KubeCon Japan (2025-06-16)

NTT

VexLLM: Silence Negligible CVE Alerts using LLM

https://github.com/AkihiroSuda/vexllm

Akihiro Suda (NTT)
akihiro.suda.cz@hco.ntt.co.jp

Background: too many vuln alerts

- Container image scanners such as Trivy produce an **extreme** amount of vuln alerts

- But you do NOT need to care all of them
  - You don’t actually use all the command line tools in the image
  - You don’t actually use all the functions of all the libraries in the image

- Only 3% of vulns are actually exploitable [1] [2]

Background: too many vuln alerts

- Example: `python:3.12.4` image contains a git binary with CVE-2024-32002 (9.0 CRITICAL)
  - A vuln about git submodules and symbolic links
  - No need to care, if you are sure that you won’t run git in this image
  - Even if you run git, if the target repository is hard-coded, the likelihood of compromise isn’t necessarily high
  - **No need to panic just because the CVSSv3 score is 9.0**

VEX can suppress false alerts

NTT

- **VEX: Vulnerability-Exploitability exChange**
  - List of unexploitable vulnerabilities
  - Useful for suppressing false alerts

- Several specifications
  - e.g., OpenVEX, CycloneDX, SPDX, CSAF
  - .trivyignore file can be also regarded as the simplest form of VEX (in the broadest sense)

- Maintaining a VEX file manually is quite cumbersome when you have hundreds or thousands of alerts

VexLLM

- https://github.com/AkihiroSuda/vexllm
- Let an LLM generate a **draft** of a VEX
- Hints can be provided via CLI arguments

```
vexllm generate python.json .trivyignore \
  --hint-not-server \
  --hint-compromise-on-availability \
  --hint-used-commands=python3 \
  --hint-unused-commands=git,wget,curl,apt,apt-get
```

VexLLM

- https://github.com/AkihiroSuda/vexllm
- Let an LLM generate a **draft** of a VEX
- Hints can be provided via CLI arguments

Input (Trivy's JSON)

Output

```
vexllm generate python.json .trivyignore \
  --hint-not-server \
  --hint-compromise-on-availability \
  --hint-used-commands=python3 \
  --hint-unused-commands=git,wget,curl,apt,apt-get
```

Not a server program

Service availability may matter less than information leakage and tampering

100% sure used commands

100% sure unused ones

Example output from LLM (python:3.12.4)

```
{
  "vulnId": "CVE-2024-32002",
  "exploitable": false,
  "confidence": 1.0,
  "reason": "This vulnerability is negligible because
the affected command 'git' is explicitly marked as
unused in the context of this container image. Since the
container does not utilize git, any vulnerabilities
associated with it cannot be exploited."
}
```

High/Critical alerts: 549 → 1

Don’t trust LLM too much

- No constant output
- Many false positives, as well as false negatives
- **Needs to be reviewed by humans**

Implementation details

- LangChainGo is used as an abstraction library for several LLMs (GPT, Gemini, Claude, Ollama, …)

- Ollama works with several local LLMs, including Open Source ones
  - Mistral (Apache 2.0), Phi (MIT), Qwen (Apache 2.0), …
  - 10B-class models work well even on a recent laptop

- VexLLM parses JSON outputs from LLMs
  - Hard to enforce a JSON schema by prompt engineering, especially for local LLMs
  - Tokens that violate the schema are rejected on the GGML level
  - LangChainGo needs to be patched (PR #1302)

TODO

- Quantitative evaluation of false positives and false negatives
- More tight integration with the Trivy CLI
- Allow maintaining hint files on VEX Hub repositories
- Promote adoption in various OSS communities

Wrap-up

- Not every vuln is a threat for you
- LLM can generate a **draft** of the negligible vuln list for you
- The output still has to be reviewed by humans
- Tool: https://github.com/AkihiroSuda/vexllm