Supply Chain Security

Aleksandr Tserepov-Savolainen

September 22, 2022

Outline

Problem statement

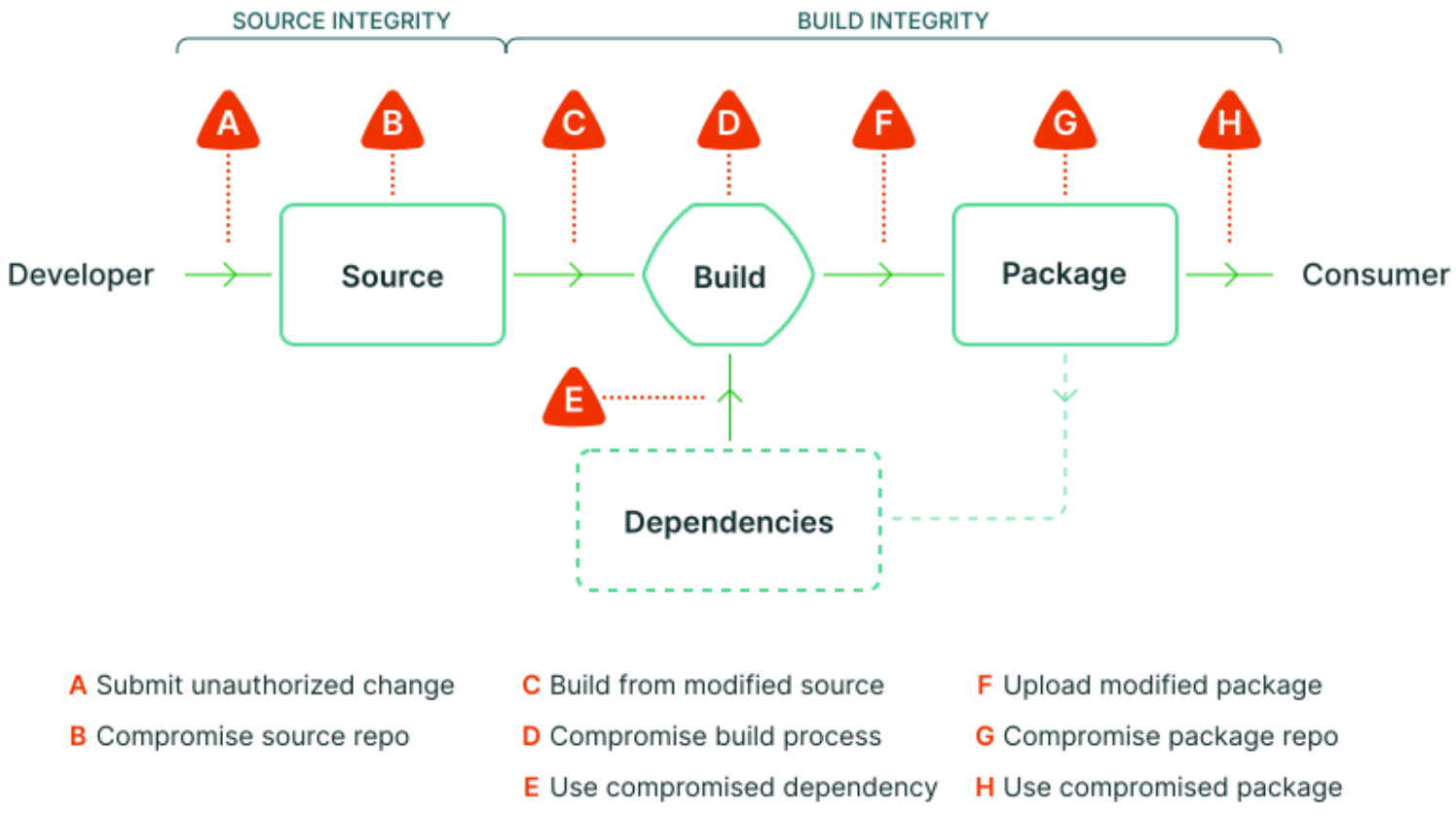


image src:
`https://slsa.dev/spec/v0.1/#supply-chain-threats`

Breach cases

TODO: Move links

1959 CIA intercepted a USSR lunar probe

`https://www.cia.gov/readingroom/collection/lunik-loan-space-age-spy-story`

2014 3rd party vendor credential leak on Home Depot's credit card terminals

`https://www.computerweekly.com/news/2240234281/Home-Depot-traces-credit-card-data-hack-to-supplier-compron`

2021 backdoor in the open source PHP Git server

`https://news-web.php.net/php.internals/113838`

SLSA Framework

Source Integrity

Ensuring every change reflects the intent of producer.

SLSA Framework

Source Integrity

Ensuring every change reflects the intent of producer.

Build Integrity

Ensuring artifacts are not modified on transit between stages of the Supply Chain.

SLSA Framework

Source Integrity

Ensuring every change reflects the intent of producer.

Build Integrity

Ensuring artifacts are not modified on transit between stages of the Supply Chain.

Availability

Ensuring that all code and change history are available for potential incident investigation.

NixOS / Spectrum Build Environment

TODO: Build environment picture
Hydra -> BinCache -> Jenkins -> Release

- — TII GitHub
- — OpenSrc locations

NixOS SLSA Solution

Hydra package signing

Binary cache package signing

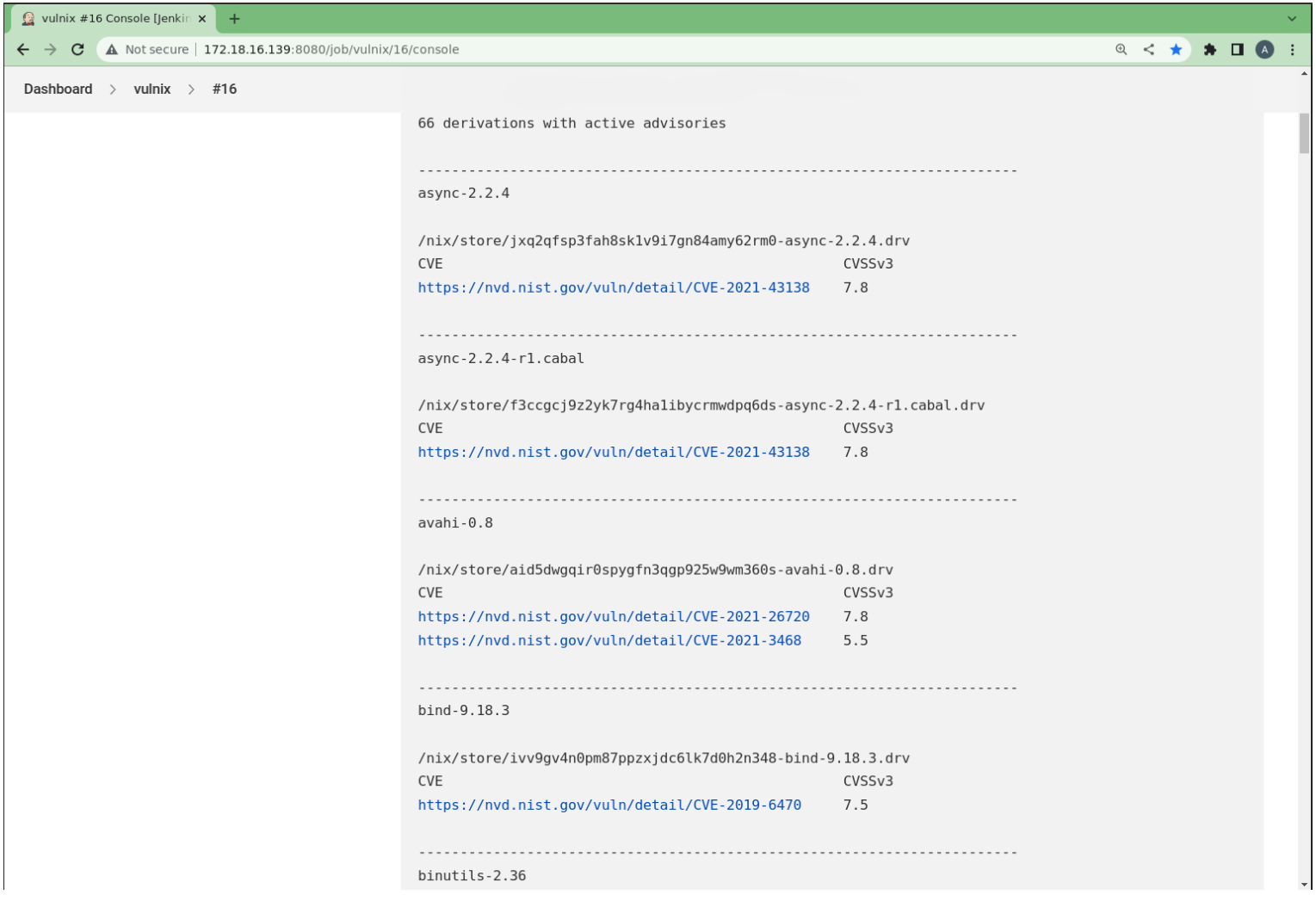
Jenkins package signature verification

SCA (Software Composition Analysis)

- Automated process that defines the open source software in the codebase.
- Companies need to be aware of potential obligations, limitations and security vulnerabilities that open source brings into play.
- As the codebase grows, tracking all of those becomes rather tricky.
- SCA takes use of automatic scanners to enable productivity without compromise on security.

Look & Feel

TODO: Vulnix results screenshot here



[Vulnix] Theory of operation

- Pulls all known CVEs from NVD
- Matches a list of derivations against CVE entries
- Whitelisting is used to suppress unwanted results

[Vulnix] Pros & Cons

[Vulnix] Pros & Cons

Pros

- Fast
- Easy integration
- Written in Python - easy to maintain

[Vulnix] Pros & Cons

Pros

- Fast
- Easy integration
- Written in Python - easy to maintain

Cons

- Simplistic mapping can lead to false positives / negatives
- Inactive development