指令結果畫面

步驟三、執行 Microsoft sbom-tool 掃描

3.1 將掃描原始檔資料夾” blaze-2.1.2” 放置到” 本機\文件” 內

3.2 在” 本機\文件” 內建立資料夾” blaze-sbom” 放置工具產生的 SBOM 檔案

完成後詳見圖 3

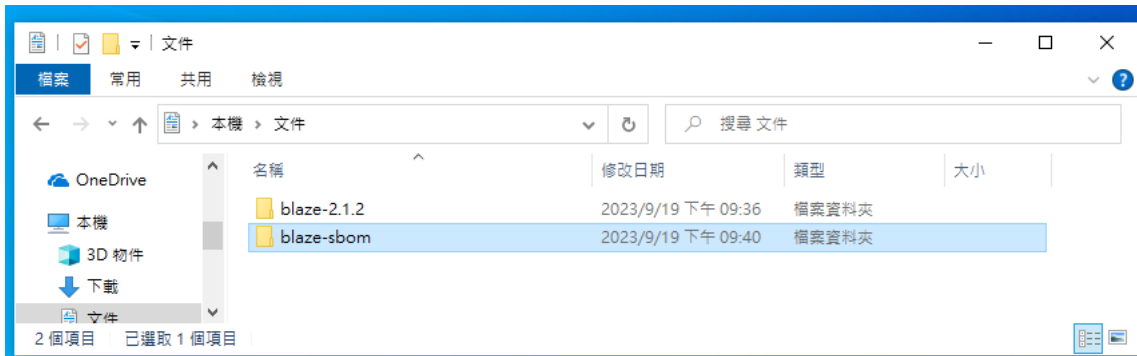


圖 3 本機\文件內的畫面

3.3 開啟 Windows PowerShell 輸入指令移動至” 本機\文件” 內

➤ `cd C:\Users\user\Documents\`

(此處使用者名稱會隨著不同電腦而變化，請依照本機使用者路徑替換)

3.4 開啟 Windows PowerShell 輸入指令執行 `sbom-tool-win-x64.exe`

➤ `sbom-tool-win-x64.exe generate -b "./blaze-sbom" -bc "./blaze-2.1.2" -pn "blaze" -pv "v1" -ps "nics" -D "true"`

參數說明：

-b "產生 SBOM 檔案放置目錄位置"

-bc "進行 SBOM 掃描的原始程式碼目錄位置"

-pn "SBOM 產生專案名稱"

-pv "版本號"

-ps "SBOM 產生單位"

-D "設置 true 將刪除目錄內已產生過的 SBOM 檔案"

完成後輸出畫面詳見圖 4

```

Windows PowerShell
PS C:\Users\user> cd C:\Users\user\Documents\
PS C:\Users\user\Documents> sbom-tool-win-x64.exe generate -b "/blaze-sbom" -bc "/blaze-2.1.2" -pn "blaze" -pv "v1" -ps "nics" -D "true"
## information Log file: "C:\Users\user\AppData\Local\Temp\GovCompDisc_Log_20230919214435674_4656.log"
## information Run correlation id: 17dd380a-66c1-409f-b742-13b1692f459a
## information Finding components...
## information
## information
## information |Component Detector Id|Detection Time|# Components Found|# Explicitly Referenced
## information |CocoaPods|0.063 seconds|10|10
## information |Go|0.062 seconds|10|10
## information |Gradle|0.063 seconds|10|10
## information |Ivy (Beta)|0.34 seconds|10|10
## information |Linux|0.21 seconds|10|10
## information |MvnCli|0.8 seconds|10|10
## information |Npm|0.33 seconds|14|10
## information |NpmLockfile3 (Beta)|0.38 seconds|10|10
## information |NpmWithRoots|0.8 seconds|1346|29
## information |NuGet|0.39 seconds|10|10
## information |NuGetPackagesConfig|0.41 seconds|10|10
## information |NuGetProjectCentric|0.41 seconds|10|10
## information |Pip|0.97 seconds|10|10
## information |Pnpm|0.8 seconds|10|10
## information |Poetry (Beta)|0.8 seconds|10|10
## information |Ruby|0.8 seconds|10|10
## information |RustCrateDetector|0.8 seconds|10|10
## information |SPDX2SBOM|0.057 seconds|10|10
## information |Vcpkg (Beta)|0.8 seconds|10|10
## information |Yarn|0.06 seconds|10|10
## information |Total|0.98 seconds|11350|29
## information
## information Detection time: 0.9835563 seconds.
## information Scan Manifest file: "C:\Users\user\AppData\Local\Temp\ScanManifest_20230919214435539.json"
PS C:\Users\user\Documents>

```

圖 4 執行 sbom-tool-win-x64 指令結果畫面

步驟四、查看 SBOM 檔案

4.1 開啟”本機\文件\blaze-sbom_manifest\spdx_2.2”目錄，找到 manifest.spdx.json 檔案詳見圖 5

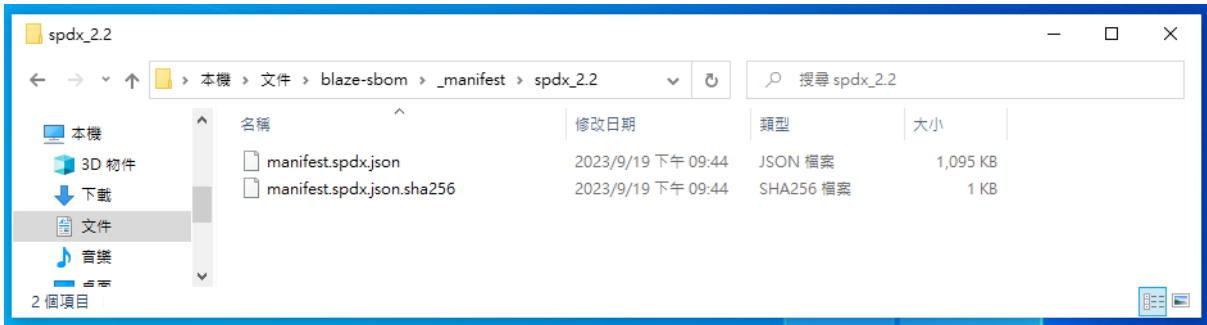


圖 5 manifest.spdx.json 檔案位置

4.2 開啟 manifest.spdx.json 檔案後，由 packages 查看元件版本詳見圖 6

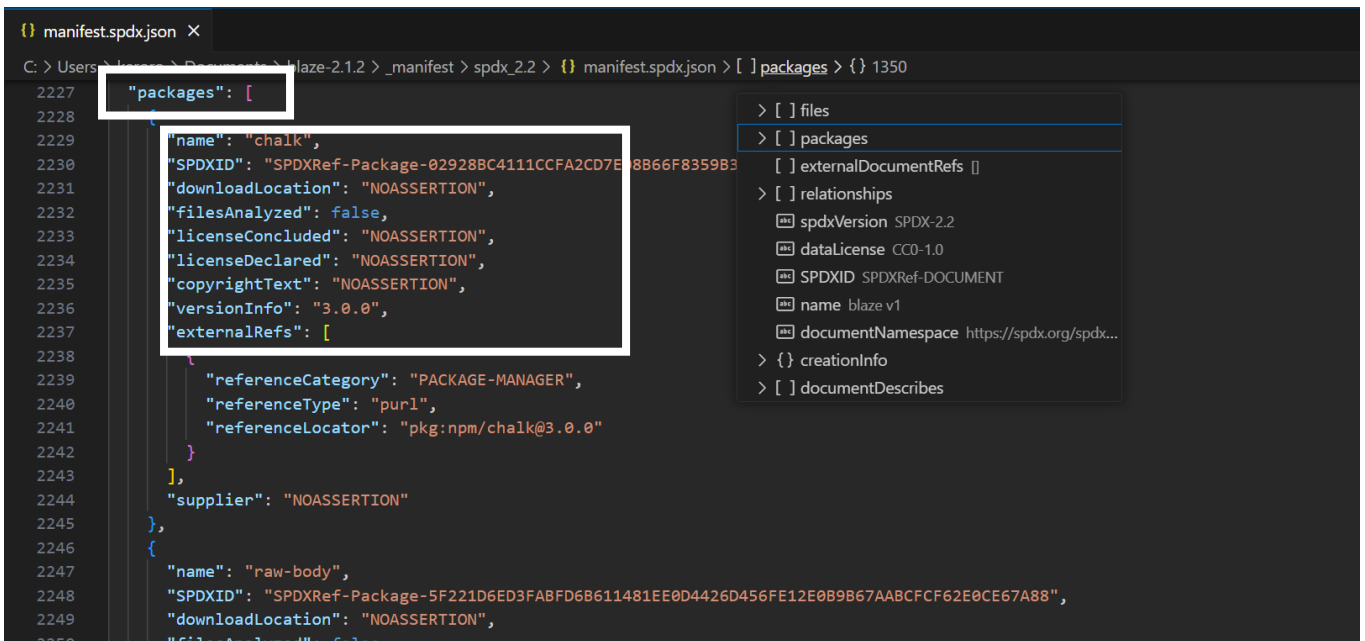
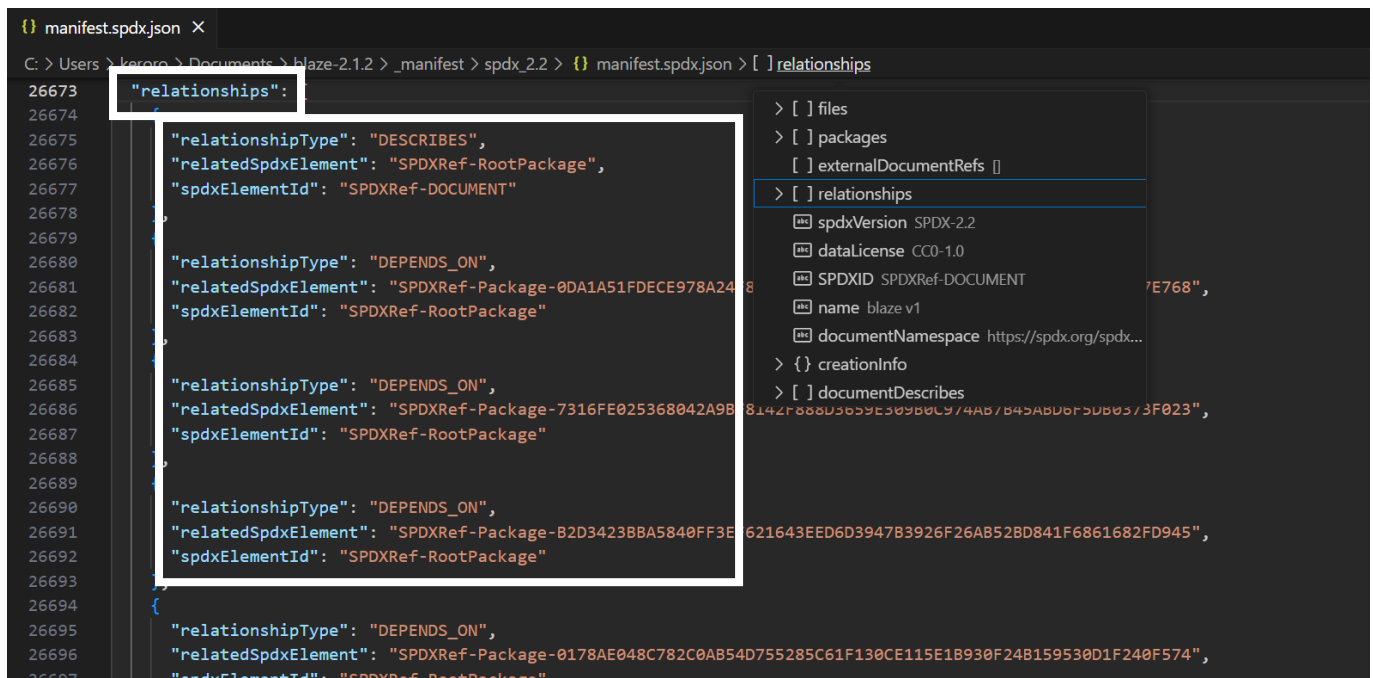


圖 6 manifest.spdx.json 查看元件版本

4.3 開啟 manifest.spdx.json 檔案後，由 relationships 查看元件關係詳見圖 7



```
manifest.spdx.json x
C: > Users > keroro > Documents > blaze-2.1.2 > _manifest > spdx_2.2 > {} manifest.spdx.json > [ ] relationships
26673 "relationships":
26674 {
26675   "relationshipType": "DESCRIBES",
26676   "relatedSpdxElement": "SPDXRef-RootPackage",
26677   "spdxElementId": "SPDXRef-DOCUMENT"
26678 },
26679 {
26680   "relationshipType": "DEPENDS_ON",
26681   "relatedSpdxElement": "SPDXRef-Package-0DA1A51FDECE978A24...",
26682   "spdxElementId": "SPDXRef-RootPackage"
26683 },
26684 {
26685   "relationshipType": "DEPENDS_ON",
26686   "relatedSpdxElement": "SPDXRef-Package-7316FE025368042A9B...",
26687   "spdxElementId": "SPDXRef-RootPackage"
26688 },
26689 {
26690   "relationshipType": "DEPENDS_ON",
26691   "relatedSpdxElement": "SPDXRef-Package-B2D3423BBA5840FF3E...",
26692   "spdxElementId": "SPDXRef-RootPackage"
26693 },
26694 {
26695   "relationshipType": "DEPENDS_ON",
26696   "relatedSpdxElement": "SPDXRef-Package-0178AE048C782C0AB54D755285C61F130CE115E18930F24B159530D1F240F574",
26697   "spdxElementId": "SPDXRef-RootPackage"
26698 }
}

> [ ] files
> [ ] packages
[ ] externalDocumentRefs []
> [ ] relationships
  [x] spdxVersion SPDX-2.2
  [x] dataLicense CC0-1.0
  [x] SPDXID SPDXRef-DOCUMENT
  [x] name blaze v1
  [x] documentNamespace https://spdx.org/spdx...
> {} creationInfo
> [ ] documentDescribes
```

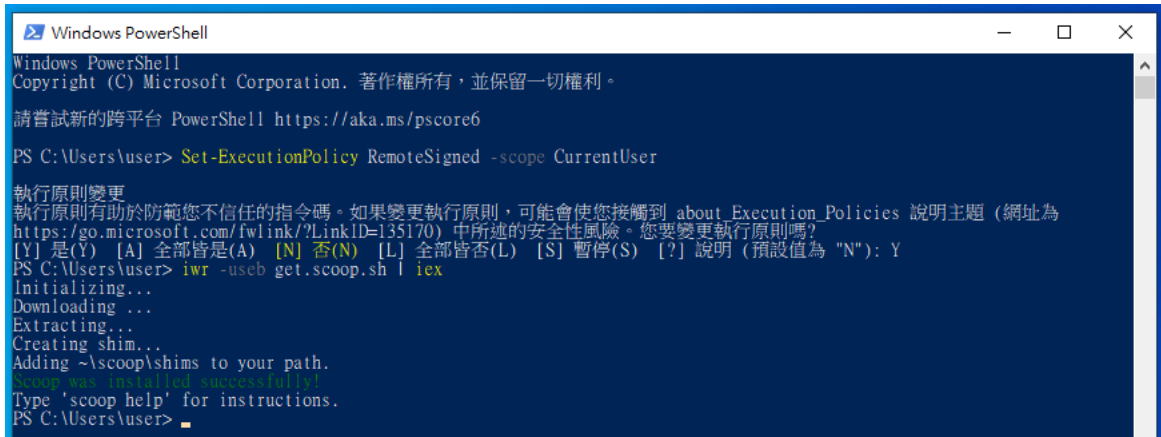
圖 7 manifest.spdx.json 查看元件關係

步驟五、安裝 scoop

開啟 Windows PowerShell 輸入安裝指令

- Set-ExecutionPolicy RemoteSigned -scope CurrentUser
- iwr -useb get.scoop.sh | iex

完成後輸出畫面詳見圖 8



```
Windows PowerShell
Copyright (C) Microsoft Corporation. 著作權所有，並保留一切權利。
請嘗試新的跨平台 PowerShell https://aka.ms/pscore6

PS C:\Users\user> Set-ExecutionPolicy RemoteSigned -scope CurrentUser

執行原則變更
執行原則有助於防範您不信任的指令碼。如果變更執行原則，可能會使您接觸到 about_Execution_Policies 說明主題 (網址為
https://go.microsoft.com/fwlink/?LinkID=135170) 中所述的安全性風險。您要變更執行原則嗎?
[Y] 是(Y) [A] 全部皆是(A) [N] 否(N) [L] 全部皆否(L) [S] 暫停(S) [?] 說明 (預設值為 "N"): Y
PS C:\Users\user> iwr -useb get.scoop.sh | iex
Initializing...
Downloading ...
Extracting...
Creating shim...
Adding ~\scoop\shims to your path.
Scoop was installed successfully!
Type 'scoop help' for instructions.
PS C:\Users\user>
```

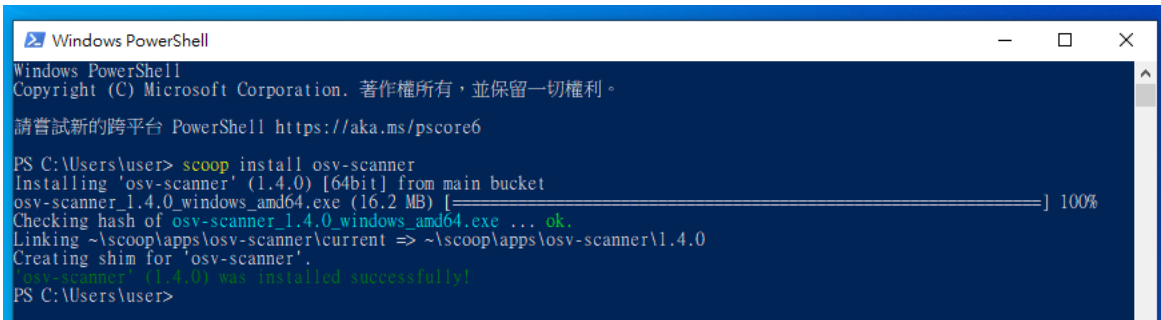
圖 8 安裝 scoop 指令結果畫面

步驟六、安裝 osv-scanner 工具進行

開啟 Windows PowerShell 輸入安裝指令

- scoop install osv-scanner

完成後輸出畫面詳見圖 9



```
Windows PowerShell
Copyright (C) Microsoft Corporation. 著作權所有，並保留一切權利。
請嘗試新的跨平台 PowerShell https://aka.ms/pscore6

PS C:\Users\user> scoop install osv-scanner
Installing 'osv-scanner' (1.4.0) [64bit] from main bucket
osv-scanner_1.4.0_windows_amd64.exe (16.2 MB) [=====] 100%
Checking hash of osv-scanner_1.4.0_windows_amd64.exe ... ok.
Linking ~\scoop\apps\osv-scanner\current => ~\scoop\apps\osv-scanner\1.4.0
Creating shim for 'osv-scanner'.
osv-scanner (1.4.0) was installed successfully!
PS C:\Users\user>
```

圖 9 安裝 osv-scanner 指令結果畫面

步驟七、執行 osv-scanner 工具掃描產出 json 檔，進行後續應對漏洞策略

7.1 開啟 Windows PowerShell 輸入指令移動至”本機\文件\blaze-sbom_manifest\spdx_2.2\”目錄

➤ `cd C:\Users\user\Documents\blaze-sbom_manifest\spdx_2.2`

(此處使用者名稱會隨著不同電腦而變化，請依照本機使用者路徑替換)

7.2 開啟 Windows PowerShell 輸入指令執行 osv-scanner

➤ `osv-scanner.exe --sbom="./manifest.spdx.json" --format json > ".file.json"`

參數說明：

--sbom "SBOM 檔案完整位置"

--format json > "json 檔案輸出位置"

完成後輸出畫面詳見圖 10



```
Windows PowerShell
Copyright (C) Microsoft Corporation. 著作權所有，並保留一切權利。
請嘗試新的跨平台 PowerShell https://aka.ms/pscore6

PS C:\Users\user> cd C:\Users\user\Documents\blaze-sbom\_manifest\spdx_2.2
PS C:\Users\user\Documents\blaze-sbom\_manifest\spdx_2.2> osv-scanner.exe --sbom="./manifest.spdx.json" --format json > ".file.json"
Scanned C:\Users\user\Documents\blaze-sbom\_manifest\spdx_2.2\manifest.spdx.json as SPDX SBOM and found 1351 packages
PS C:\Users\user\Documents\blaze-sbom\_manifest\spdx_2.2>
```

圖 10 執行 osv-scanner 指令結果畫面

7.3 開啟”本機\文件\blaze-sbom_manifest\spdx_2.2\”目錄，找到產生 file.json 檔案詳見圖 11

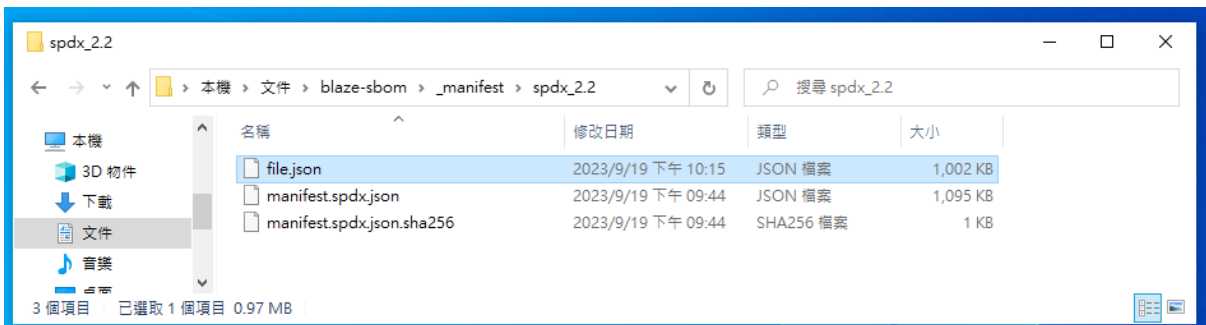
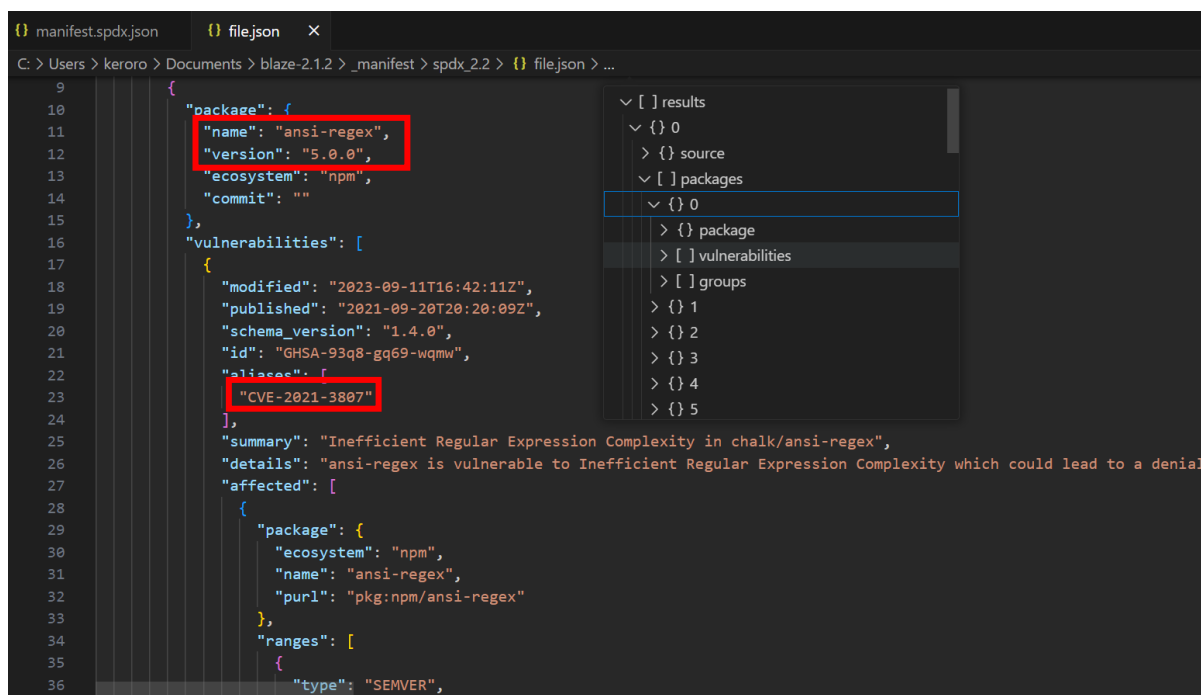


圖 11 file.json 檔案位置

步驟八、查看 file.json 檔案

開啟 file.json 檔案，文件詳細說明各元件弱點版本與 CVE 編號詳見圖 12，開始進行後續應對漏洞策略規劃



```
9
10 {
11   "package": {
12     "name": "ansi-regex",
13     "version": "5.0.0",
14     "ecosystem": "npm",
15     "commit": ""
16   },
17   "vulnerabilities": [
18     {
19       "modified": "2023-09-11T16:42:11Z",
20       "published": "2021-09-20T20:20:09Z",
21       "schema_version": "1.4.0",
22       "id": "GHSA-93q8-gq69-wqmw",
23       "aliases": [
24         "CVE-2021-3807"
25       ],
26       "summary": "Inefficient Regular Expression Complexity in chalk/ansi-regex",
27       "details": "ansi-regex is vulnerable to Inefficient Regular Expression Complexity which could lead to a denial",
28       "affected": [
29         {
30           "package": {
31             "ecosystem": "npm",
32             "name": "ansi-regex",
33             "purl": "pkg:npm/ansi-regex"
34           },
35           "ranges": [
36             {
37               "type": "SEMVER",
```

圖 12 file.json 檔案查看元件弱點版本與 CVE 編號

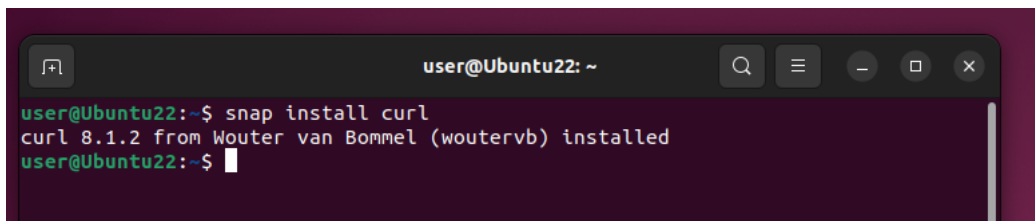
- Ubuntu Desktop 22.04 LTS 環境操作步驟

- 步驟一、安裝 curl

開啟 terminal 輸入安裝指令

- `snap install curl`

完成後輸出畫面詳見圖 13



```
user@Ubuntu22: ~  
user@Ubuntu22:~$ snap install curl  
curl 8.1.2 from Wouter van Bommel (woutervb) installed  
user@Ubuntu22:~$
```

圖 13 安裝 curl 指令結果畫面

- 步驟二、下載 sbom-tool

2.1 開啟 terminal 輸入指令下載 sbom-tool

- `curl -Lo sbom-tool https://github.com/microsoft/sbom-tool/releases/latest/download/sbom-tool-linux-x64`

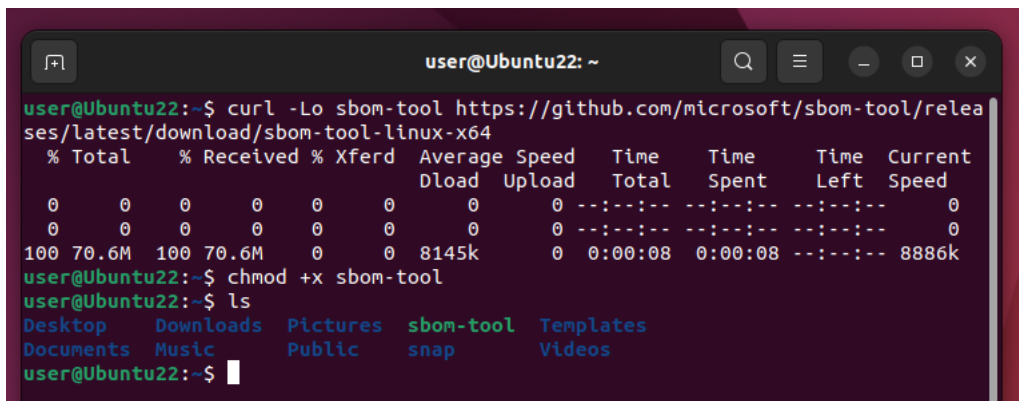
2.2 開啟 terminal 輸入指令設定 sbom-tool 權限

- `chmod +x sbom-tool`

2.3 開啟 terminal 輸入指令確認 sbom-tool 是否設定完成

- `ls`

完成後輸出畫面詳見圖 14



```
user@Ubuntu22: ~  
user@Ubuntu22:~$ curl -Lo sbom-tool https://github.com/microsoft/sbom-tool/releases/latest/download/sbom-tool-linux-x64  
% Total % Received % Xferd Average Speed Time Time Time Current  
 Dload Upload Total Spent Left Speed  
 0 0 0 0 0 0 0 0 0:00:00 0:00:00 0:00:00 0  
 0 0 0 0 0 0 0 0 0:00:00 0:00:00 0:00:00 0  
100 70.6M 100 70.6M 0 0 8145k 0 0:00:08 0:00:08 --:--:-- 8886k  
user@Ubuntu22:~$ chmod +x sbom-tool  
user@Ubuntu22:~$ ls  
Desktop Downloads Pictures sbom-tool Templates  
Documents Music Public snap Videos  
user@Ubuntu22:~$
```

圖 14 下載 sbom-tool 並設定權限結果畫面

步驟三、執行 Microsoft sbom-tool 掃描

3.1 將下載的原始檔檔案按右鍵選擇”Extract to...”進行解壓縮 zip，詳見圖 15

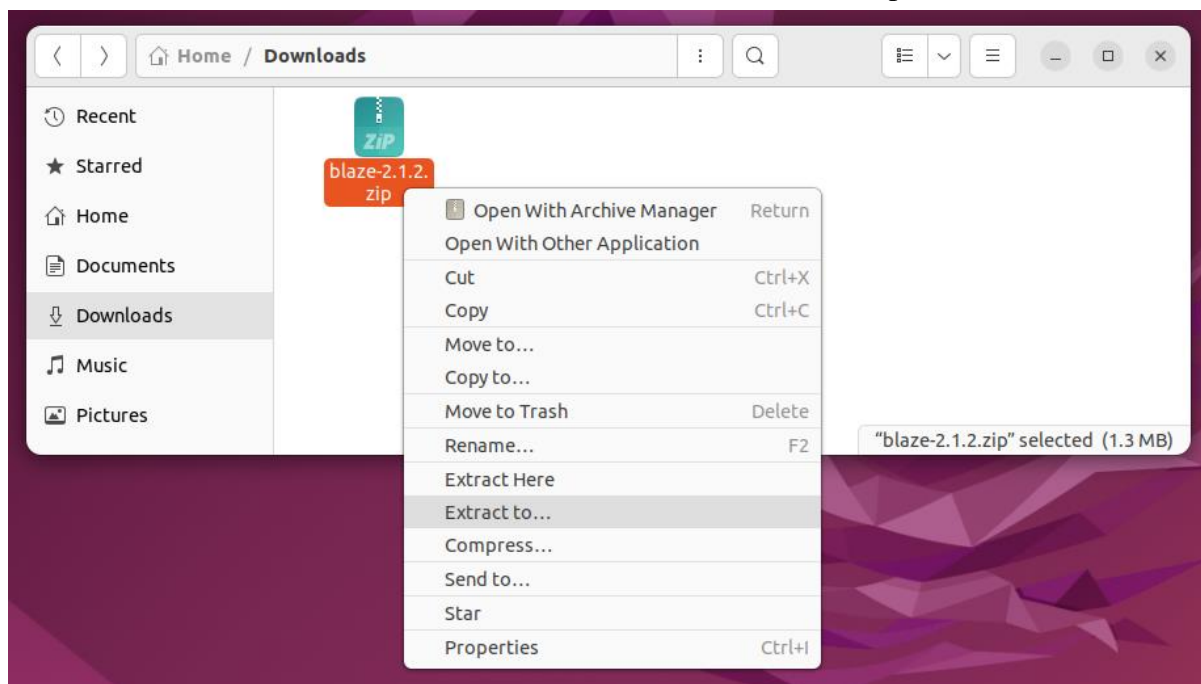


圖 15 對壓縮檔進行解壓縮操作(一)

3.2 選擇 Home 並按下 Select，詳見圖 16

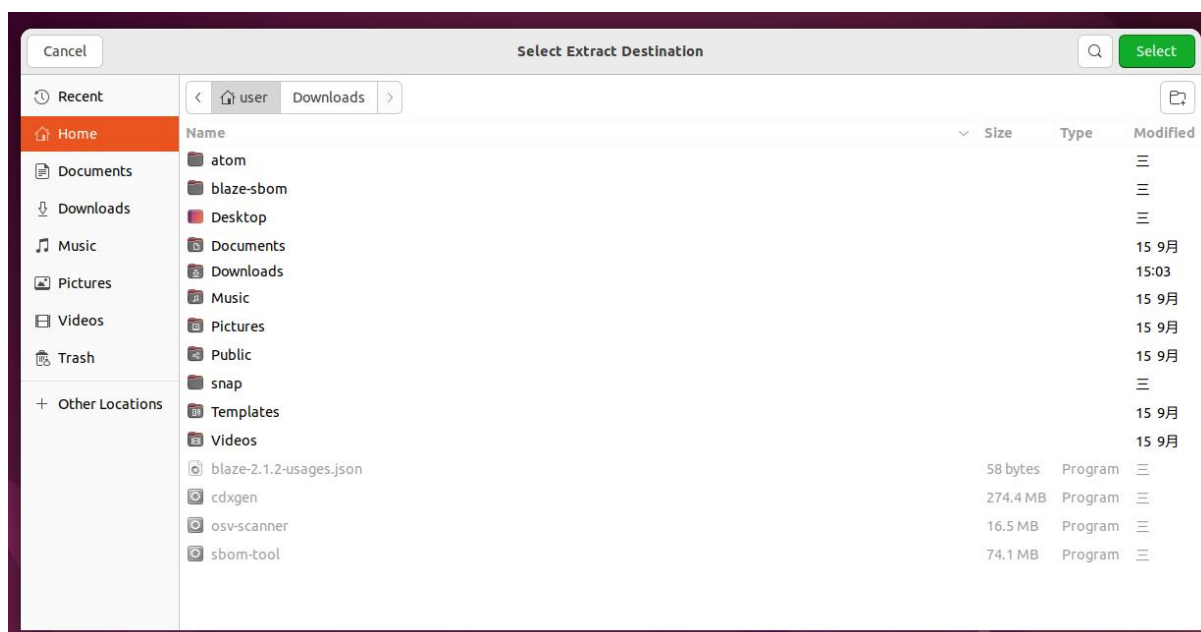


圖 16 對壓縮檔進行解壓縮操作(二)

3.3 於 Home 建立一個目錄” blaze-sbom” 放置 SBOM 檔案

完成後詳見圖 17

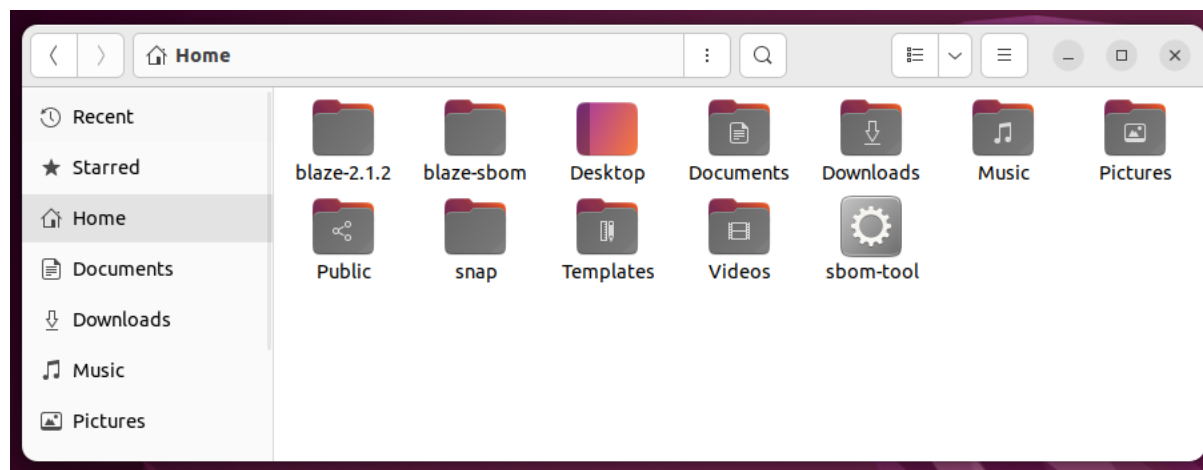


圖 17 放置原始碼與建立放置 SBOM 檔案的目錄

3.4 開啟 terminal 輸入執行 sbom-tool

➤ `./sbom-tool generate -b "/blaze-sbom" -bc "/blaze-2.1.2" -pn "blaze" -pv "v1" -ps "nics" -D "true"`

參數說明：

-b "產生 SBOM 檔案放置的目錄位置"

-bc "進行 SBOM 掃描的原始程式碼目錄位置"

-pn "SBOM 產生專案名稱"

-pv "版本號"

-ps "SBOM 產生單位"

-D "設置 true 將刪除目錄內已產生過的 SBOM 檔案"

完成後輸出畫面詳見圖 18 與圖 19

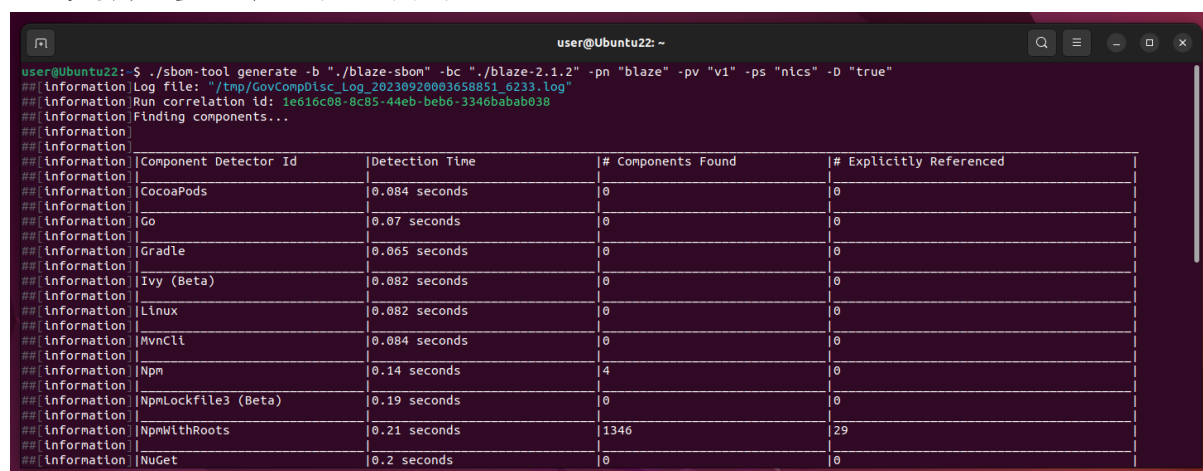


圖 18 執行 sbom-tool 指令結果畫面(上)

```
user@Ubuntu22: ~  
## Information  
## Information |-----|-----|-----|  
## Information | NUGet          | 0.2 seconds | 0         | 0         |  
## Information |-----|-----|-----|  
## Information | NUGetPackagesConfig | 0.2 seconds | 0         | 0         |  
## Information |-----|-----|-----|  
## Information | NUGetProjectCentric | 0.2 seconds | 0         | 0         |  
## Information |-----|-----|-----|  
## Information | Pip            | 0.23 seconds | 0         | 0         |  
## Information |-----|-----|-----|  
## Information | Pnpm           | 0.22 seconds | 0         | 0         |  
## Information |-----|-----|-----|  
## Information | Poetry (Beta)  | 0.22 seconds | 0         | 0         |  
## Information |-----|-----|-----|  
## Information | Ruby           | 0.22 seconds | 0         | 0         |  
## Information |-----|-----|-----|  
## Information | RustCrateDetector | 0.22 seconds | 0         | 0         |  
## Information |-----|-----|-----|  
## Information | SPDX22SBOM     | 0.069 seconds | 0         | 0         |  
## Information |-----|-----|-----|  
## Information | Vcpkg (Beta)   | 0.22 seconds | 0         | 0         |  
## Information |-----|-----|-----|  
## Information | Yarn           | 0.069 seconds | 0         | 0         |  
## Information |-----|-----|-----|  
## Information | Total          | 0.26 seconds | 1350      | 29        |  
## Information |-----|-----|-----|  
## Information |  
## Information | Detection time: 0.2564672 seconds.  
## Information | Scan Manifest file: "/tmp/ScanManifest_20230920003658697.json"  
user@Ubuntu22: $
```

圖 19 執行 sbom-tool 指令結果畫面(下)

步驟四、查看 SBOM 檔案

4.1 開啟” /home/blaze-sbom/_manifest/spdx_2.2” 目錄，找到 manifest.spdx.json 檔案詳見圖 20

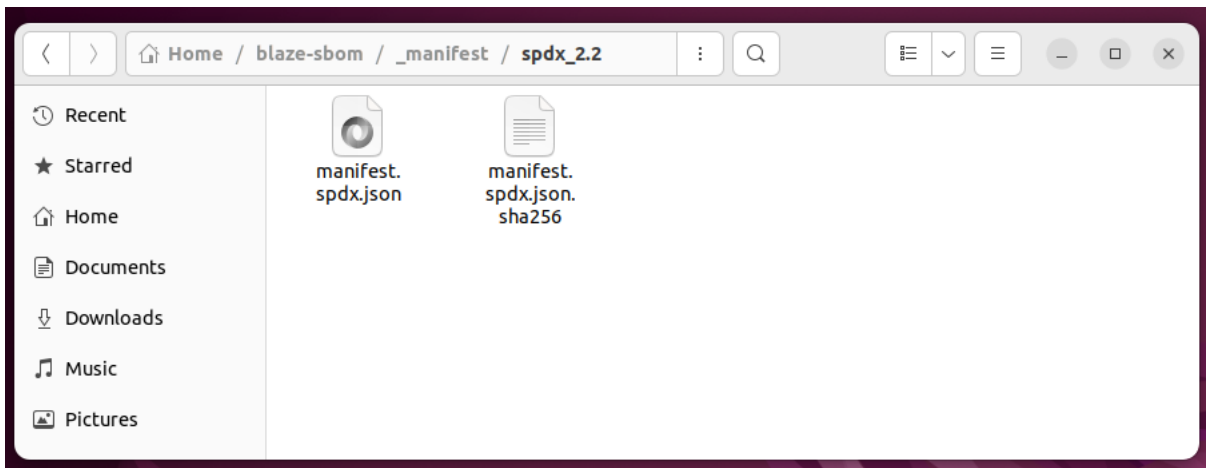


圖 20 manifest.spdx.json 檔案位置

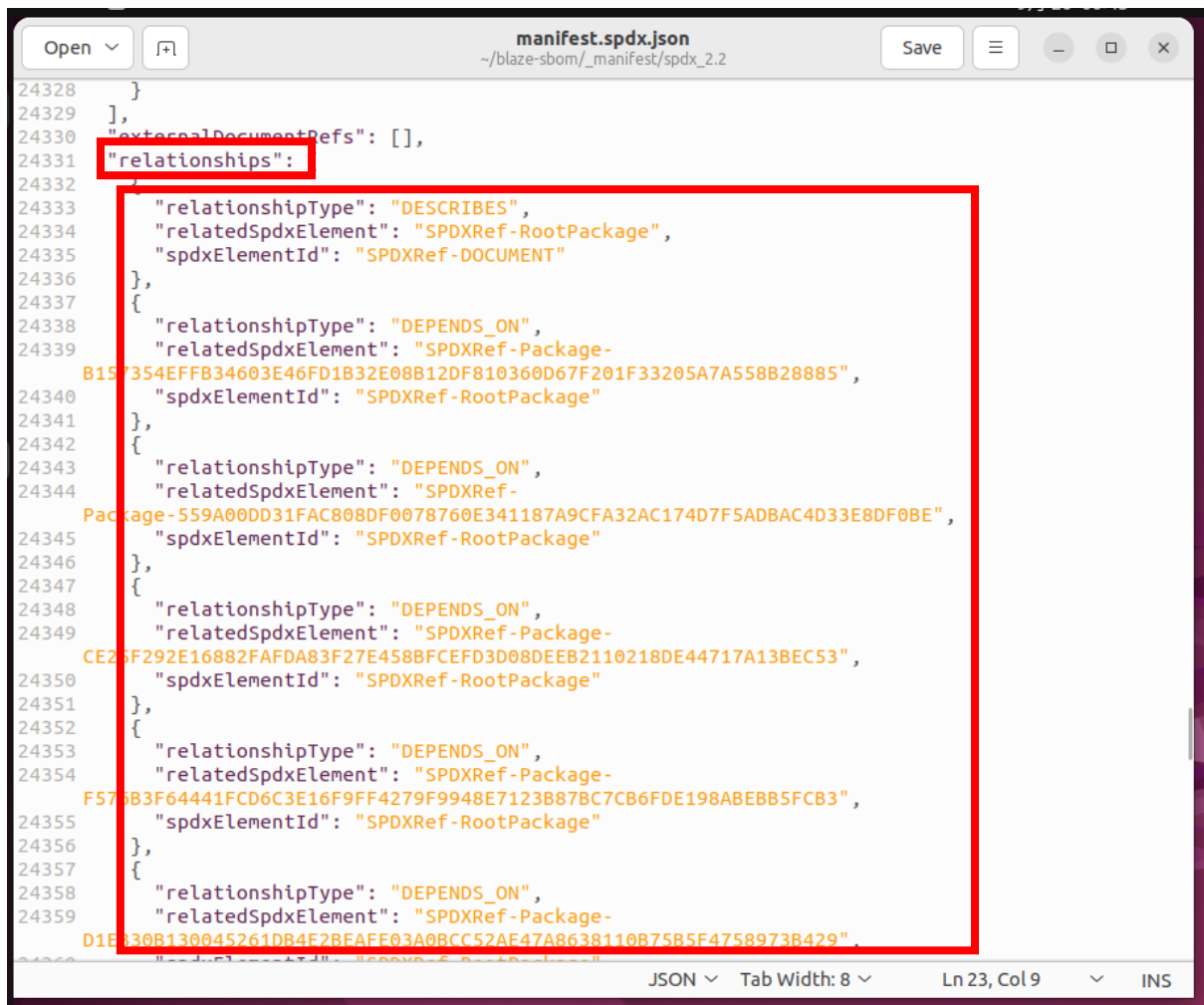
4.2 開啟 manifest.spdx.json 檔案後，由 packages 查看元件版本詳見圖 21



```
1 {
2   "files": [],
3   "packages": [
4     {
5       "name": "blaze-frontend",
6       "SPDXID": "SPDXRef-Package-
7       D95B0FC8B2F94D2F04ECA2F070995A241C1C35FB3379007579C4837F0E12DAE6",
8       "downloadLocation": "NOASSERTION",
9       "filesAnalyzed": false,
10      "licenseConcluded": "NOASSERTION",
11      "licenseDeclared": "NOASSERTION",
12      "copyrightText": "NOASSERTION",
13      "versionInfo": "2.1.2",
14      "externalRefs": [
15        {
16          "referenceCategory": "PACKAGE-MANAGER",
17          "referenceType": "purl",
18          "referenceLocator": "pkg:npm/blaze-frontend@2.1.2"
19        }
20      ],
21      "supplier": "Organization: Akash Hamirwasia"
22    },
23    {
24      "name": "move-concurrently",
25      "SPDXID": "SPDXRef-
26      Package-18C57CD53EF307288EAB0F87CA352B025B010578D23AAFB705E1351D18921B5",
27      "downloadLocation": "NOASSERTION",
28      "filesAnalyzed": false,
29      "licenseConcluded": "NOASSERTION",
30      "licenseDeclared": "NOASSERTION",
31      "copyrightText": "NOASSERTION",
32      "versionInfo": "1.0.1",
33      "externalRefs": [
34        {
35          "referenceCategory": "PACKAGE-MANAGER",
36          "referenceType": "purl",
37          "referenceLocator": "pkg:npm/move-concurrently@1.0.1"
38        }
39      ]
40    }
41  ]
42 }
```

圖 21 manifest.spdx.json 查看元件版本

4.3 開啟 manifest.spdx.json 檔案後，由 relationships 查看元件關係詳見圖 22



```
24328 }
24329 ],
24330 "externalDocumentRefs": [],
24331 "relationships":
24332 {
24333   "relationshipType": "DESCRIBES",
24334   "relatedSpdxElement": "SPDXRef-RootPackage",
24335   "spdxElementId": "SPDXRef-DOCUMENT"
24336 },
24337 {
24338   "relationshipType": "DEPENDS_ON",
24339   "relatedSpdxElement": "SPDXRef-Package-
B157354EFFB34603E46FD1B32E08B12DF810360D67F201F33205A7A558B28885",
24340   "spdxElementId": "SPDXRef-RootPackage"
24341 },
24342 {
24343   "relationshipType": "DEPENDS_ON",
24344   "relatedSpdxElement": "SPDXRef-
Package-559A00DD31FAC808DF0078760E341187A9CFA32AC174D7F5ADBAC4D33E8DF0BE",
24345   "spdxElementId": "SPDXRef-RootPackage"
24346 },
24347 {
24348   "relationshipType": "DEPENDS_ON",
24349   "relatedSpdxElement": "SPDXRef-Package-
CE25F292E16882FAFDA83F27E458BFCFD3D08DEEB2110218DE44717A13BEC53",
24350   "spdxElementId": "SPDXRef-RootPackage"
24351 },
24352 {
24353   "relationshipType": "DEPENDS_ON",
24354   "relatedSpdxElement": "SPDXRef-Package-
F576B3F64441FCD6C3E16F9FF4279F9948E7123B87BC7CB6FDE198ABEBB5FCB3",
24355   "spdxElementId": "SPDXRef-RootPackage"
24356 },
24357 {
24358   "relationshipType": "DEPENDS_ON",
24359   "relatedSpdxElement": "SPDXRef-Package-
D1E130B130045261DB4E2BEAFF03A0BCC52AF47A8638110B75B5F4758973B429",
24360   "spdxElementId": "SPDXRef-RootPackage"

```

圖 22 manifest.spdx.json 查看元件關係

步驟五、下載 osv-scanner 工具

5.1 下載 osv-scanner 工具

<https://github.com/google/osv-scanner/releases/latest/>

以 v1.4.0 版為例

假設環境為 Linux 64 位元一般環境，找尋 linux-amd64 選擇：

https://github.com/google/osv-scanner/releases/download/v1.4.0/osv-scanner_1.4.0_linux_amd64

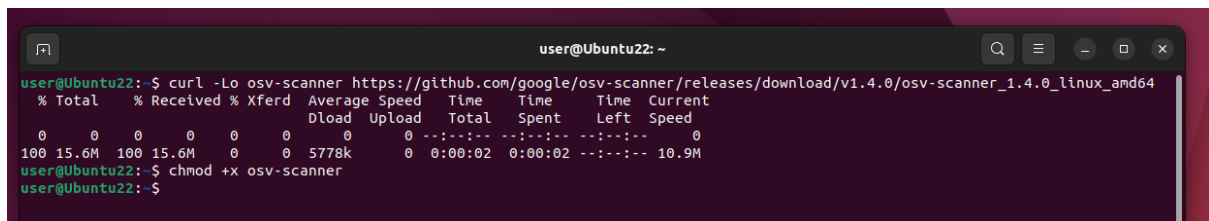
5.2 開啟 terminal 輸入指令下載 osv-scanner 工具

➤ `curl -Lo osv-scanner https://github.com/google/osv-scanner/releases/download/v1.4.0/osv-scanner_1.4.0_linux_amd64`

5.3 開啟 terminal 輸入指令設定 osv-scanner 工具權限

➤ `chmod +x osv-scanner`

完成後輸出畫面詳見圖 23



```
user@Ubuntu22: ~  
user@Ubuntu22:~$ curl -Lo osv-scanner https://github.com/google/osv-scanner/releases/download/v1.4.0/osv-scanner_1.4.0_linux_amd64  
% Total % Received % Xferd Average Speed Time Time Time Current  
 Dload Upload Total Spent Left Speed  
0 0 0 0 0 0 0 0 0:00:00 0:00:00 0:00:00 0  
100 15.6M 100 15.6M 0 0 5778k 0 0:00:02 0:00:02 --:--:-- 10.9M  
user@Ubuntu22:~$ chmod +x osv-scanner  
user@Ubuntu22:~$
```

圖 23 下載 osv-scanner 工具與設定權限的指令結果畫面

步驟六、執行 osv-scanner 工具掃描產出 json 檔，進行後續應對漏洞策略

6.1 開啟 terminal 輸入指令執行 osv-scanner 工具

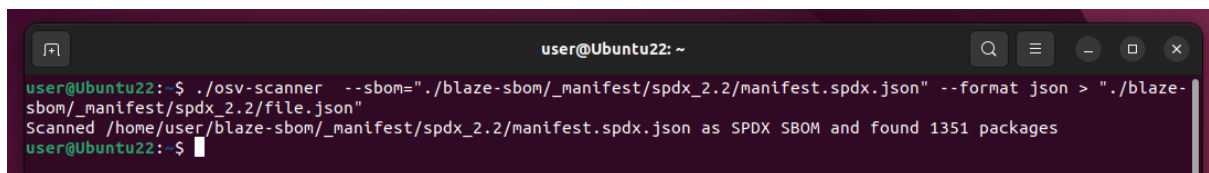
➤ `./osv-scanner --sbom="/blaze-sbom/_manifest/spdx_2.2/manifest.spdx.json" --format json > "/blaze-sbom/_manifest/spdx_2.2/file.json"`

參數說明：

`--sbom` "SBOM 檔案位置"

`--format json >` "json 檔案輸出位置"

完成後輸出畫面詳見圖 24



```
user@Ubuntu22: ~  
user@Ubuntu22:~$ ./osv-scanner --sbom="/blaze-sbom/_manifest/spdx_2.2/manifest.spdx.json" --format json > "/blaze-sbom/_manifest/spdx_2.2/file.json"  
Scanned /home/user/blaze-sbom/_manifest/spdx_2.2/manifest.spdx.json as SPDX SBOM and found 1351 packages  
user@Ubuntu22:~$
```

圖 24 執行 osv-scanner 工具指令結果畫面

6.2 開啟” /home/blaze-sbom/_manifest/spdx_2.2” 目錄，找到 file.json 檔案詳見圖 25

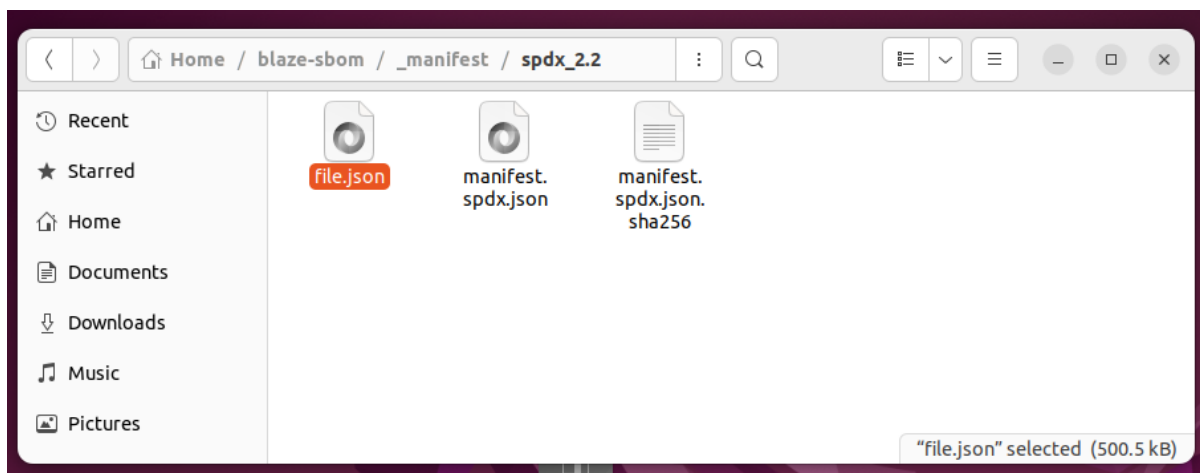


圖 25 file.json 檔案位置

步驟七、查看 file.json 檔案

開啟 file.json 檔案，文件詳細說明各元件弱點版本與 CVE 編號詳見圖 26，開始進行後續應對漏洞策略規劃

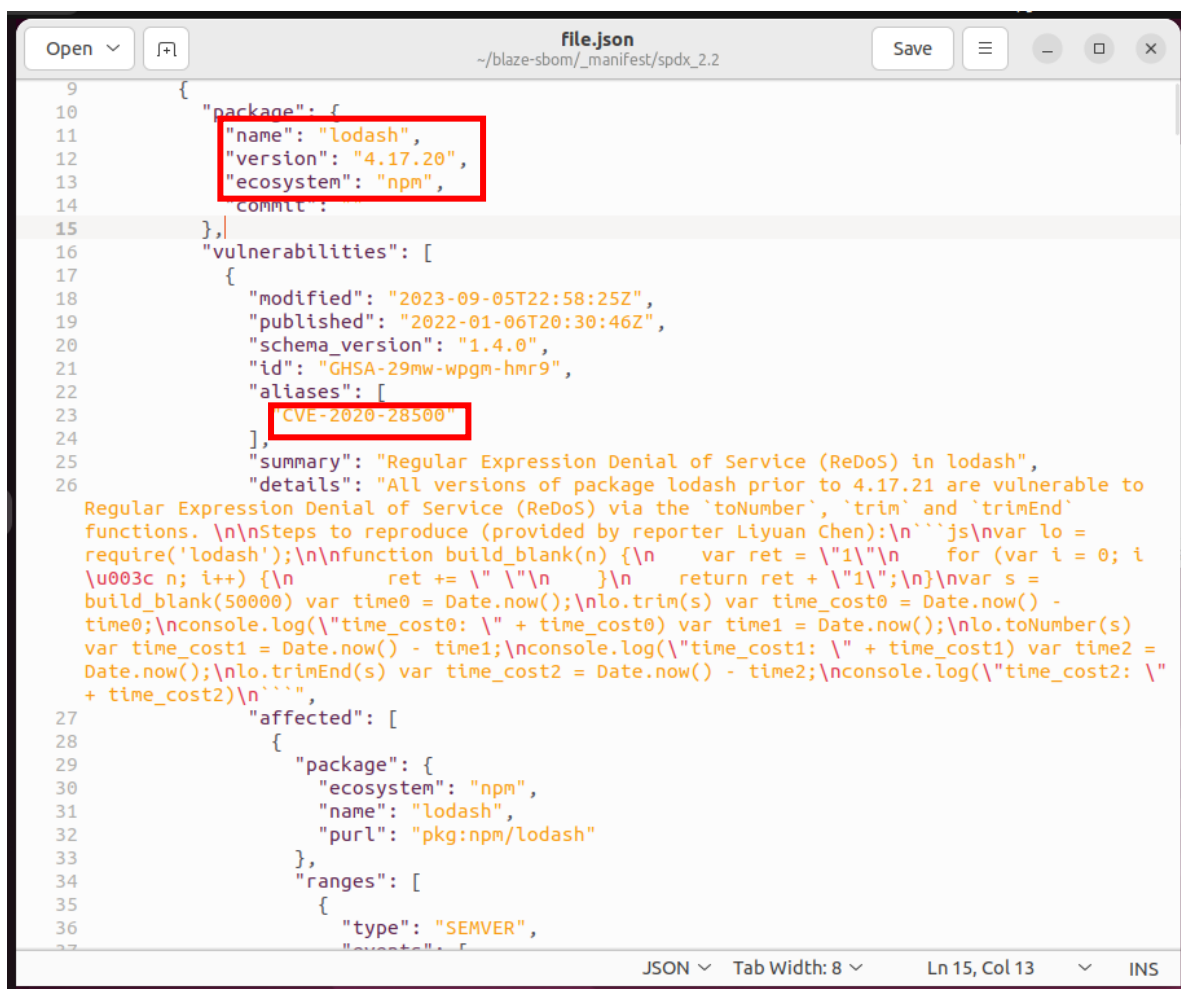


圖 26 file.json 檔案查看元件弱點版本與 CVE 編號

CycloneDX Generator 開源工具說明

CycloneDX Generator 開源工具，工具支援目前主流的 Package Managers，依照 Package Managers 與語言關係對應整理如下表：

語言	支援 Package Managers
node.js	npm-shrinkwrap.json, package-lock.json, pnpm-lock.yaml, yarn.lock, rush.js, bower.json, .min.js
java	maven (pom.xml), gradle (build.gradle, .kts), scala (sbt), bazel
php	composer.lock
python	pyproject.toml, setup.py, requirements.txt, Pipfile.lock, poetry.lock, pdm.lock, bdist_wheel, .whl, .egg-info
go	binary, go.mod, go.sum, Gopkg.lock
ruby	Gemfile.lock, gemspec
rust	binary, Cargo.toml, Cargo.lock
.Net	.csproj, packages.config, project.assets.json, packages.lock.json, .nupkg
dart	pubspec.lock, pubspec.yaml
haskell	cabal.project.freeze
elixir	mix.lock
c/c++	conan.lock, conanfile.txt
clojure	Clojure CLI (deps.edn), Leiningen (project.clj)
swift	Package.resolved, Package.swift (swiftpm)

資料彙整：<https://github.com/CycloneDX/cdxgen>

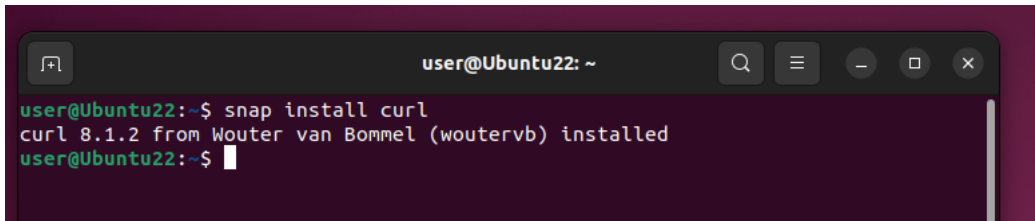
- Ubuntu Desktop 22.04 LTS 環境操作步驟

- 步驟一、安裝 curl

開啟 terminal 輸入安裝指令

- `snap install curl`

完成後輸出畫面詳見圖 27



```
user@Ubuntu22: ~  
user@Ubuntu22:~$ snap install curl  
curl 8.1.2 from Wouter van Bommel (woutervb) installed  
user@Ubuntu22:~$
```

圖 27 安裝 curl 指令結果畫面

- 步驟二、下載 cdxgen

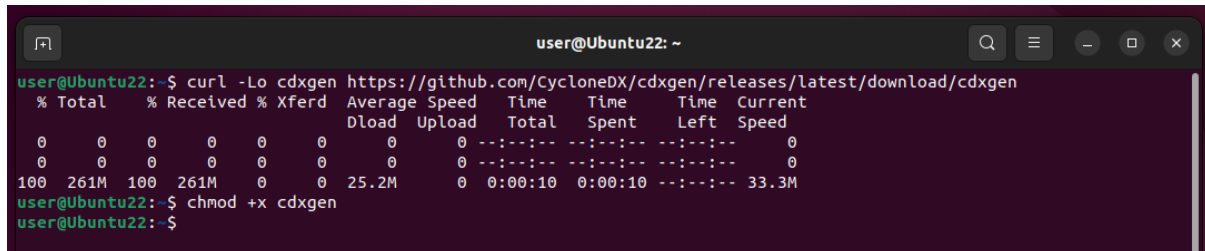
2.1 開啟 terminal 輸入指令下載 cdxgen

- `curl -Lo cdxgen https://github.com/CycloneDX/cdxgen/releases/latest/download/cdxgen`

2.2 開啟 terminal 輸入指令設定 cdxgen 權限

- `chmod +x cdxgen`

完成後輸出畫面詳見圖 28



```
user@Ubuntu22: ~  
user@Ubuntu22:~$ curl -Lo cdxgen https://github.com/CycloneDX/cdxgen/releases/latest/download/cdxgen  
% Total % Received % Xferd Average Speed Time Time Time Current  
Dload Upload Total Spent Left Speed  
0 0 0 0 0 0 0 0 0:00:00 0:00:00 0:00:00 0  
0 0 0 0 0 0 0 0 0:00:00 0:00:00 0:00:00 0  
100 261M 100 261M 0 0 25.2M 0 0:00:10 0:00:10 0:00:00 33.3M  
user@Ubuntu22:~$ chmod +x cdxgen  
user@Ubuntu22:~$
```

圖 28 下載 cdxgen 並設定權限結果畫面

步驟三、執行 cdxgen 掃描

3.1 將下載的原始檔檔案按右鍵選擇”Extract to...”進行解壓縮 zip，詳見圖 29

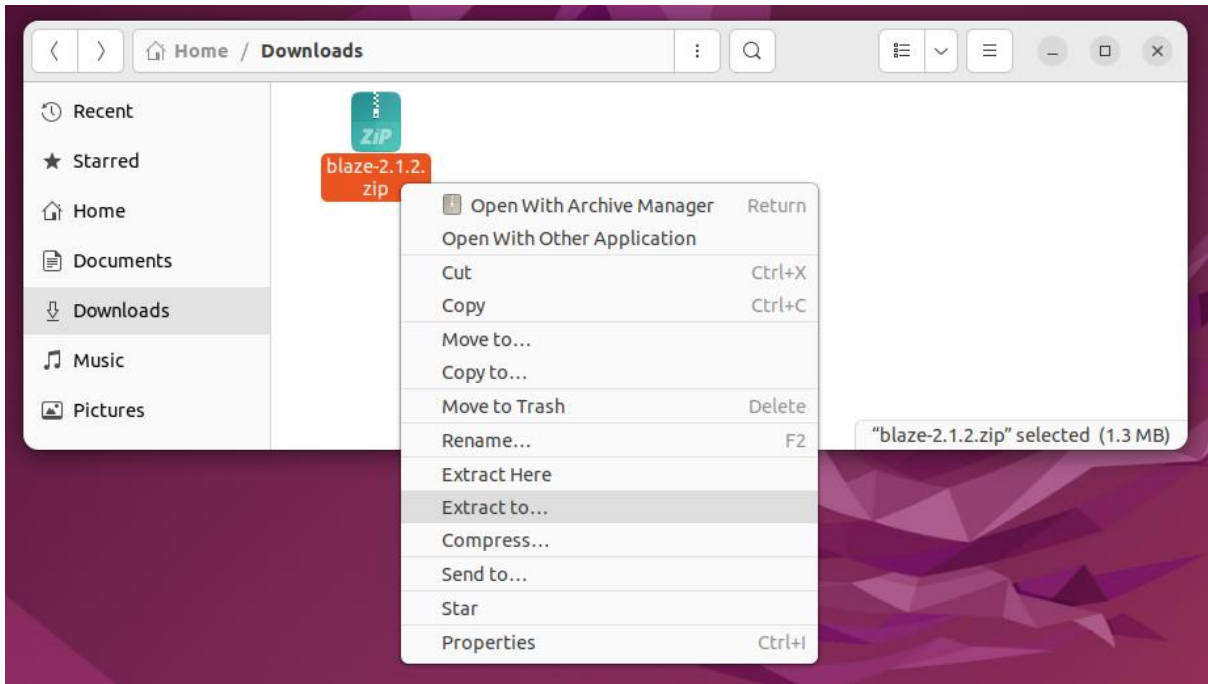


圖 29 對壓縮檔進行解壓縮操作(一)

3.2 選擇 Home 並按下 Select，詳見圖 30

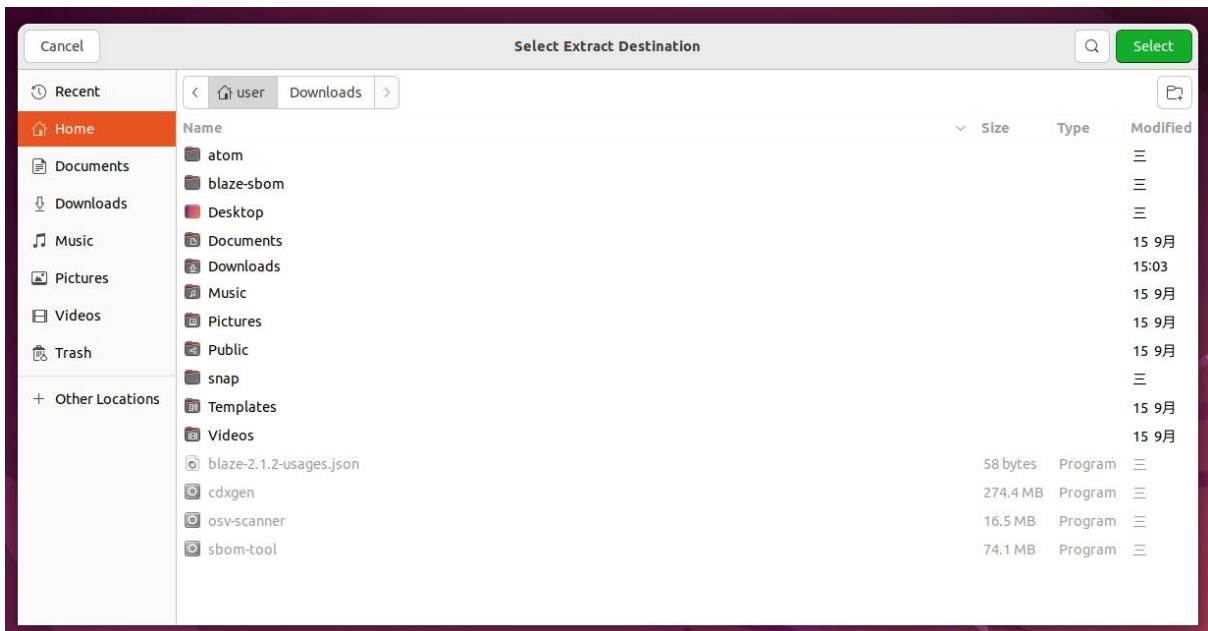


圖 30 對壓縮檔進行解壓縮操作(二)

3.3 於 Home 建立一個目錄” blaze-sbom” 放置 SBOM 檔案
完成後詳見圖 31

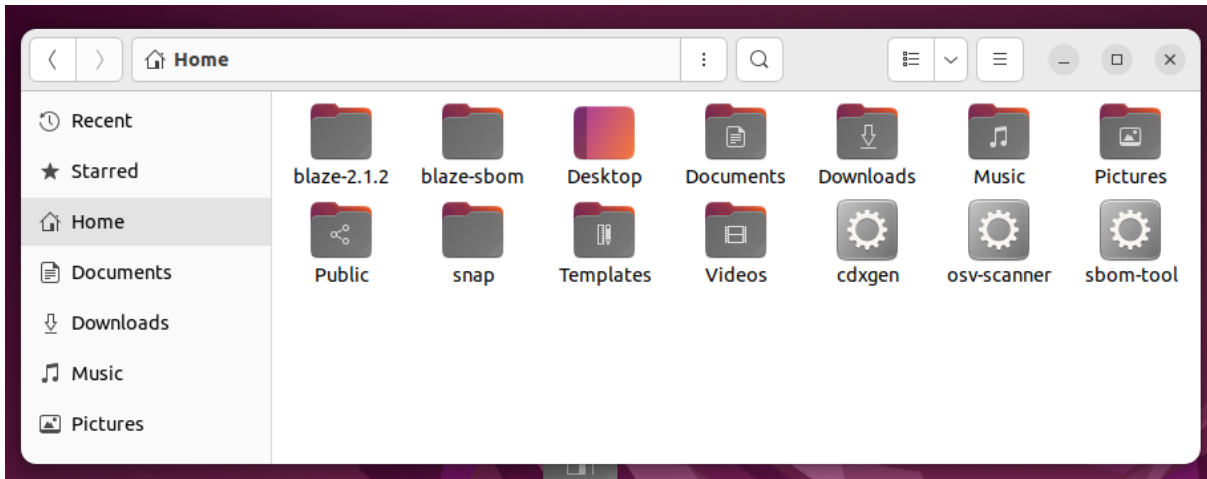


圖 31 放置原始碼與建立放置 SBOM 檔案的目錄

3.2 開啟 terminal 輸入指令執行 cdxgen

➤ `./cdxgen -r "./blaze-2.1.2" -o "./blaze-sbom/sbom.json" --spec-version 1.4`

參數說明：

-r "進行 SBOM 掃描的原始程式碼目錄位置"

-o "產生 SBOM 檔案放置位置"

--spec-version 1.4 (目前 1.4 版本支援度較完整)

完成後輸出畫面詳見圖 32

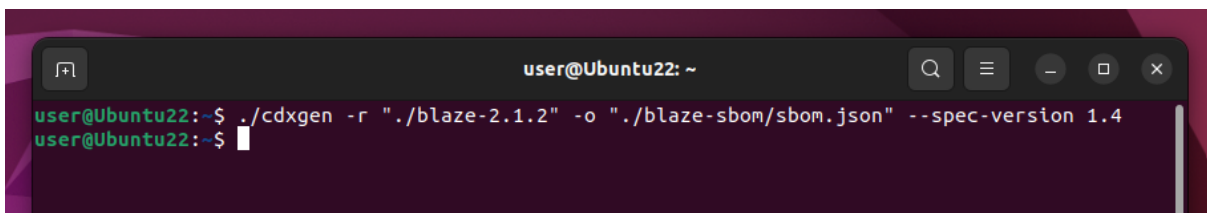


圖 32 執行 cdxgen 指令結果畫面

步驟四、查看 SBOM 檔案

4.1 "/home/blaze-sbom" 目錄，找到 sbom.json 檔案詳見圖 33

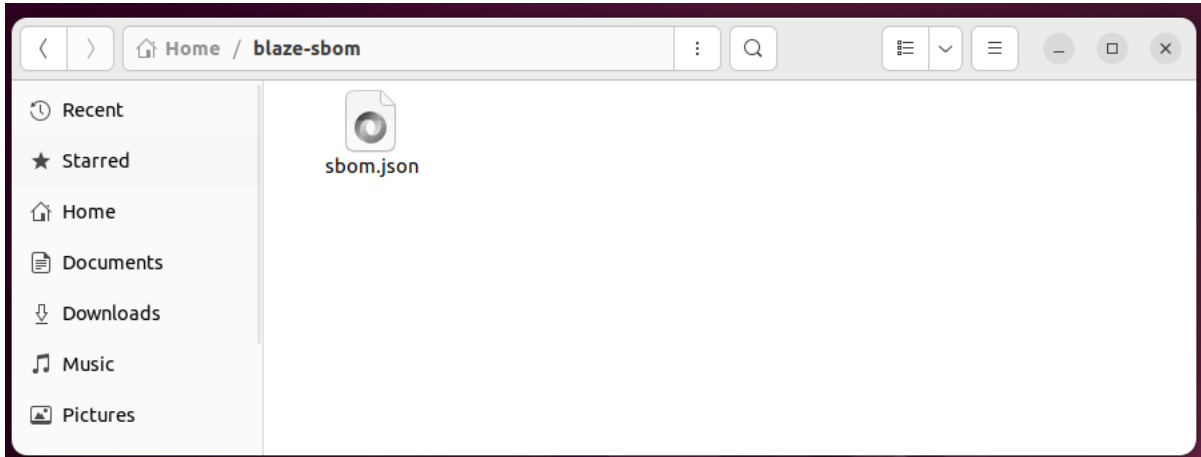


圖 33 sbom.json 檔案位置

4.2 開啟 sbom.json 檔案後，由 component 查看元件版本詳見圖 34

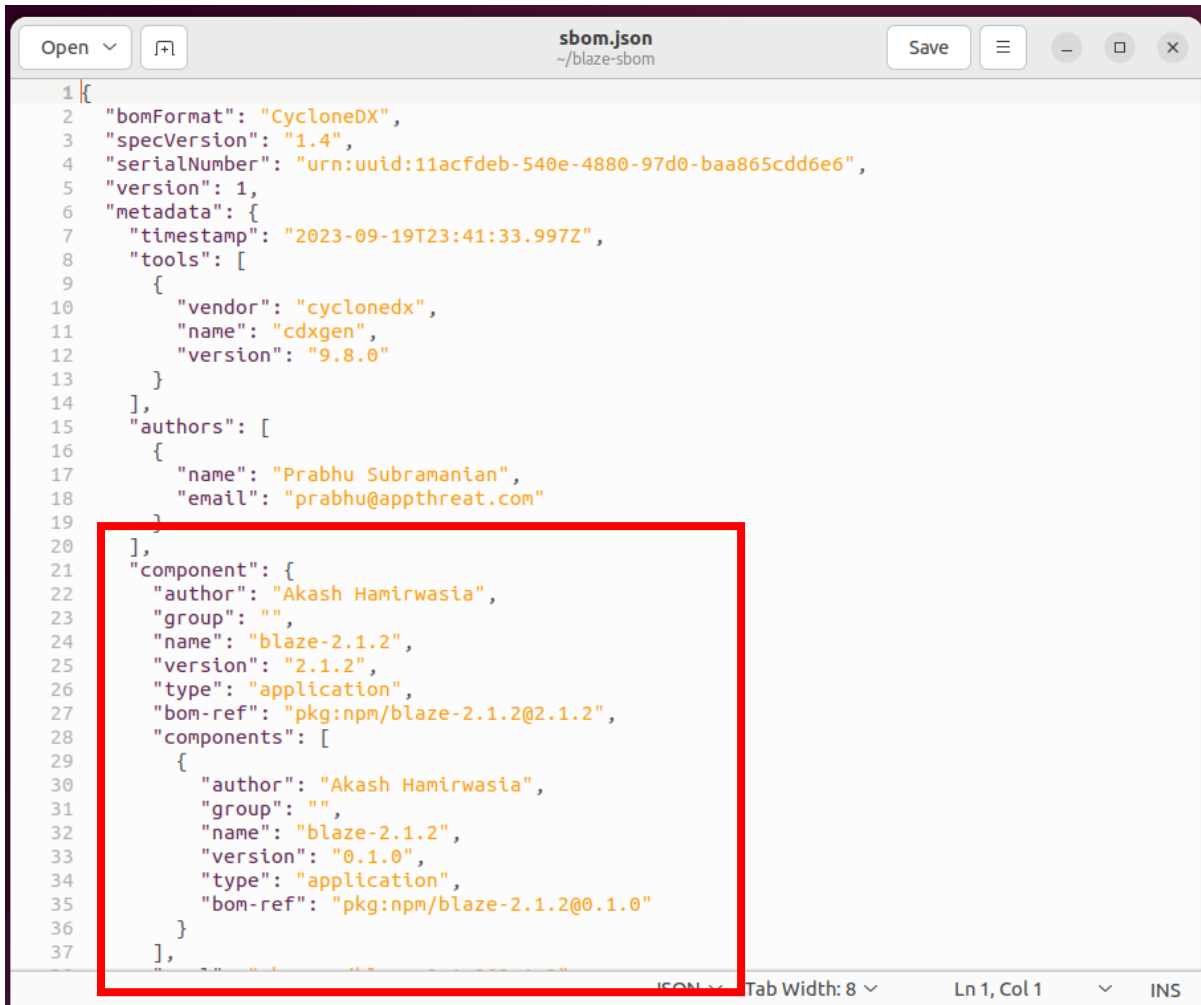


圖 34 sbom.json 查看元件版本

4.3 開啟 sbom.json 檔案後，由 ref 查看查看元件關係詳見圖 35



```
34361 },
34362 {
34363   "ref": "pkg:npm/brace-expansion@1.1.11",
34364   "dependsOn": [
34365     "pkg:npm/balanced-match@1.0.0",
34366     "pkg:npm/concat-map@0.0.1"
34367   ]
34368 },
34369 {
34370   "ref": "pkg:npm/minimatch@3.0.4",
34371   "dependsOn": [
34372     "pkg:npm/brace-expansion@1.1.11"
34373   ]
34374 },
34375 {
34376   "ref": "pkg:npm/path-is-absolute@1.0.1",
34377   "dependsOn": []
34378 },
34379 {
34380   "ref": "pkg:npm/glob@7.1.6",
34381   "dependsOn": [
34382     "pkg:npm/fs.realpath@1.0.0",
34383     "pkg:npm/inflight@1.0.6",
34384     "pkg:npm/inherits@2.0.3",
34385     "pkg:npm/inherits@2.0.4",
34386     "pkg:npm/minimatch@3.0.4",
34387     "pkg:npm/once@1.4.0",
34388     "pkg:npm/path-is-absolute@1.0.1"
34389   ]
34390 },
34391 {
34392   "ref": "pkg:npm/lines-and-columns@1.1.6",
34393   "dependsOn": []
34394 },
34395 {
34396   "ref": "pkg:npm/any-promise@1.3.0",
34397   "dependsOn": []
34398 },
34399 }
```

圖 35 sbom.json 查看元件關係

步驟五、下載 osv-scanner 工具

5.1 下載 osv-scanner 工具

<https://github.com/google/osv-scanner/releases/latest/>

以 v1.4.0 版為例

假設環境為 Linux 64 位元一般環境，找尋 linux-amd64 選擇：

https://github.com/google/osv-scanner/releases/download/v1.4.0/osv-scanner_1.4.0_linux_amd64

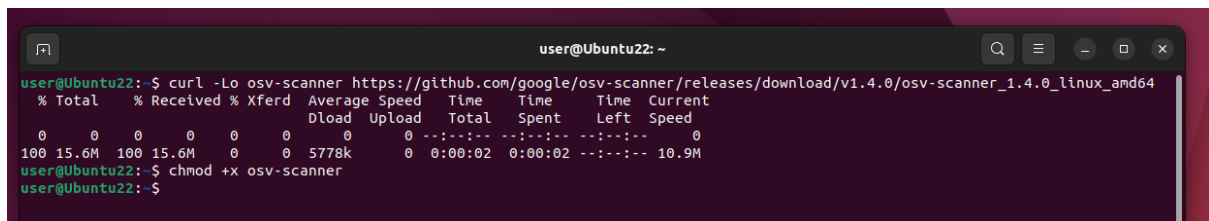
5.2 開啟 terminal 輸入指令下載 osv-scanner 工具

➤ `curl -Lo osv-scanner https://github.com/google/osv-scanner/releases/download/v1.4.0/osv-scanner_1.4.0_linux_amd64`

5.3 開啟 terminal 輸入指令設定 osv-scanner 工具權限

➤ `chmod +x osv-scanner`

完成後輸出畫面詳見圖 36



```
user@Ubuntu22: ~  
user@Ubuntu22:~$ curl -Lo osv-scanner https://github.com/google/osv-scanner/releases/download/v1.4.0/osv-scanner_1.4.0_linux_amd64  
% Total % Received % Xferd Average Speed Time Time Time Current  
 Dload Upload Total Spent Left Speed  
0 0 0 0 0 0 0 0 0:00:00 0:00:00 0:00:00 0  
100 15.6M 100 15.6M 0 0 5778k 0 0:00:02 0:00:02 --:--:-- 10.9M  
user@Ubuntu22:~$ chmod +x osv-scanner  
user@Ubuntu22:~$
```

圖 36 下載 osv-scanner 工具與設定權限的指令結果畫面

步驟六、執行 osv-scanner 工具掃描產出 json 檔，進行後續應對漏洞策略

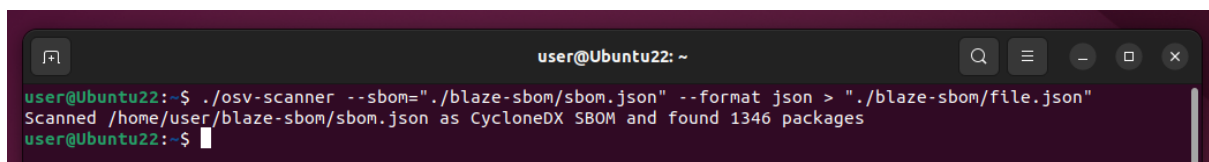
6.1 開啟 terminal 輸入指令執行 osv-scanner 工具

➤ `./osv-scanner --sbom="/blaze-sbom/sbom.json" --format json > "/blaze-sbom/file.json"`
參數說明：

`--sbom "SBOM 檔案位置"`

`--format json > "json 檔案輸出位置"`

完成後輸出畫面詳見圖 37



```
user@Ubuntu22: ~  
user@Ubuntu22:~$ ./osv-scanner --sbom="/blaze-sbom/sbom.json" --format json > "/blaze-sbom/file.json"  
Scanned /home/user/blaze-sbom/sbom.json as CycloneDX SBOM and found 1346 packages  
user@Ubuntu22:~$
```

圖 37 執行 osv-scanner 指令結果畫面

6.2 開啟” /home/blaze-sbom” 目錄，找到 file.json 檔案詳見圖 38

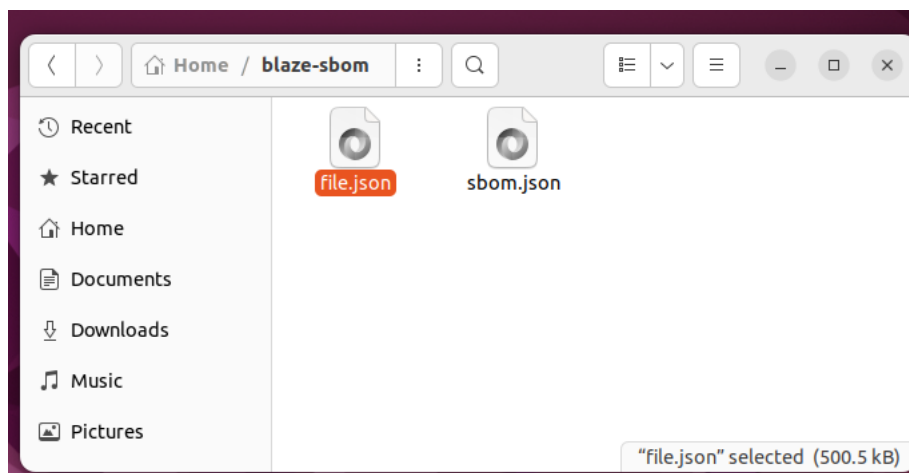


圖 38 file.json 檔案位置

步驟七、查看 file.json 檔案

開啟 file.json 檔案，文件詳細說明各元件弱點版本與 CVE 編號詳見圖 39，開始進行後續應對漏洞策略規劃

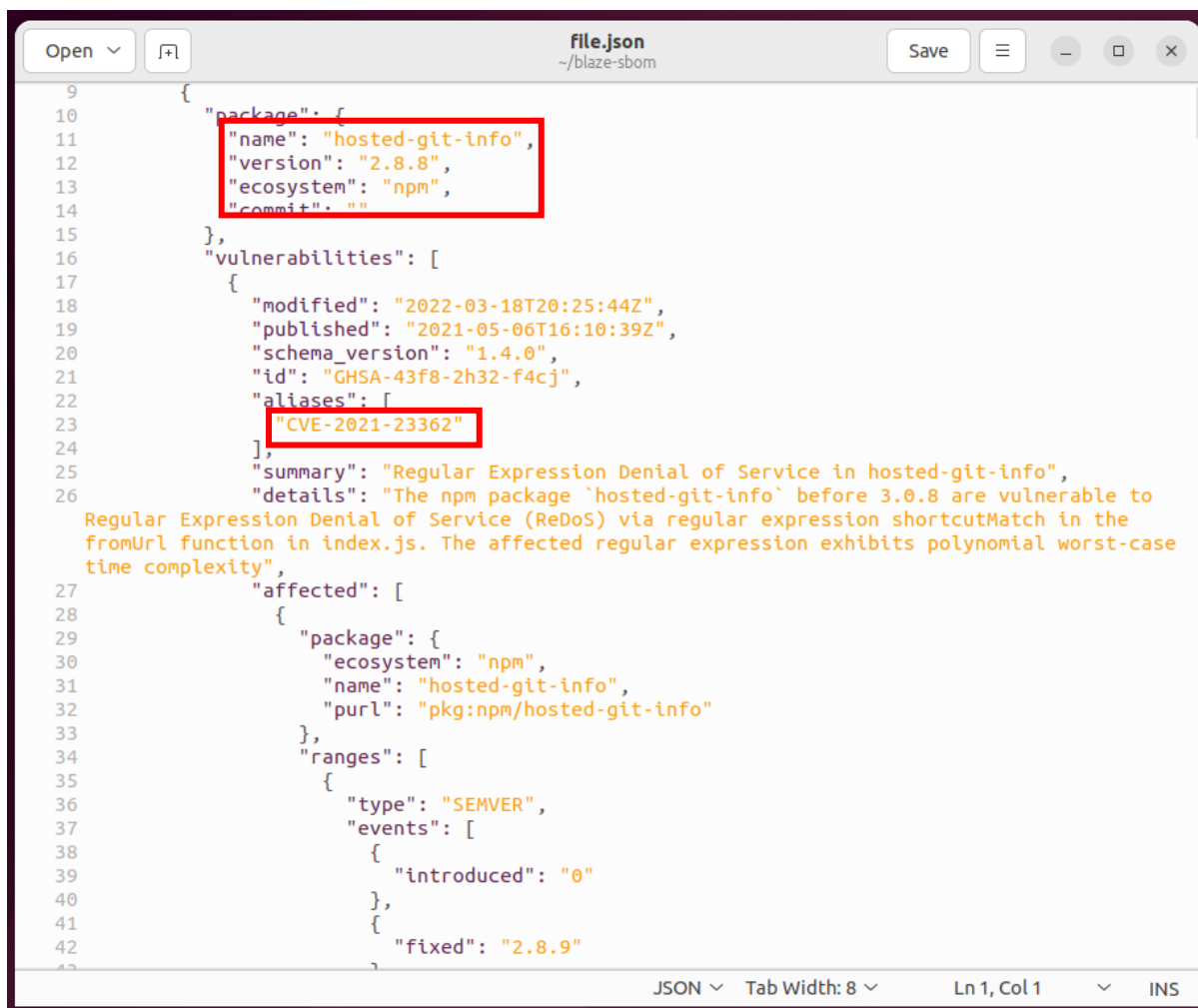


圖 39 file.json 檔案查看元件弱點版本與 CVE 編號