Rezilion

MI·X User Guide

ABOUT THE TOOL

MI-X is an open source project aimed at effectively determining whether a local host or running container is truly exploitable to a specific vulnerability.

- Installation Requirements
- Dependencies Installation Requirements
- Install MI-X
- Execute Scanning Template
- Arguments
- Execute Scanning Example

Installation Requirements

Before installing MI-X, make sure your machine has the following:

- ✓ python version 3
- ✓ pip3
- ✓ graphviz (optional, needed only for the graph capabilities)
- ✓ xdg-utils (optional, needed only for the graph capabilities)
- ✓ openjdk with jcmd support (needed when running in container mode and the openjdk version on the container is lower than openjdk10)

Additional installation requirements:

1. Check your os distribution using the following command:
   ```
   cat /etc/os-release
   ```
2. Understand which package manager your os distribution is using:
   apt — Ubuntu, Debian
   yum — Red Hat, CentOS, Fedora, SUSE, SLES, Amazon
   apk — Alpine
3. Install the relevant packages using your os distribution package manager

Dependencies Installation Requirements

In order to execute MI-X correctly, you have to install the graphviz python module requirement using pip:

```
pip install -r requirements.txt
```

Install MI-X

1. Clone or download the project files (no compilation nor installation is required)
   ```
   git clone https://github.com/Rezilion/mi-x.git
   ```
2. Execute MI-X menu
   ```
   cd mi-x && python3 am _ i _ exploitable.py
   ```

Execute Scanning Template

Scanning command template:

```
python3 am_i_exploitable.py --vulnerability_identifier cve_yyyy_xxxx --
container True --graph True
```

Arguments

vulnerability_identifier

Specifies the vulnerability that will be checked (Not set by default).

Syntax:

- ✓ CVE-YEAR-ID — scans your system for specific vulnerability by the vulnerability cve id
- ✓ name — scans your system for specific vulnerability by the vulnerability name
- ✓ all — scans your system for all the vulnerabilities in the cves directory

If the argument is not set, a menu message will appear presenting the currently supported vulnerabilities.

container

Scans all running containers on the host (False by default).

- ✓ When running in container mode, the user will need to insert the user's password for sudo use.

container_name

Scans specific containers on the host by inserting running containers names separated by commas only (Not set by default).

- ✓ When running in container mode, the user will need to insert the user's password for sudo use.

describe

Specifies whether to see the vulnerability description or not (True by default).

graph

Specifies whether to see the validation flowchart (False by default).

help

Help to understand how to run the code

Execute Scanning Example

Scan the machine running containers for log4shell:

```
python3 am_i_exploitable.py --vulnerability_identifier log4shell --container True
```

About Rezilion

Rezilion's platform automatically secures the software you deliver to customers. Rezilion's continuous runtime analysis detects vulnerable software components on any layer of the software stack and determines their exploitability, filtering out up to 95% of identified vulnerabilities. Rezilion then automatically mitigates exploitable vulnerabilities across the SDLC, reducing vulnerability backlogs and remediation timelines from months to hours, while giving DevOps teams time back to build.

Learn more about Rezilion's software attack surface management platform at www.rezilion.com and get your 30-day free trial.