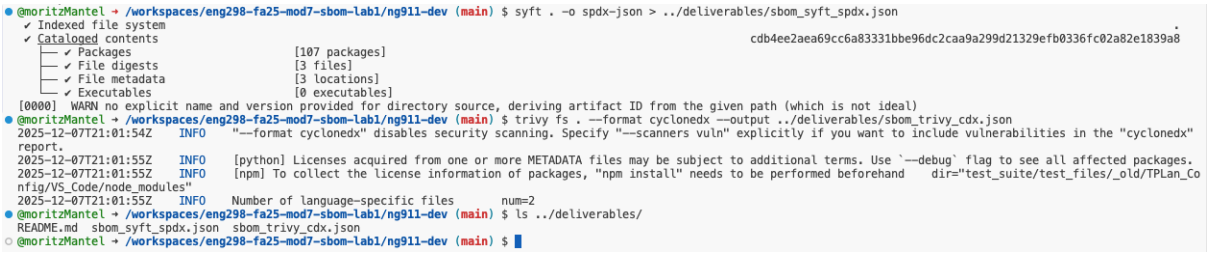
Module 7: Assignment: Secure Software Supply Chain with SBOMs

Screenshots:



Comparing Syft and Trivy Output:

Syft lists 108 packages, while Trivy shows 106 components.

Syft captures a more diverse set of relationships, including “contains”, “other” and “depends on”, while Trivy simply lists dependencies. Trivy on the other hand includes a list of vulnerabilities out of the scanned components. Even though this was empty in this case.

Grype Vulnerabilities:

CVE	Severity	Component	Version	Comment
CVE-2025-47273	High	Setuptools	72.1.0	multicast in source builds from vulnerable setuptools dependency
CVE-2025-6176	High	Brotli	1.1.0	Scrapy is vulnerable to a denial of service (DoS) attack due to flaws in brotli decompression implementation
CVE-2025-66471	High	Urllib3	2.2.2	urllib3 streaming API improperly handles highly compressed data
CVE-2025-66034	Medium	fonttools	4.57.0	fontTools is Vulnerable to Arbitrary File Write and XML injection in fontTools.varLib
CVE-2024-47081	Medium	Requests	2.32.3	Requests vulnerable to .netrc credentials leak via malicious URLs

My output of grype gave GHSA references, which I looked up manually to get the corresponding CVE.

CVE-2025-47273:

Taking a closer look at the setuptools vulnerability:
It is essentially the result of improper input sanitization, that could allow an attacker to submit a malicious url, which would means an attacker could potentially write files to arbitrary locations. This could be one of the primitives needed to escalate to remote code execution.

Reflection:

It is great to have simple and openly available tools to quickly analyse software components. While clearly important for security concerns, I can imagine it is also helpful for maintaining and shipping software.
Together with a tool like grype, which can then look up known vulnerabilities, developers have an easy and cheap way of eliminating easily exploitable security problems.
With the exception of scapy, all the vulnerabilities found by grype can be fixed by upgrading to a newer version. This is especially important for an open source project, where every attacker can run the same diagnostic, look up the known exploits and immediately take advantage of them.