

Module 7: Assignment: Secure Software Supply Chain with SBOMs

This assignment consists of the generation of SBOM (Software Bill of Materials), a detailed list of components that make up a software, and vulnerability analysis based on the generated SBOM. The software assessed for vulnerabilities will be the NG911 repository, an application aligned with Next Generation 9-1-1 standards.

Part 1: SBOM Generation (in Codespaces)

This part consists of the generation of SBOM using two different tools: Syft and Trivy.

Syft is an SBOM generator that analyzes the project's files and identifies all software packages, producing an SBOM in the SPDX format. On the other hand, Trivy is a multifunction security scanner that can also generate SBOMs.

Using both tools allows comparison between formats and demonstrates how different scanners identify components in slightly different ways.

```
vscode → /workspaces/eng298-fa25-mod7-sbom-lab1/ng911-dev (main) $ syft . -o spdx-json > ../deliverables/sbom_syft_spdx.json
✓ Indexed file system
✓ Cataloged contents
  ✓ Packages [107 packages]
  ✓ Executables [0 executables]
  ✓ File digests [3 files]
  ✓ File metadata [3 locations]
[0000] WARN no explicit name and version provided for directory source, deriving artifact ID from the given path (which is not ideal)
vscode → /workspaces/eng298-fa25-mod7-sbom-lab1/ng911-dev (main) $ trivy fs . --format cyclonedx --output ../deliverables/sbom_trivy_cdx.json
2025-12-03T02:55:22Z INFO "--format cyclonedx" disables security scanning. Specify "--scanners vuln" explicitly if you want to include vulnerabilities in the "cyclonedx" report.
2025-12-03T02:55:22Z INFO [python] Licenses acquired from one or more METADATA files may be subject to additional terms. Use "--debug" flag to see all affected packages.
2025-12-03T02:55:22Z INFO [npm] To collect the license information of packages, "npm install" needs to be performed beforehand dir="test_suite/test_files/_old/TPLan_config/VS_code/node_modules"
2025-12-03T02:55:22Z INFO Number of language-specific files num=2
```

```
vscode → /workspaces/eng298-fa25-mod7-sbom-lab1/ng911-dev (main) $ ls ../deliverables/
README.md sbom_syft_spdx.json sbom_trivy_cdx.json
```

Component count between tools

Syft **108**

Trivy **106**

I noticed that Syft's SPDX SBOM identified more components than Trivy's CycloneDX SBOM. This is likely due to SPDX capturing more metadata, while CycloneDX prioritizes higher-level application dependencies.

Part 2: SBOM Vulnerability Analysis

In this part, we use Grype to scan the SBOM produced by Syft for known vulnerabilities. Grype is an open-source vulnerability scanner that checks each component against public databases of CVEs (Common Vulnerabilities and Exposures), which are standardized identifiers for publicly disclosed security flaws.

This allows us to determine whether any dependencies in the NG911 software contain known weaknesses.

```
vscode → /workspaces/eng298-fa25-mod7-sbom-lab1/ng911-dev (main) $ grype sbom:../deliverables/sbom_syft_spdx.json -o table > ../deliverables/vuln_analysis_grype.txt
✓ Vulnerability DB [updated]
✓ Scanned for vulnerabilities [9 vulnerability matches]
└─ by severity: 0 critical, 2 high, 6 medium, 1 low, 0 negligible
```

Vulnerabilities found using Grype:

```
vscode → /workspaces/eng298-fa25-mod7-sbom-lab1/ng911-dev (main) $ head -20 ../deliverables/vuln_analysis_grype.txt
NAME      INSTALLED  FIXED IN  TYPE    VULNERABILITY  SEVERITY  EPSS          RISK
cryptography  43.0.0    44.0.1    python  GHSA-79v4-65xg-pq4g  Low       1.1% (77th)  0.3
setuptools    72.1.0    78.1.1    python  GHSA-5rjg-fvgr-3xxf  High      < 0.1% (25th) < 0.1
fonttools     4.57.0    4.60.2    python  GHSA-768j-98cg-p3fv  Medium    < 0.1% (26th) < 0.1
requests      2.32.3    2.32.4    python  GHSA-9hjb-9r4m-mvj7  Medium    < 0.1% (25th) < 0.1
brotli        1.1.0     1.2.0     python  GHSA-2qfp-q593-8484  High      < 0.1% (3rd)  < 0.1
urllib3       2.2.2     2.5.0     python  GHSA-pq67-6m6q-mj2v  Medium    < 0.1% (2nd)  < 0.1
urllib3       2.2.2     2.5.0     python  GHSA-48p4-8xcf-vxj5  Medium    < 0.1% (0th)  < 0.1
cryptography  43.0.0    43.0.1    python  GHSA-h4gh-qq45-vh27  Medium    N/A          N/A
scapy         2.5.0     -         python  GHSA-cq46-m9x9-j8w2  Medium    N/A          N/A
```

Top 5 CVEs found

CVE / GHSA	Severity	Component	Installed Version	Comment
GHSA-5rjg-fvgr-3xxf	High	setuptools	72.1.0	setuptools has a path traversal vulnerability in PackageIndex.download
GHSA-2qfp-q593-8484	High	brotli	1.1.0	Scrapy is vulnerable to a denial of service (DoS)
GHSA-768j-98cg-p3fv	Medium	fonttools	4.57.0	fontTools is Vulnerable to Arbitrary File Write
GHSA-9hjb-9r4m-mvj7	Medium	requests	2.32.3	Requests vulnerable to .netrc credentials leak via malicious URLs
GHSA-pq67-6m6q-mj2v	Medium	urllib3	2.2.2	urllib3 redirects are not disabled

CVE GHSA-5rjg-fvgr-3xxf

This vulnerability is caused by insufficient sanitization of filenames in PackageIndex.download, allowing attackers to exploit path traversal and write files to arbitrary locations on the system, which can potentially lead to code execution.