null/OWASP combined meetup, Bangalore
14-Dec-2024

EDR INTERNALS

:Understanding the data, gaps and detection techniques

ABOUT ME:

Monika Sahu

Security Researcher at QuickHeal
Exp ~1 year
Working majorly on malware analysis and MITRE coverage.

KEY POINTS OF THIS TALK

- What is an EDR?
- What’s this topic about?
- Event Source and Raw data
- Understanding the data
- The Internals - The architecture
- Detection-working of EDR
- What the EDRs can’t capture.
- The Gaps and limitations
- The UseCase and advancement of the EDR
- The correlation.
- Conclusion

WHAT IS AN EDR?

Endpoint **D**etection and **R**esponse

- If we have AV, NGAVs then why EDRs?
- If EDRs then why XDRs and MDRs?
- Why and who should implement it?
- What magical things does the EDR has?

- Continuous real time visibility to the endpoint
- N/w, telemetry
- About the own system health and current processes.

WHAT'S THIS TOPIC ABOUT?

The data to deal with ✓

What data is useful, how the data is processed, which one to filter and which one to process

The architectural overview ✓

The basic overview and important components of an EDR. Flow of the data and interruption.

Detection technique

The magic behind detection and the fundamental core technique. ✓

The gaps or limitations

Why does the word “Bypass” exist for EDRs? ✓

EVENT SOURCE AND RAW DATA

Source of data: The Endpoints

- Different sources of the events. The Telemetry
- Integration with third party application
- Format of the raw data
- Storing and processing.

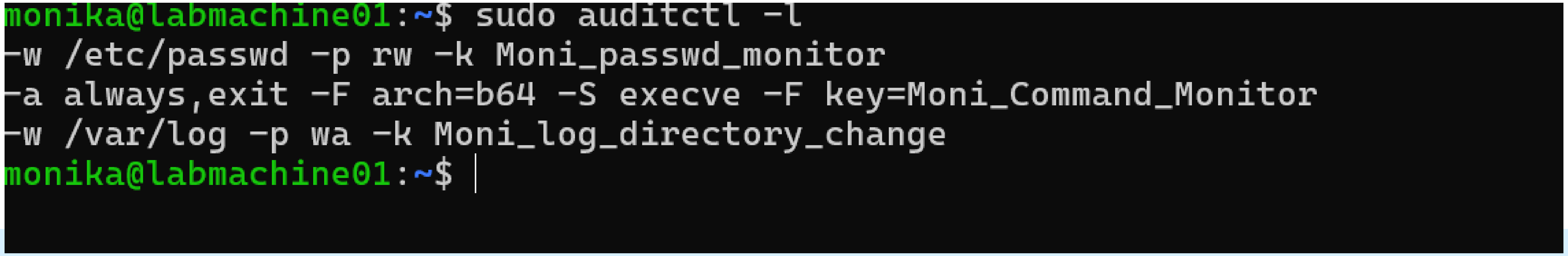
Key Components of a EDR

- Sensors/Agent
- Telemetry
- Detection

Examples of the event sources: Linux

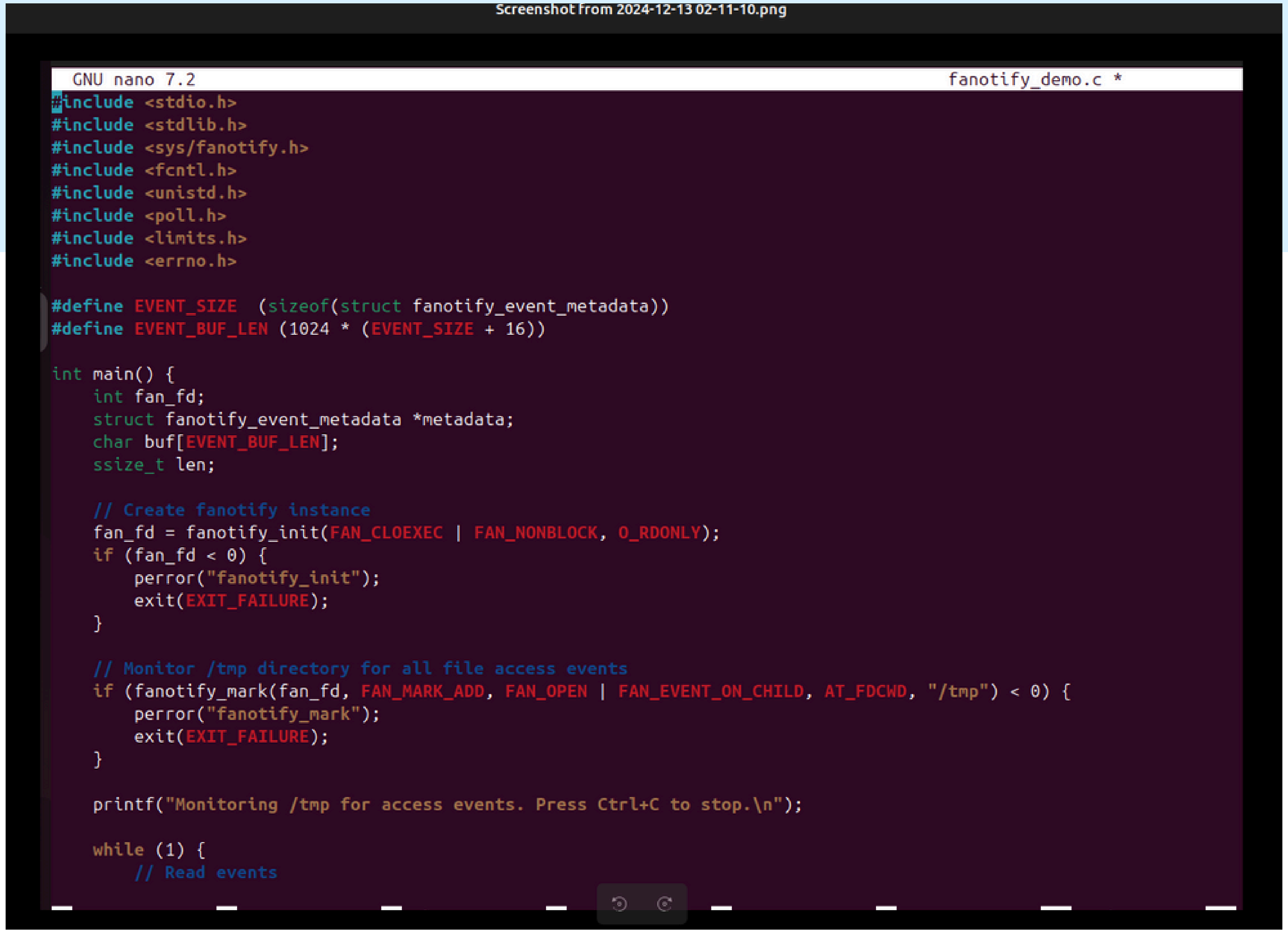
- Linux audit subsystem
- eBPF (Extended Berkeley Packet Filter)
- Inotify (inode notify) / fanotify - kernel subsystem

- Linux audit subsystem



```
monika@labmachine01:~$ sudo auditctl -l
-w /etc/passwd -p rw -k Moni_passwd_monitor
-a always,exit -F arch=b64 -S execve -F key=Moni_Command_Monitor
-w /var/log -p wa -k Moni_log_directory_change
monika@labmachine01:~$
```

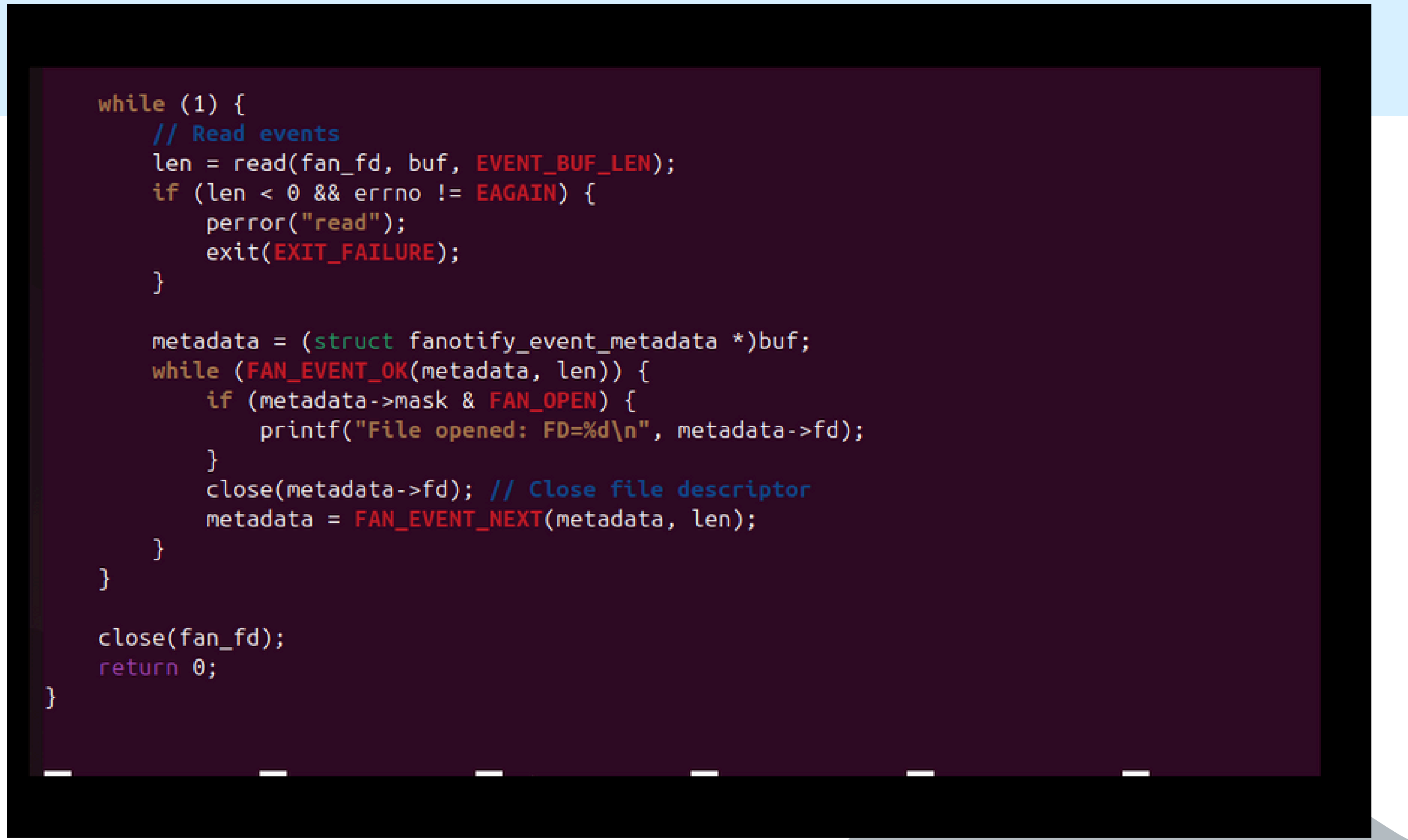
- Inotify (inode notify) / fanotify - kernel subsystem



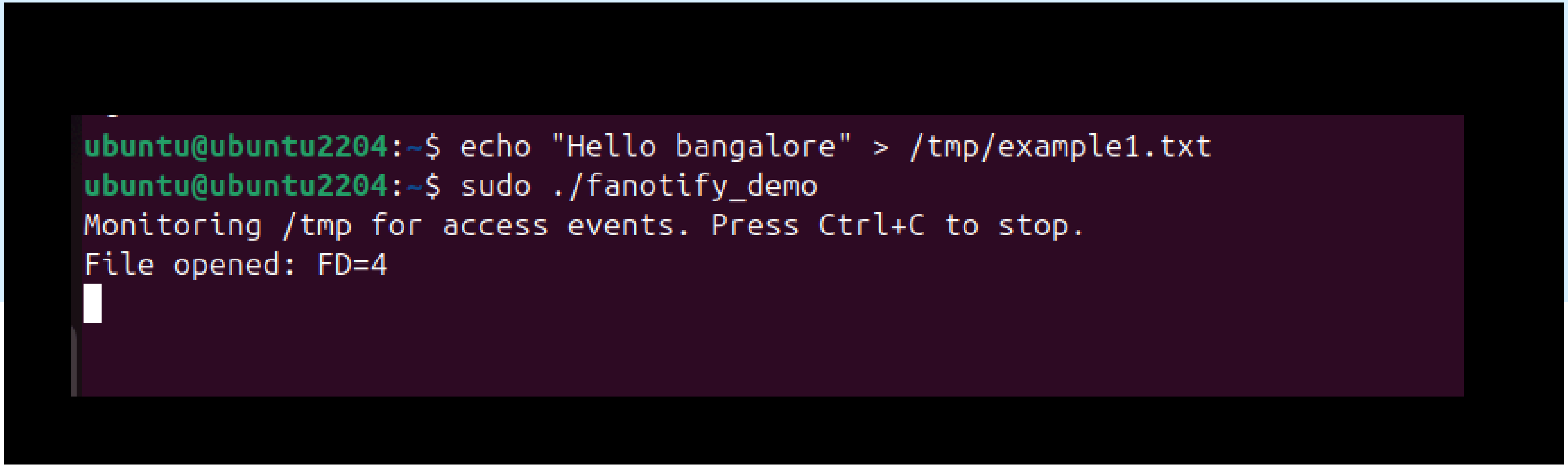
Screenshot from 2024-12-13 02-11-10.png

GNU nano 7.2 fanotify_demo.c *

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/fanotify.h>
#include <fcntl.h>
#include <unistd.h>
#include <poll.h>
#include <limits.h>
#include <errno.h>

#define EVENT_SIZE  (sizeof(struct fanotify_event_metadata))
#define EVENT_BUF_LEN (1024 * (EVENT_SIZE + 16))

int main() {
    int fan_fd;
    struct fanotify_event_metadata *metadata;
    char buf[EVENT_BUF_LEN];
    ssize_t len;

    // Create fanotify instance
    fan_fd = fanotify_init(FAN_CLOEXEC | FAN_NONBLOCK, O_RDONLY);
    if (fan_fd < 0) {
        perror("fanotify_init");
        exit(EXIT_FAILURE);
    }

    // Monitor /tmp directory for all file access events
    if (fanotify_mark(fan_fd, FAN_MARK_ADD, FAN_OPEN | FAN_EVENT_ON_CHILD, AT_FDCWD, "/tmp") < 0) {
        perror("fanotify_mark");
        exit(EXIT_FAILURE);
    }

    printf("Monitoring /tmp for access events. Press Ctrl+C to stop.\n");

    while (1) {
        // Read events
```



```
    while (1) {
        // Read events
        len = read(fan_fd, buf, EVENT_BUF_LEN);
        if (len < 0 && errno != EAGAIN) {
            perror("read");
            exit(EXIT_FAILURE);
        }

        metadata = (struct fanotify_event_metadata *)buf;
        while (FAN_EVENT_OK(metadata, len)) {
            if (metadata->mask & FAN_OPEN) {
                printf("File opened: FD=%d\n", metadata->fd);
            }
            close(metadata->fd); // Close file descriptor
            metadata = FAN_EVENT_NEXT(metadata, len);
        }
    }

    close(fan_fd);
    return 0;
}
```

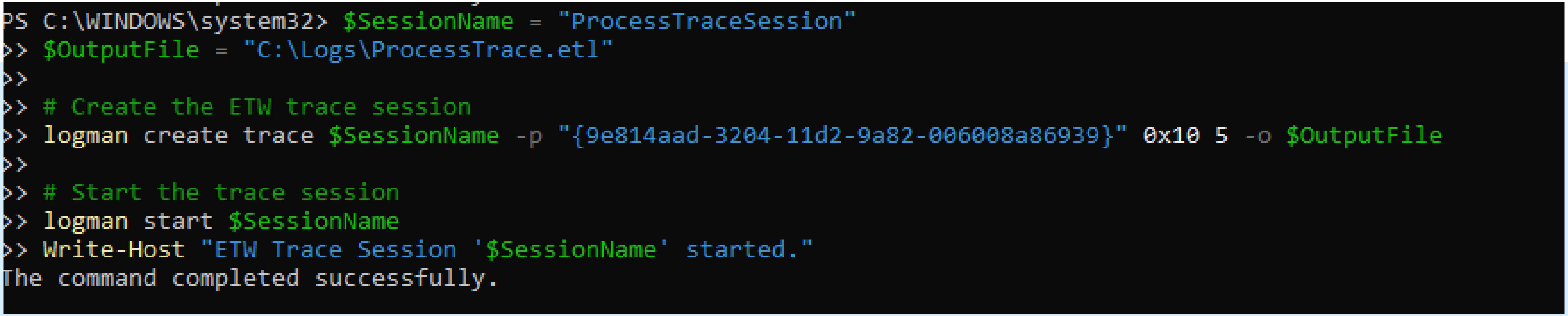
- Inotify (inode notify) / fanotify - kernel subsystem
  - 

```
ubuntu@ubuntu2204:~$ echo "Hello bangalore" > /tmp/example1.txt
ubuntu@ubuntu2204:~$ sudo ./fanotify_demo
Monitoring /tmp for access events. Press Ctrl+C to stop.
File opened: FD=4
```

Examples of the event sources: Windows

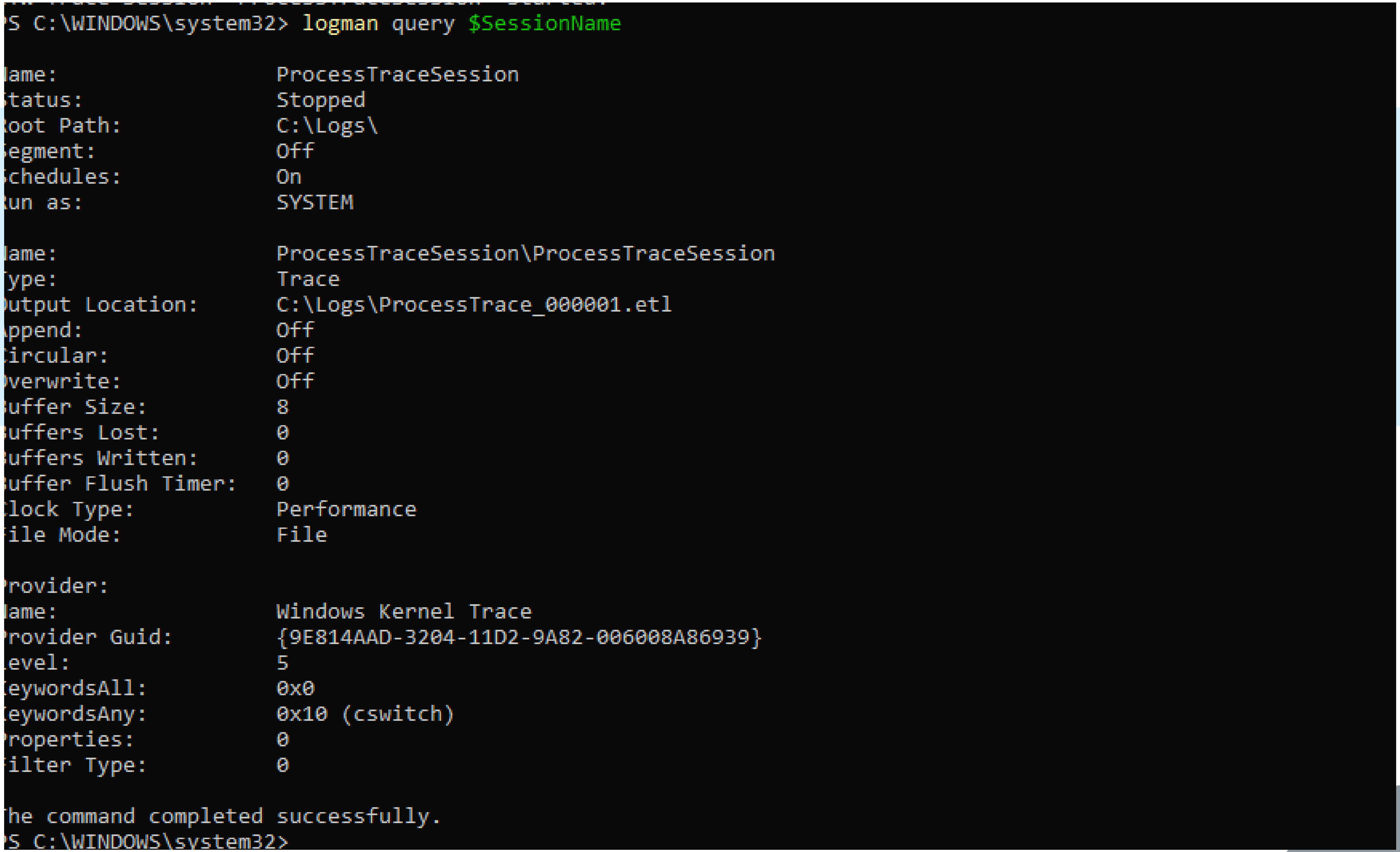
- ETW (Event Tracing for Windows)
    - Start-EtwTraceSession
    - Stop-EtwTraceSession
- Event Viewer (For GUI)
- commands: Get-EventLog
    - Get-WinEvent
- Windows Detours.
- Windows Subsystem for Linux (WSL)

ETW (Event Tracing for Windows)



```
PS C:\WINDOWS\system32> $SessionName = "ProcessTraceSession"
>> $OutputFile = "C:\Logs\ProcessTrace.etl"
>>
>> # Create the ETW trace session
>> logman create trace $SessionName -p "{9e814aad-3204-11d2-9a82-006008a86939}" 0x10 5 -o $OutputFile
>>
>> # Start the trace session
>> logman start $SessionName
>> Write-Host "ETW Trace Session '$SessionName' started."
The command completed successfully.
```

ETW (Event Tracing for Windows)



```
'S C:\WINDOWS\system32> logman query $SessionName

lame:                ProcessTraceSession
tatus:               Stopped
oot Path:            C:\Logs\
egment:              Off
chedules:            On
un as:               SYSTEM

lame:                ProcessTraceSession\ProcessTraceSession
ype:                 Trace
utput Location:      C:\Logs\ProcessTrace_000001.etl
ppend:               Off
ircular:             Off
verwrite:            Off
uffer Size:          8
uffers Lost:         0
uffers Written:      0
uffer Flush Timer:   0
lock Type:           Performance
ile Mode:            File

rovider:
lame:                Windows Kernel Trace
rovider Guid:        {9E814AAD-3204-11D2-9A82-006008A86939}
evel:                5
eywordsAll:          0x0
eywordsAny:          0x10 (cswitch)
roperties:           0
ilter Type:          0

he command completed successfully.
'S C:\WINDOWS\system32>
```

UNDERSTANDING THE DATA

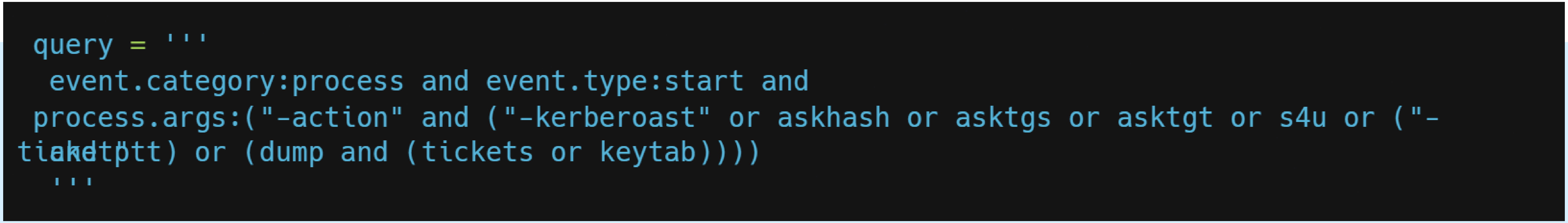
Brittle VS Robust

- **Brittle design**
  - False positive is very less. False -ve is high.
- **Robust way**
  - False +ve high, false -ve will less.
- **Hybrid approach**

Examples:

Brittle:

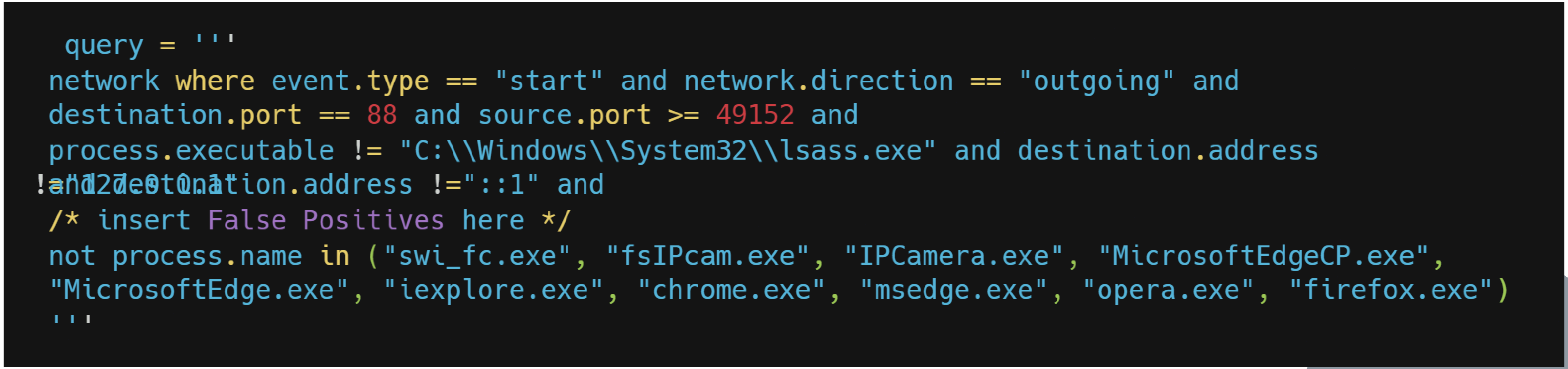
The attacker may try to rename and recompile the names



```
query = '''
 event.category:process and event.type:start and
process.args:("-action" and ("-kerberoast" or askhash or asktgs or asktgt or s4u or ("-
ticket" and ptt) or (dump and (tickets or keytab))))
 '''
```

Robust:

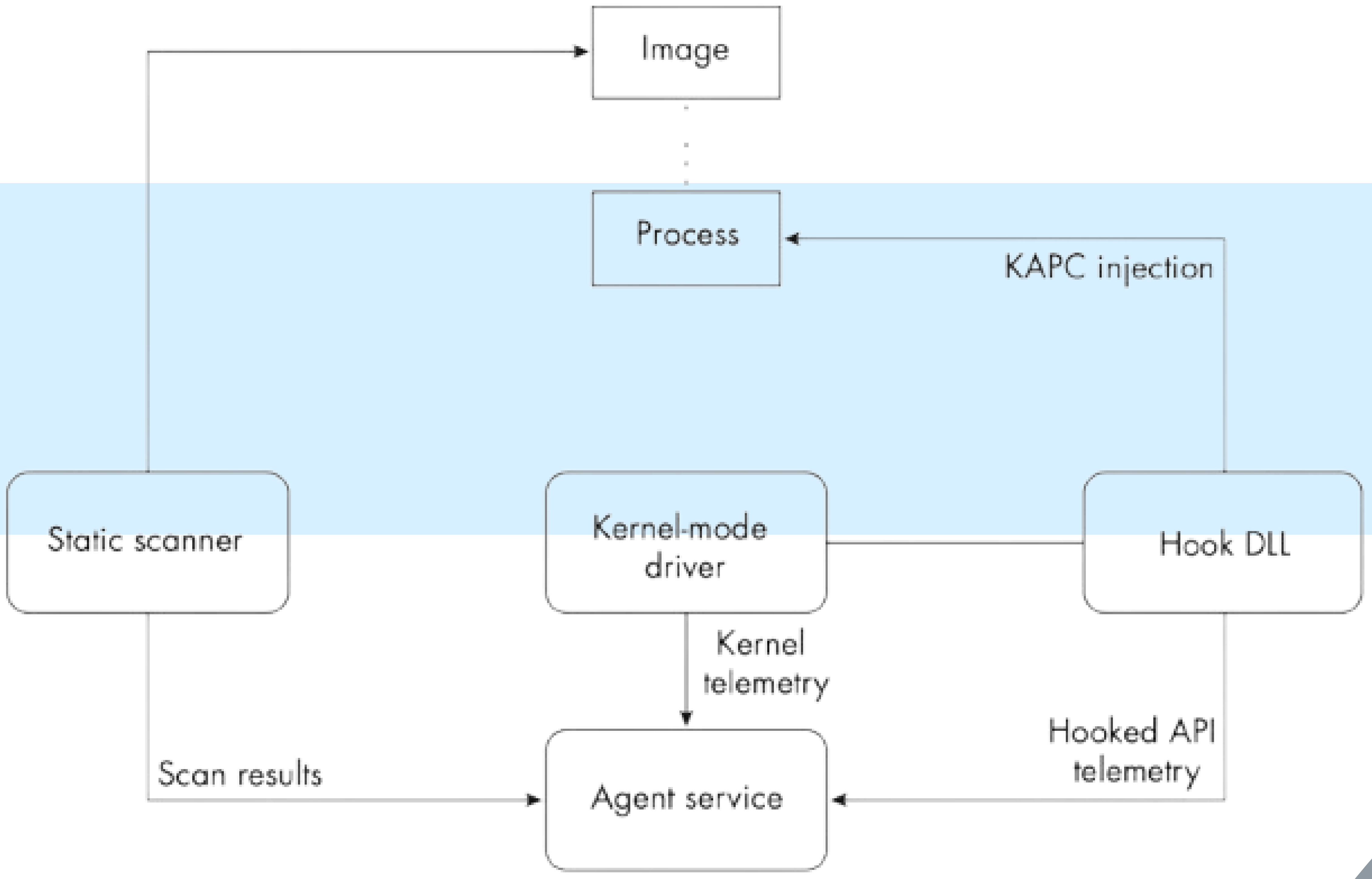
TCP port 88, the standard Kerberos port



```
 query = '''
network where event.type == "start" and network.direction == "outgoing" and
destination.port == 88 and source.port >= 49152 and
process.executable != "C:\\Windows\\System32\\lsass.exe" and destination.address
!="127.0.0.1" and destination.address !="::1" and
/* insert False Positives here */
not process.name in ("swi_fc.exe", "fsIPcam.exe", "IPCamera.exe", "MicrosoftEdgeCP.exe",
"MicrosoftEdge.exe", "iexplore.exe", "chrome.exe", "msedge.exe", "opera.exe", "firefox.exe")
'''
```

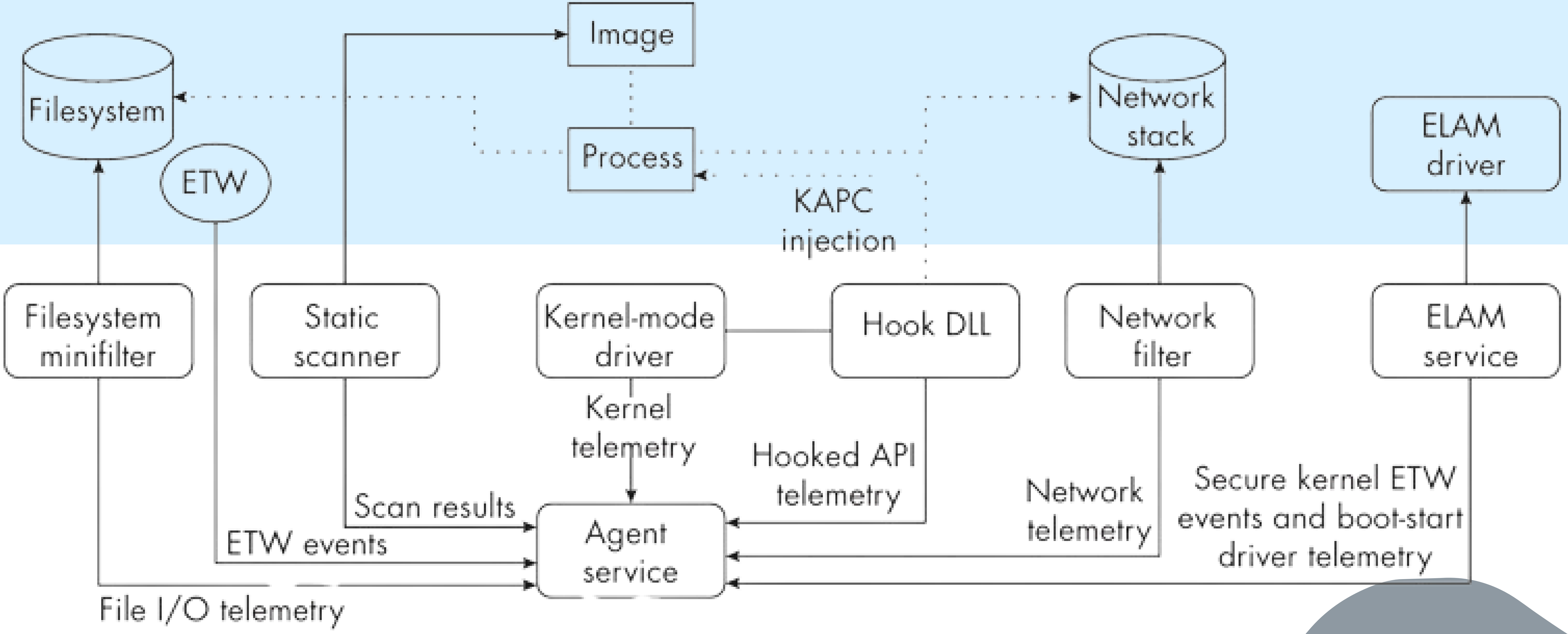
INTERNALS - ARCHITECTURE

Level - 1 Basic Architecture of EDR



INTERNALS - ARCHITECTURE

Level - 2 Intermediate Architecture of EDR



INTERNALS - ARCHITECTURE

Level - 3 Advanced Architecture of EDR

What else can make the EDR Advanced!

The best approach for securing endpoints is “multi-layered"

DETECTION-WORKING OF EDR

Detection Logics

- What is a detection logic?
- Why they are needed?
- How they are written?
- A good detection logic

WHAT THE EDRS CAN'T CAPTURE

- Encrypted and Obfuscated content.
- Memory and Kernel Level data
- Network outside the range.
- Data out of detection logic.
- Tricks used by Attackers

THE GAPS & LIMITATIONS

- Why does the word “EDR Bypass” even exist?
- Where are we lagging?
- Eg: Renaming the arguments in the source code like-

  changing ***-action*** to ***-dothis***
- Some common strategies which leads to bypass.

THE USECASE AND ADVANCEMENT OF THE EDR

- How we can enhance the security?
- Practices nowadays for enhancement.
- How AI/ML has solved many of the problems.

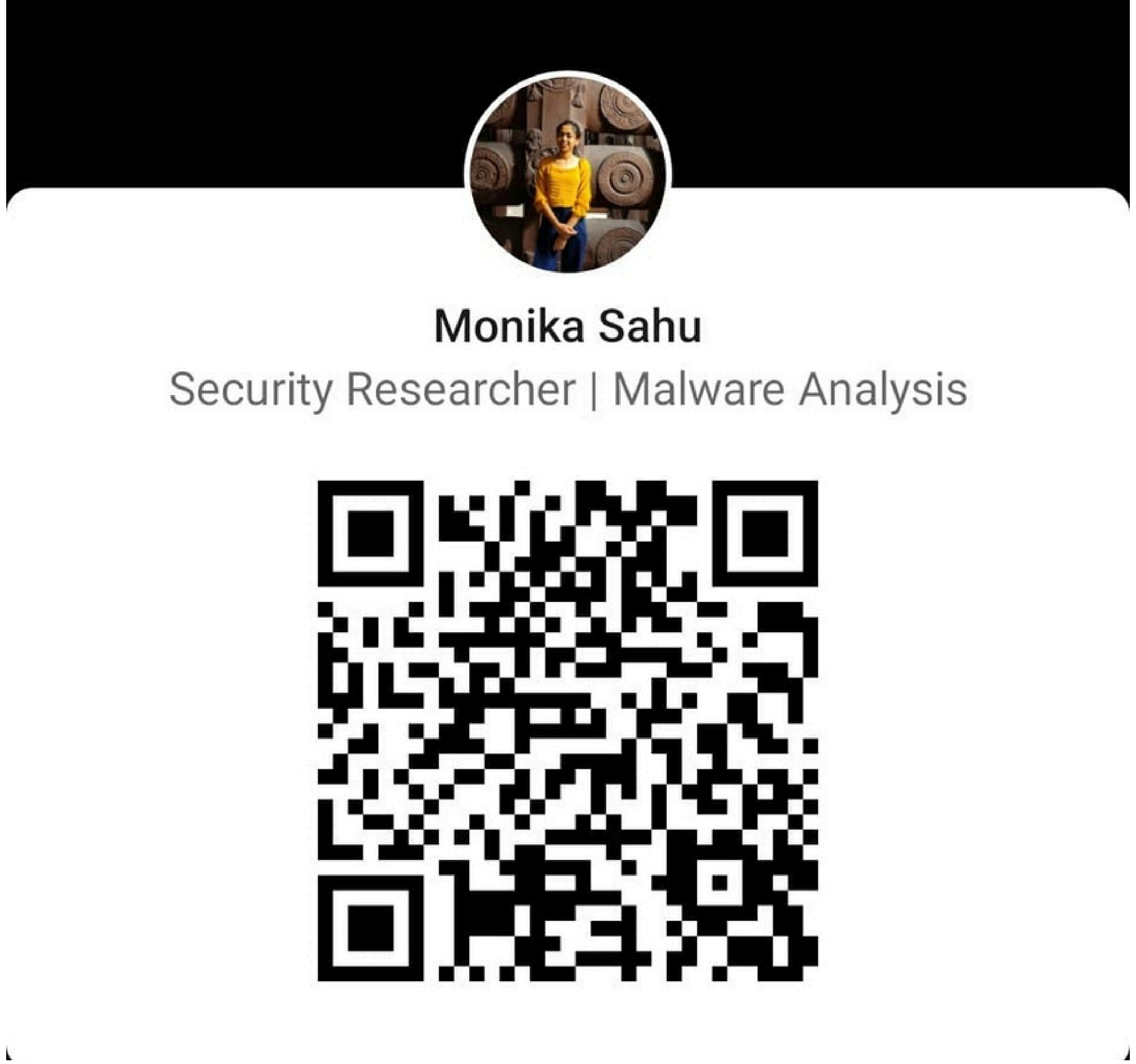
WHY & WHERE CORRELATION IS NEEDED?

- What do we mean by Correlation here?
- Key Components of Correlation
  - Source, Data, Volume, Time, Logic, Intelligence.
- Performance and False Positive/Negative.
- Behavioral Correlation, process manipulation, creation.
- Correlation and analyzing multiple branches of a process.
- Prioritizing and actions.

THE CONCLUSION

- Is an ideal EDR just a myth?
- The importance of the EDR.

Q&A



Monika Sahu

Security Researcher | Malware Analysis

THANK YOU!