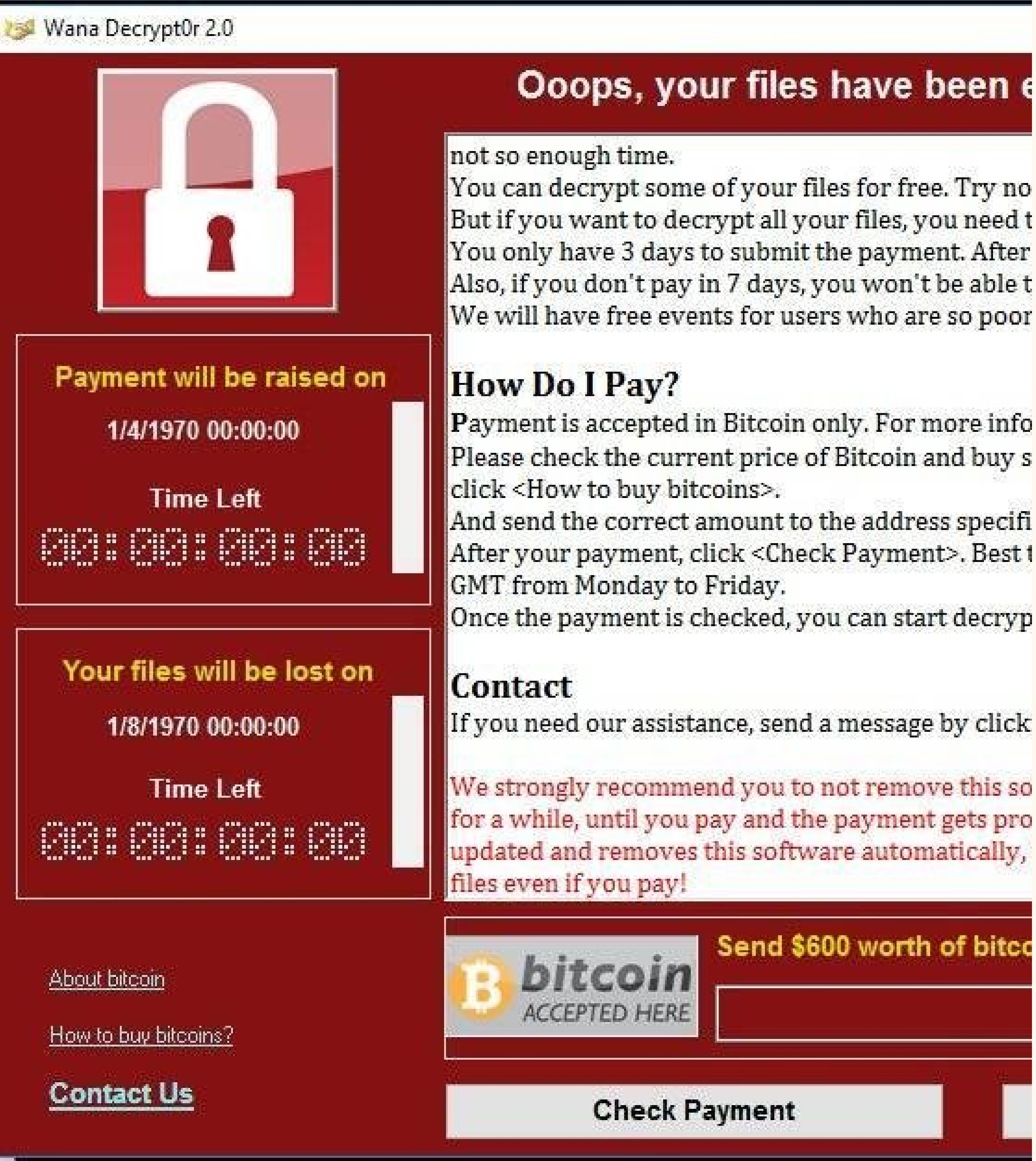
Malware: Creation and Detection of basic malware

$whoami

Nischay Hegde

ECE at ASEB (2022), Security Researcher at Uptycs

- Work on automating lots of the offensive processes
- Worked on several ransomware, Botnets, Flooders, Command and Control infrastructure
- Contributed to some eBPF libraries



Ooops, your files have been e

not so enough time.
You can decrypt some of your files for free. Try no
But if you want to decrypt all your files, you need t
You only have 3 days to submit the payment. After
Also, if you don't pay in 7 days, you won't be able t
We will have free events for users who are so poor

How Do I Pay?

Payment is accepted in Bitcoin only. For more info
Please check the current price of Bitcoin and buy s
click <How to buy bitcoins>.
And send the correct amount to the address specifi
After your payment, click <Check Payment>. Best t
GMT from Monday to Friday.
Once the payment is checked, you can start decryp

Contact

If you need our assistance, send a message by click

We strongly recommend you to not remove this so
for a while, until you pay and the payment gets pro
updated and removes this software automatically,
files even if you pay!

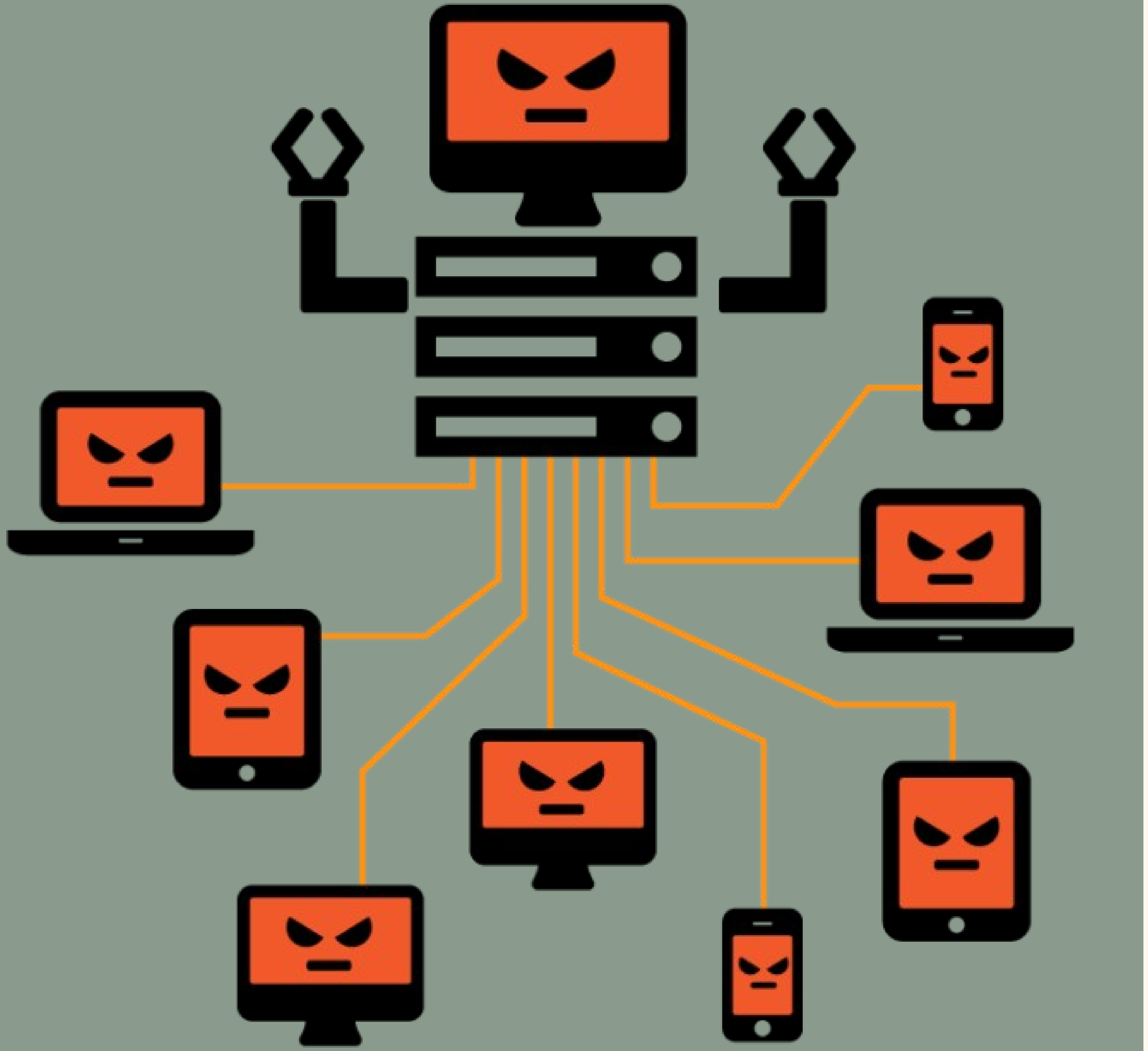
Send $600 worth of bitco

What is Malware?

Malware is any software that is designed to disrupt the normal operation of a computer system. That can be by stealing data, corrupting files, or even taking control of the system.

It includes viruses, worms, Trojans, ransomware, and spyware.

Malware can cause data loss, identity theft, and financial damage.



Some common types of Malware

A botnet is a network of computers that are all infected with malicious software, and made to operate as a network.

A ransomware is a program that encrypts files in certain important directories, and then usually demands money (ransom) in order to decrypt it.

A rootkit is a piece of software that maliciously persists in a system, without the user’s knowledge.

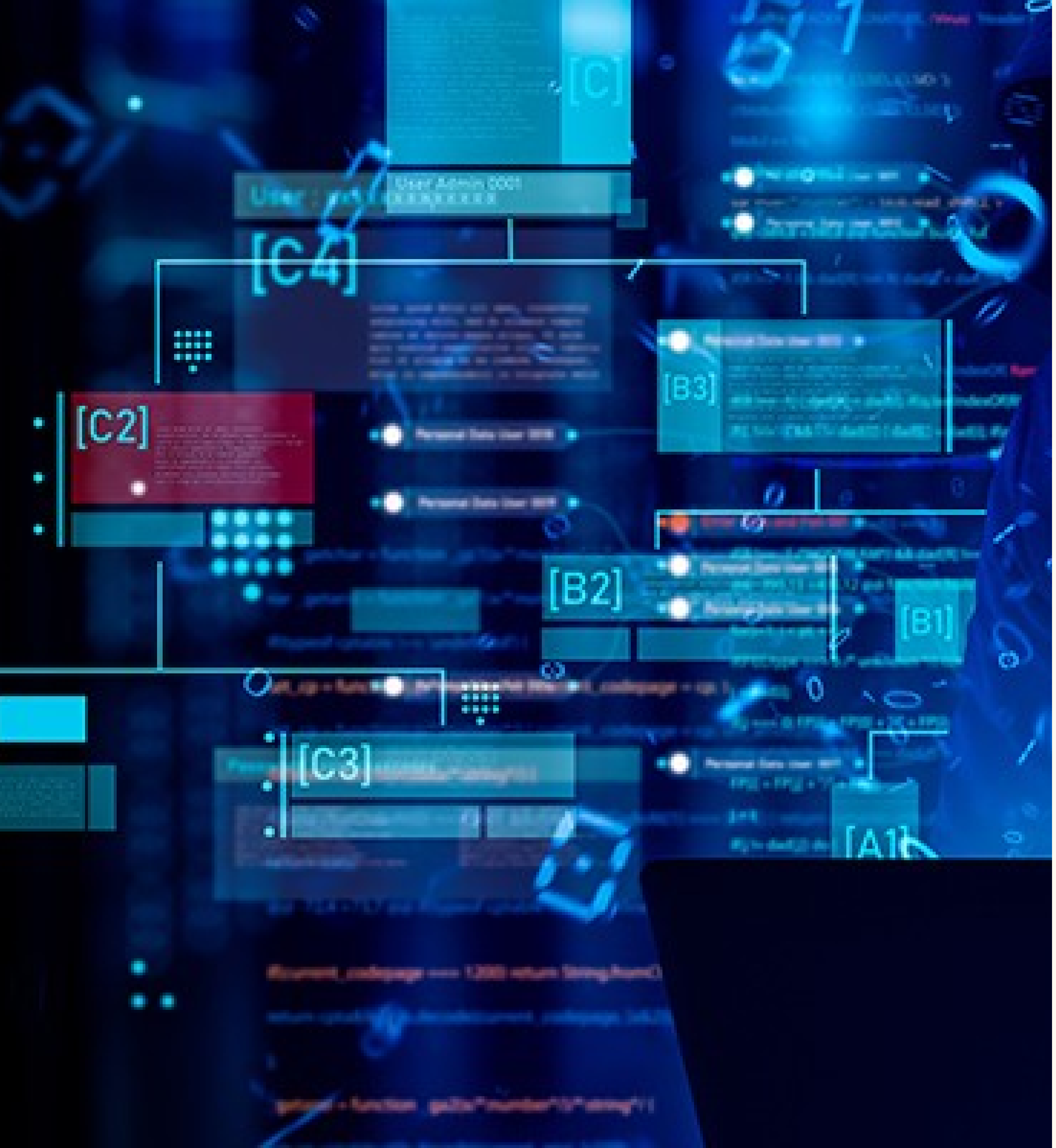
An Aside: MITRE ATT&CK

The MITRE ATT&CK Framework is a knowledge base of adversary tactics and techniques based on the real world.

Usually, it's used by Incident Response teams to easily report what malware has done in a succint and standardized manner.

But it can also be used by us to understand how malware do certain things.

DEMO: MITRE Page



Creating some basic malware

We require multiple steps to make malware.

Some Tactics that will be covered here:

- Execution
- Persistence
- Defense Evasion
- Discovery
- Command and Control
- Impact

DEMO: Malware

Malware detection

We have created malware, which operates in two places, i.e. Endpoint and Network.

Obviously, we would love to figure out how to detect it in both places.

MITRE Tactics

Tactics used:

T1053.003 – Persistence (Scheduled Task/Job: Cron)
T1059.004 – Execution (Command and Scripting Interpreter: Unix Shell)
T1071.001 – Command and Control (Application Layer Protocol)
T1041 – Exfiltration (Exfiltration Over C2 Channel)

DEMO: Network Detection

Detecting File changes

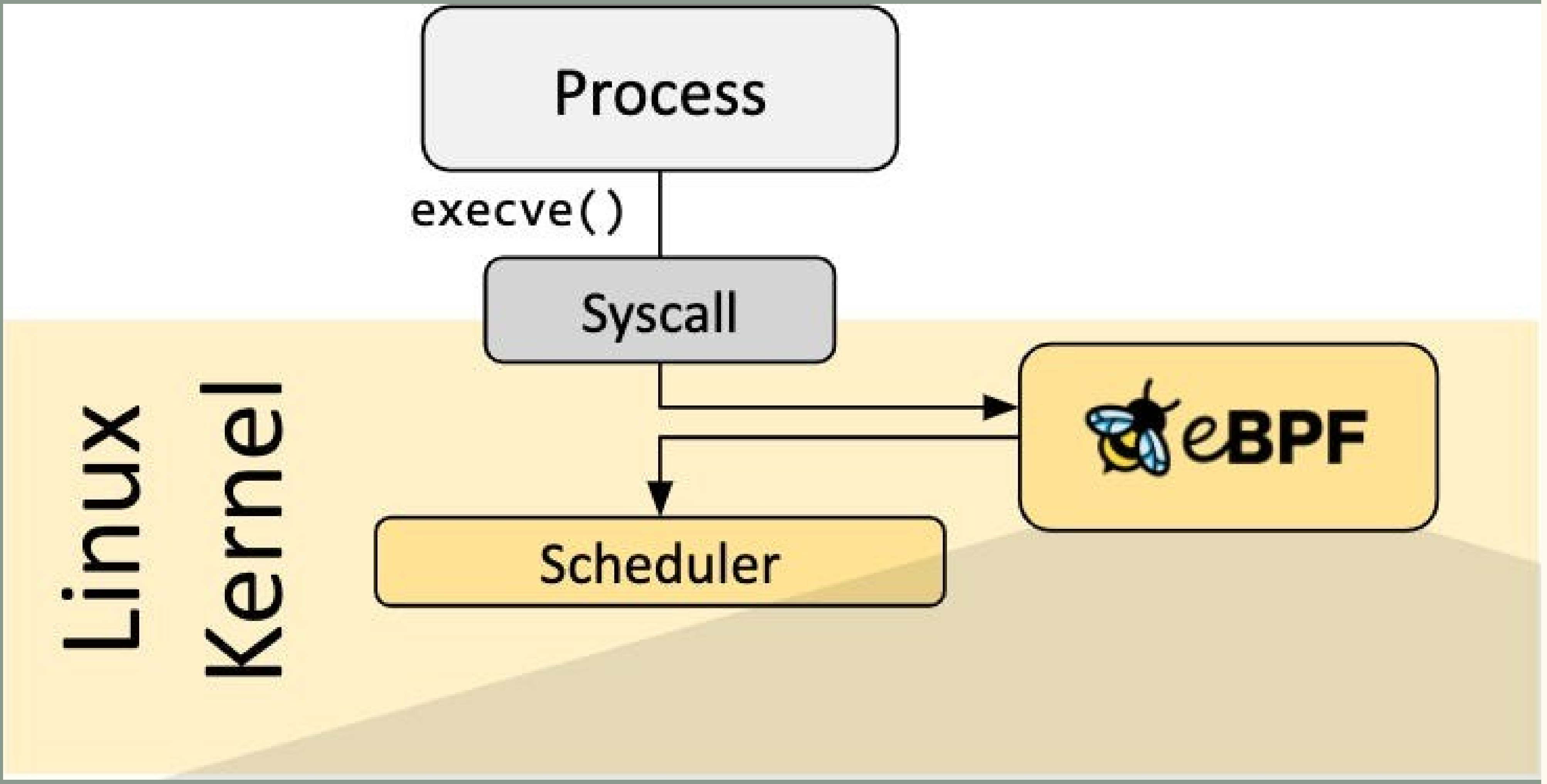
There are three APIs to do this on the kernel level:

- dnotify (old; deprecated)
- inotify (Linux v4.x)
- fanotify (Linux v5.x)

- We will be using fanotify as an example

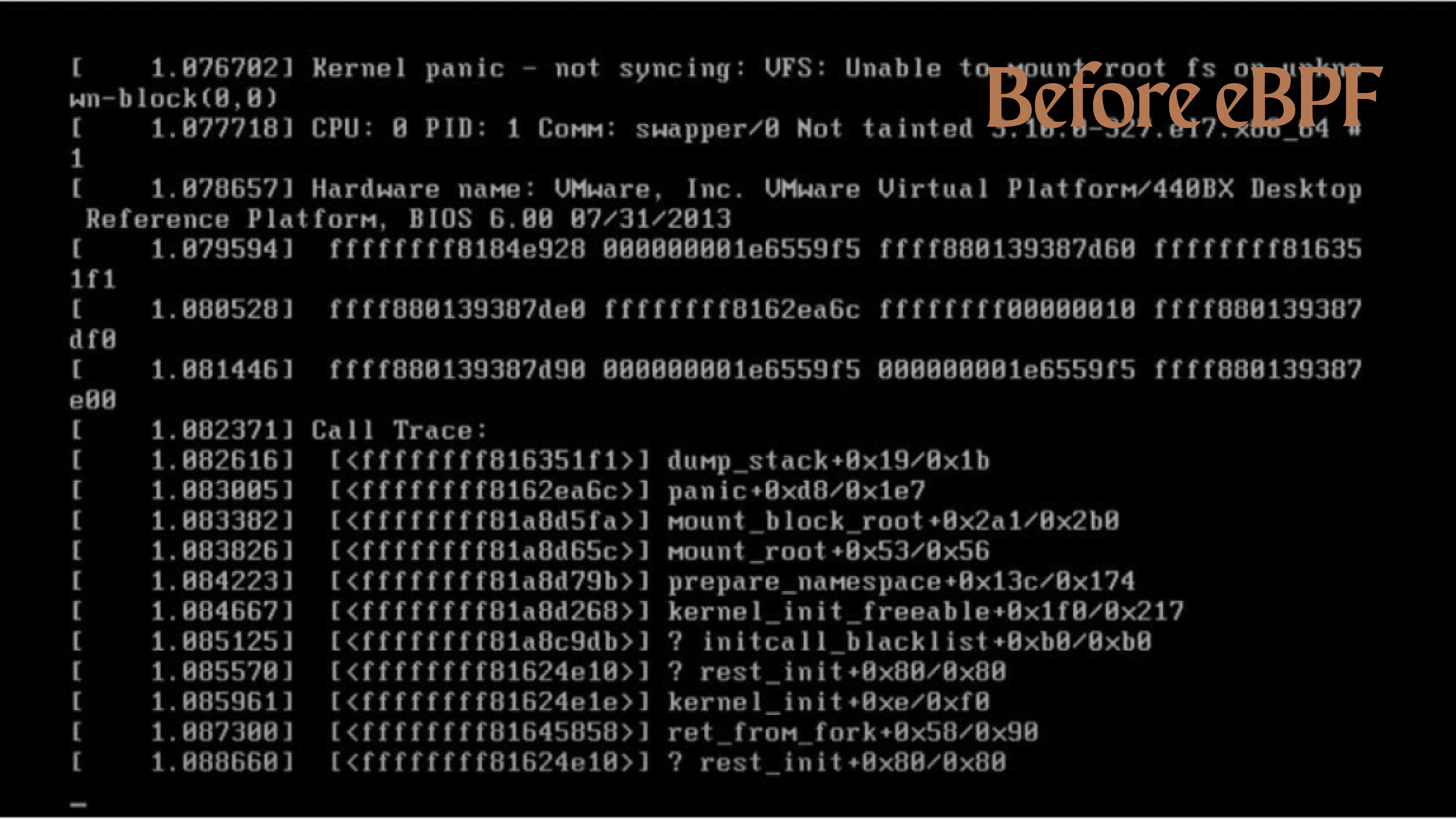
DEMO: fanotify program

eBPF



eBPF is the "new" (released in 2014), fancy technology that allows for developers (and security engineers) to get events directly from the kernel at nearly no cost and next to no kernel crashes.

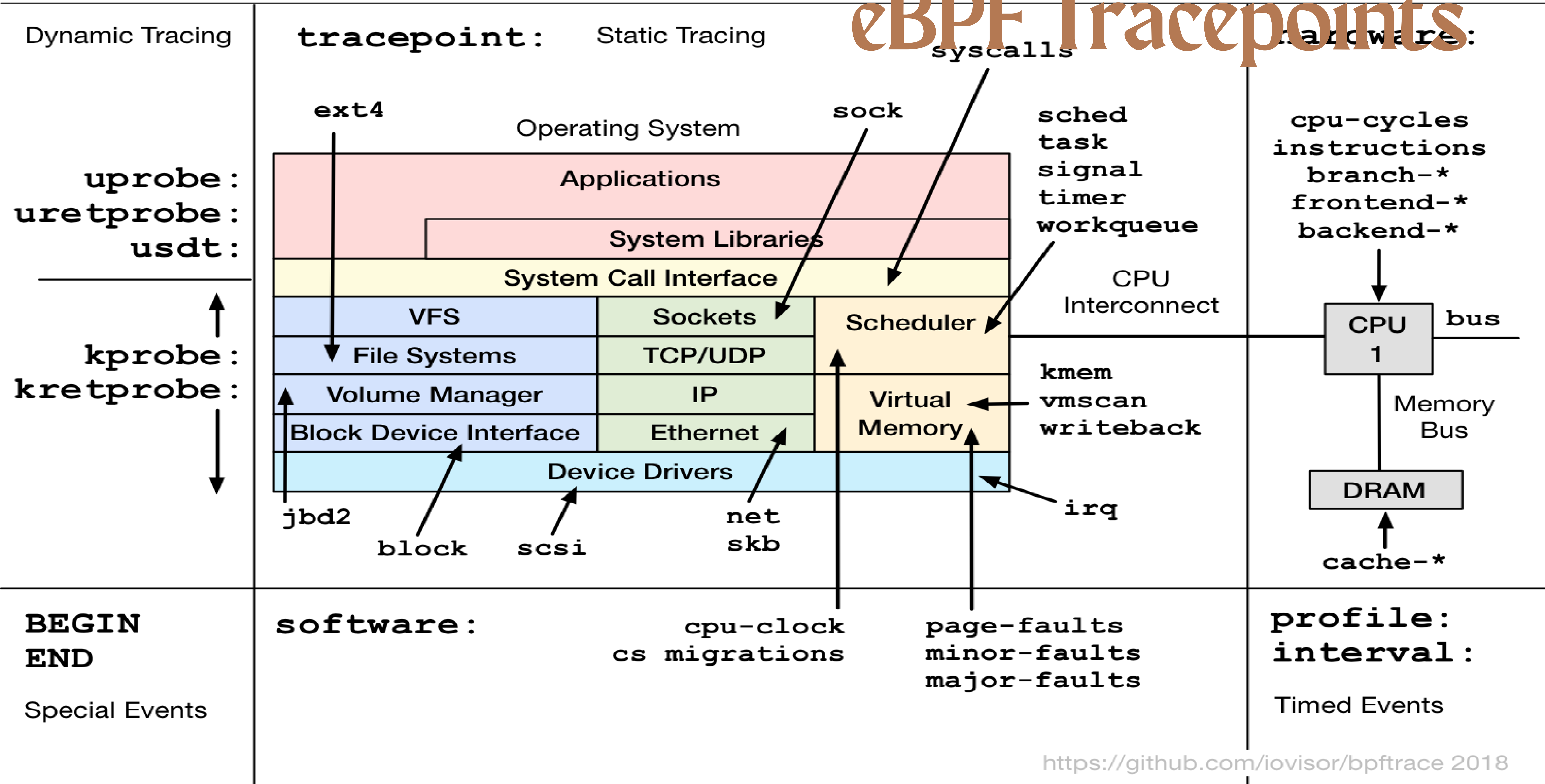
Before eBPF



```
[    1.076702] Kernel panic - not syncing: VFS: Unable to mount root fs on unkno
wn-block(0,0)
[    1.077718] CPU: 0 PID: 1 Comm: swapper/0 Not tainted 3.10.0-327.el7.x86_64 #
1
[    1.078657] Hardware name: VMware, Inc. VMware Virtual Platform/440BX Desktop
 Reference Platform, BIOS 6.00 07/31/2013
[    1.079594]  ffffffff8184e928 000000001e6559f5 ffff880139387d60 ffffffff81635
1f1
[    1.080528]  ffff880139387de0 ffffffff8162ea6c ffffffff00000010 ffff880139387
df0
[    1.081446]  ffff880139387d90 000000001e6559f5 000000001e6559f5 ffff880139387
e00
[    1.082371] Call Trace:
[    1.082616]  [<ffffffff816351f1>] dump_stack+0x19/0x1b
[    1.083005]  [<ffffffff8162ea6c>] panic+0xd8/0x1e7
[    1.083382]  [<ffffffff81a8d5fa>] mount_block_root+0x2a1/0x2b0
[    1.083826]  [<ffffffff81a8d65c>] mount_root+0x53/0x56
[    1.084223]  [<ffffffff81a8d79b>] prepare_namespace+0x13c/0x174
[    1.084667]  [<ffffffff81a8d268>] kernel_init_freeable+0x1f0/0x217
[    1.085125]  [<ffffffff81a8c9db>] ? initcall_blacklist+0xb0/0xb0
[    1.085570]  [<ffffffff81624e10>] ? rest_init+0x80/0x80
[    1.085961]  [<ffffffff81624e1e>] kernel_init+0xe/0xf0
[    1.087300]  [<ffffffff81645858>] ret_from_fork+0x58/0x90
[    1.088660]  [<ffffffff81624e10>] ? rest_init+0x80/0x80
_
```

eBPF Tracepoints

bpftrace Probe Types



Dynamic Tracing

uprobe:
uretprobe:
usdt:

kprobe:
kretprobe:

tracepoint: Static Tracing

hardware:

BEGIN
END

Special Events

software:

cpu-clock
cs migrations

page-faults
minor-faults
major-faults

profile:
interval:

Timed Events

https://github.com/iovisor/bpftrace 2018

DEMO: bpftrace showcase

Get in touch!

@thatloststudent

thatloststudent@infosec.exchange

✉ hegdenischay@proton.me

🌐 https://nischay.me

[matrix] @nischay:intothematrix.in

References

- https://github.com/iovisor/bpftrace/blob/master/docs/tutorial_one_liners.md
- https://attack.mitre.org/matrices/enterprise/
- https://pyfanotify.readthedocs.io/en/latest/

Thank you for joining us in the fight against cyber threats. Stay vigilant and stay safe!