CIRCL
Computer Incident
Response Center
Luxembourg

Kunai Updates

Quentin JEROME

04/07/2024

Computer Incident Response Center Luxembourg (CIRCL)

TLP:CLEAR

Introduction

TLP:CLEAR

Brief History

Project[1] started **end-2022** as a “good first Rust project”:

12/2022 - 01/2024: worked on it under my own company
since 01/2024: joined **CIRCL** and working on the project in the context of an EU co-funded project

Why starting such a project:

- I was disappointed by **Sysmon for Linux** for many reasons
- Yet there are many good ideas in Sysmon and I think we can do much better by:
  - getting rid of XML (for configuration and events)
  - do not transpose something primarily done for Windows into Linux

[1] https://github.com/kunai-project/kunai

TLP:CLEAR

What can we do with Kunai ?

An **open-source** monitoring tool designed for threat-detection/hunting

- Monitor many **events**[2] (execve, shared object loaded, BPF programs loaded, files read/write/delete ...)
- Events comes with the following:
  - Relevant information to build solid **behavioral detections**
  - In chronological order
  - Grouping capability through a uuid
  - Parent/child tracking
  - Enriched with data from previous events (i.e. network connect/send)
- Accurately track security events generated by Linux container solutions

[2] https://why.kunai.rocks/docs/category/kunai—events

TLP:CLEAR

What’s new since last public talk ?

TLP:CLEAR

Add Missing events

- Clone
  - improves task tracking
- Prctl
  - some malware use this to change task name
- File Unlink (i.e. deletion)
  - to be able to detect crypto-lockers
- Bpf Socket Filter (used to filter specific network traffic on a socket)
  - used by BPFDoor[3] malware on a raw socket

[3] https://github.com/gwillgues/BPFDoor/blob/main/bpfdoor.c

TLP:CLEAR

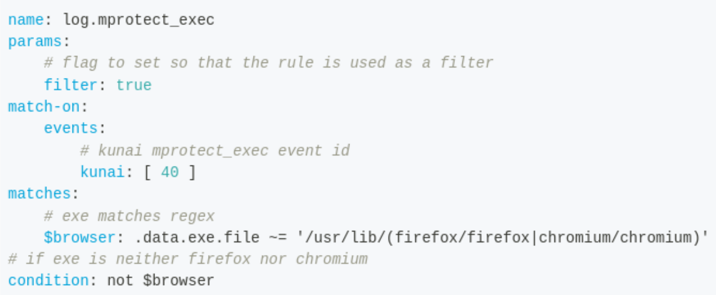
Tacking a Step Back

TLP:CLEAR

Filtering rules

Kunai generates a lot of activity, it was already possible to turn **on/off** events but there is a need for **event filtering** :

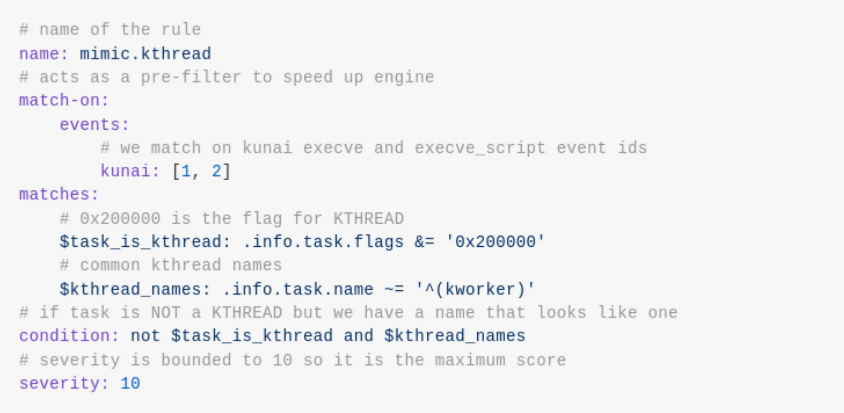
1. reduce noise
2. without context a security alert is **useless** !



```yaml
name: log.mprotect_exec
params:
    # flag to set so that the rule is used as a filter
    filter: true
match-on:
    events:
        # kunai mprotect_exec event id
        kunai: [ 40 ]
matches:
    # exe matches regex
    $browser: .data.exe.file ~= '/usr/lib/(firefox/firefox|chromium/chromium)'
# if exe is neither firefox nor chromium
condition: not $browser
```

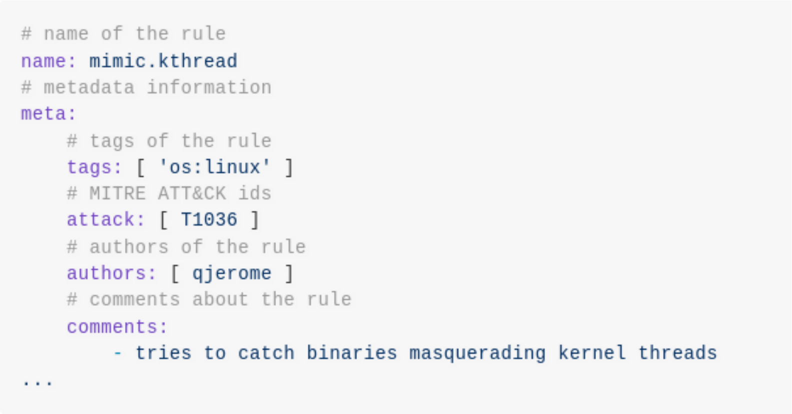
Detection rules

Addresses the need to detect a very specific pattern



```
# name of the rule
name: mimic.kthread
# acts as a pre-filter to speed up engine
match-on:
    events:
        # we match on kunai execve and execve_script event ids
        kunai: [1, 2]
matches:
    # 0x200000 is the flag for KTHREAD
    $task_is_kthread: .info.task.flags &= '0x200000'
    # common kthread names
    $kthread_names: .info.task.name ~= '^(kworker)'
# if task is NOT a KTHREAD but we have a name that looks like one
condition: not $task_is_kthread and $kthread_names
# severity is bounded to 10 so it is the maximum score
severity: 10
```

Detection rules



```
# name of the rule
name: mimic.kthread
# metadata information
meta:
    # tags of the rule
    tags: [ 'os:linux' ]
    # MITRE ATT&CK ids
    attack: [ T1036 ]
    # authors of the rule
    authors: [ qjerome ]
    # comments about the rule
    comments:
        - tries to catch binaries masquerading kernel threads
...
```

Configuration with IoCs

Kunai uses a straightforward **IoC format**

```
{"uuid": "ioc_uuid", "source":"Some IoC source", "value":"ioc_value"}
```

1. kunai perfectly know which field of its events can be an IoC
2. so it takes only a few lookups (per events) in a **hash map**

This make **IoC scanning** very fast and not depending on the number of **IoCs** being loaded

TLP:CLEAR

IoC Provider Integration

- So far it is integrated with **MISP** through the **misp-to-kunai.py** [4]
  - Can be configured to ingest **MISP feeds** (no API key needed)
  - Can be configured to export events from a given MISP instance (API key needed)
  - Able to run as a service to regularly pull updates

This script lives in a repository [5] where you can find other **tools** (mainly written in Python)

[4] misp-to-kunai.py

[5] kunai tools repository

TLP:CLEAR

Demo

TLP:CLEAR

Conclusion / Future Work

Kunai is a fairly young project but we believe it can bring added value to the Linux ecosystem and more precisely as a cheap, free and open solution to make advanced threat hunting and detection. Many improvements are foreseen to make it even more powerful:

- make it capable of executing **actions** defined in detection rules
  - kill process, dump memory, collect information ...
- embed a Yara scanning engine using the very recent **Yara-X**[6]
  - trigger scan as an **action** or to scan every file executed
- continuous integration
  - add several Linux distributions testing in CI/CD
  - monitor kernel changes impacting kunai functionalities

[6] https://github.com/VirusTotal/yara-x

TLP:CLEAR

Final Words

Q & A

Do you want to practice ?

Join workshop on friday morning ;)

References:

- Project: https://github.com/kunai-project/
- Documentation: https://why.kunai.rocks/docs/quickstart
- Tools: https://github.com/kunai-project/tools

TLP:CLEAR

Thank you all !

Kunai is an **Open-Source project** developed in the context of **NGSOTI** a co-funded project under **DEP** (Digital Europe Programme) via the **ECCC** (European Cybersecurity Competence Centre) and the **CIRCL** (Computer Incident Response Center Luxembourg).

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or ECCC. Neither the European Union nor the granting authority can be held responsible for them.

ECCC
EUROPEAN CYBERSECURITY
COMPETENCE CENTRE

Co-funded by
the European Union

TLP:CLEAR