YGREKY

Vulnerability Management at Scale for the Yocto Project

Marta Rybczynska and Samantha Jalabert

What is cve-check?

YGREKY

- Yocto Project tool for checking for known vulnerabilities
- Features
  - Fast (2-3 min overhead of a complete build)
  - Uses metadata from recipes (versions, machine-readable package name...)
  - Has an option of overrides (CVE_STATUS/CVE_IGNOREs)
  - Supports patches fixing CVEs (if tagged)
- Limitations
  - Uses package versions only, no tests if vulnerability is present
  - No support of embedded code (either copied, or using package managers)
  - Uses only the NVD database (see next slide)
- To learn more
  - OE Workshop 2023 talk “cve-check: all you wanted to know” https://www.youtube.com/watch?v=32UYr0K2PR0

The “NVD crisis”

YGREKY

- NVD stops adding new entries mid-February 2024
  - They were answering with more and more delay before
  - Little communication about the cause
  - Restarted work around May 20
  - Reasons for the slowdown **still** not clear
- API issues in December 2024
  - Hours to download the database
- 2025 situation unclear
  - NVD is an US government organizations. Budget cuts possible?
  - Less than 10% of entries processes in January (source: Tom Alrich)

Big ideas

- The dependency on NVD is a risk
  - The raw CVE database contains machine readable data from 2024
  - National/continental databases are likely to show up
- SBOM (Software Bill of Materials) contains package versions
  - Can we reuse it as a source for vulnerability checks?
- cve-check 5 years after the original build, anyone?
  - As of today, you need to keep the build directory or rebuild again
  - We only need selected information, not all that is in the build directory

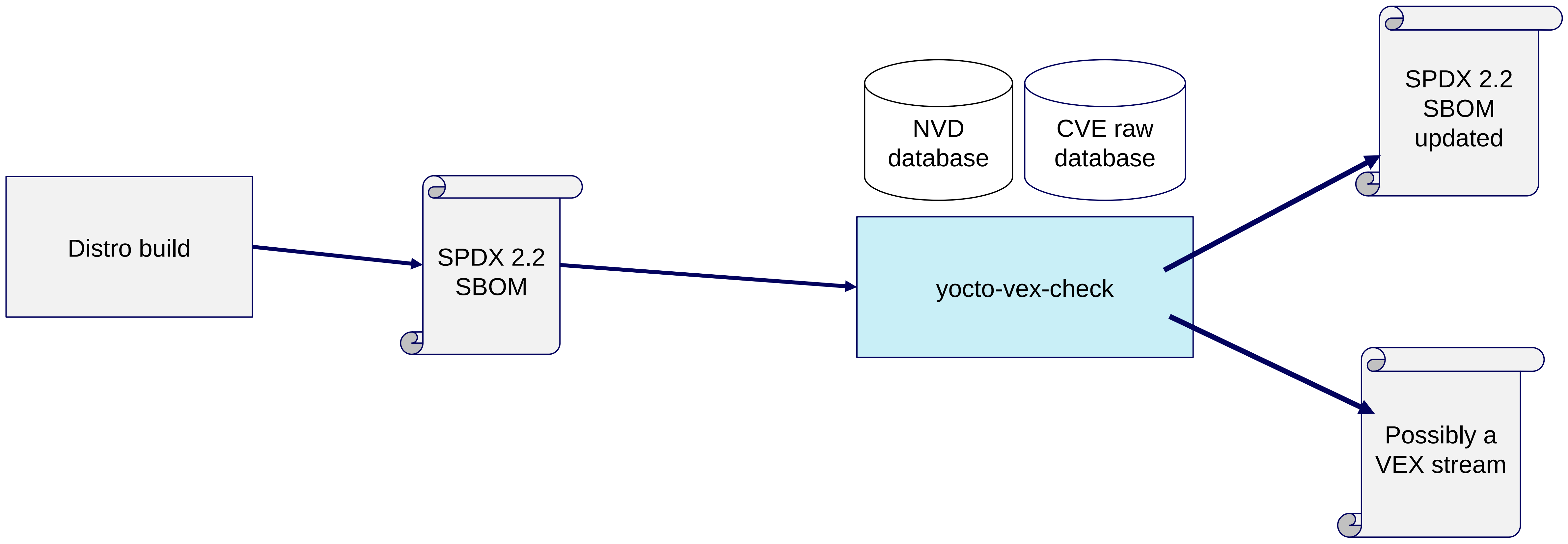
So, what is that VEX stuff?
Explaining abbreviations

- CVE (Common Vulnerability Enumeration)
  - THE vulnerability database
- SBOM (Software Bill of Materials)
  - Contains the list of package, their versions, hashes, licences....
  - Can come as SPDX or CycloneDX, the official YP (Yocto Project) supports SPDX
- VEX (Vulnerability Exchange)
  - “Annotations” for vulnerabilities
  - Allows to say: “vulnerable”, “not vulnerable because of a configuration option”, “not vulnerable because compiled out”
  - Two formats: CSAF and OpenVEX

The external checker idea

- Allow to run externally from an YP build
  - Based on data collected at build
  - Can be years later
- Pluggable architecture
  - Switch the database if we need to
  - ... or combine outputs
- YP-specific metadata
  - Overrides etc

Architecture, as we imagined



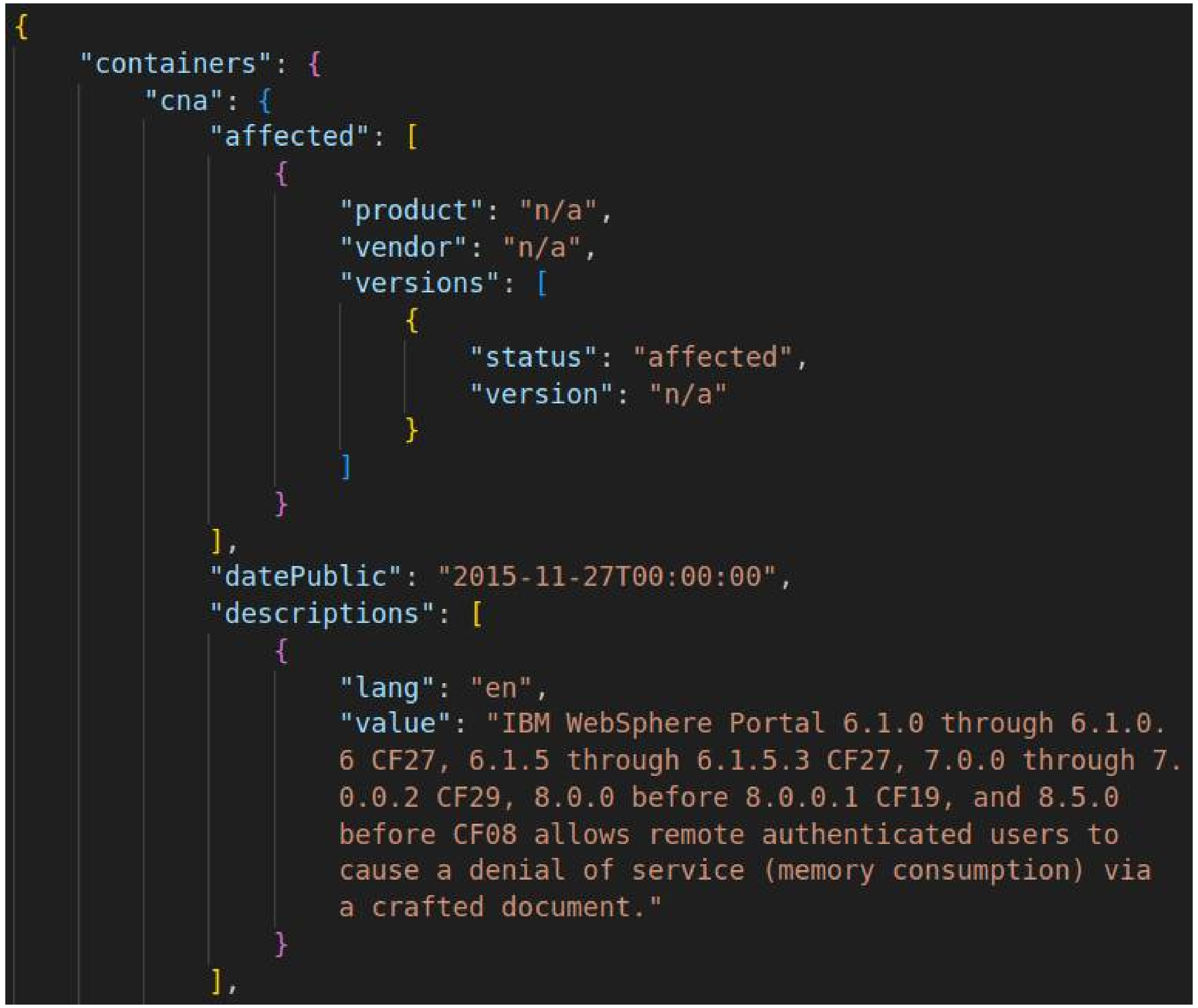
The reality check

YGREKY

- Missing SPDX generation/assembly for many cases
  - Designed for “images”
- Challenges with the raw CVE database
  - Undefined products/vendors: parsing is… complex
  - More complex description of affected versions (example: Linux kernel entries)

The reality check – CVE parsing

- CVE-2004-1599 :

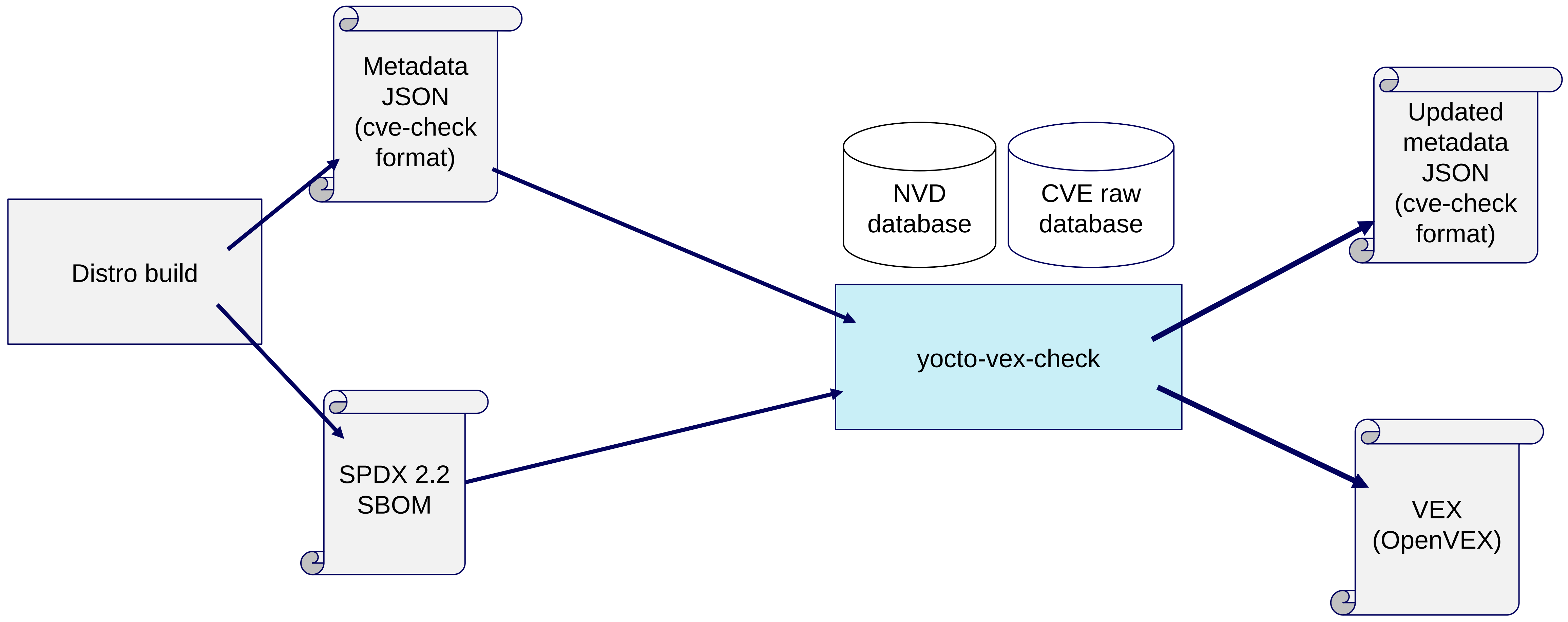


```
{
    "containers": {
        "cna": {
            "affected": [
                {
                    "product": "n/a",
                    "vendor": "n/a",
                    "versions": [
                        {
                            "status": "affected",
                            "version": "n/a"
                        }
                    ]
                }
            ],
            "datePublic": "2015-11-27T00:00:00",
            "descriptions": [
                {
                    "lang": "en",
                    "value": "IBM WebSphere Portal 6.1.0 through 6.1.0.
                    6 CF27, 6.1.5 through 6.1.5.3 CF27, 7.0.0 through 7.
                    0.0.2 CF29, 8.0.0 before 8.0.0.1 CF19, and 8.5.0
                    before CF08 allows remote authenticated users to
                    cause a denial of service (memory consumption) via
                    a crafted document."
                }
            ],
```

The reality check - Metadata

- Metadata and decision logic
    - No VEX expression for “the entry is wrong, waiting for their update” or “disputed” or “abandoned project, there will be no fix”
    - Between the override and the scanner (cve-check): what has priority?

Architecture, after the reality check



The status today

- Vex.bbclass in the Yocto Project
  - Metadata generator
  - Uses the extended format of the cve-check output
- Yocto-vex-check hosted separately
  - For now, at https://gitlab.com/syslinbit/public/yocto-vex-check/-/tree/main?ref_type=heads
  - Work pending on the overrides format (CVE format extension)
  - You choose the database on runtime
  - Generates the VEX output in the OpenVEX format

Demo: Preparations (1/3)

- Set up your YP build
  - In local.conf, add `INHERIT += “vex”`
  - Incompatible with cve-check
  - Build as usual
- Prepare databases
  - NVD: cve-update-nvd2-native.py
  - CVE: git clone git@github.com:mrybczyn/cvelistV5-overrides.git (also works with https://github.com/CVEProject/cvelistV5)

Demo: processing (2/3)

```
# ./wrap-yocto-vex-check.py \
-i ../openembedded-core/build/tmp/log/cve/cve-
summary.json \
-o out-check-test \
-t temp11-out-check \
-db ../cvelistV5-overrides/ -db-type CVE \

Copying the CVE JSON file
Converting CVE JSON to input VEX
Grabbing CVE information
CVE check finished
#
```

Demo: reviewing results (3/3)

```
# ./script/cve-report.py -s -i out-check-test/cve-
summary.json
Issues for package linux-yocto (version 6.12.9):

	Unpatched: CVE-2021-46978 CVE-2021-47089 CVE-2021-
47137 CVE-2024-26666 CVE-2024-34027 CVE-2024-35919
	Count: 6

Global issue count: 6
#
```

Future plans

- Bring yocto-vex-check to the official repo
- Seamlessly integrate with the YP build
- More testing, documentation...
- Possible new features
    - Merge analysis by multiple databases
    - More databases like OSV
    - Vendor-specific overrides

Questions?