

**7**

**Apr 2025**

# Linux Detective Toolkit

360° debugging and black-box analysis without printf() or GDB



**Bartosz Moczulski**

<https://bartosz.codes/>



**POZNAŃSKA**  
IMPREZA WOLNEGO  
OPROGRAMOWANIA



# printf()-less debugging



**software  
integration**



**security  
pentesting**



**home lab  
self-hosting**

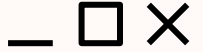
# Similarities

## Challenge

- Embedded
- Home-lab
- Pentesting

## Scenario

- no source code
- no debugger and/or no debug symbols
- no logs
- (or limited availability of either of these)



# Disclaimer



## Hacking

Using computer system in a clever way, often unforeseen by its authors.



## Cyber-crime

Illegal activity, unauthorised access, identity and data theft, ransoms, money laundering, ...



# whoami



**Bartosz  
Moczulski**

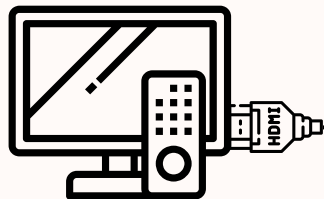


**programmer  
>20y**



**Linux  
detective**

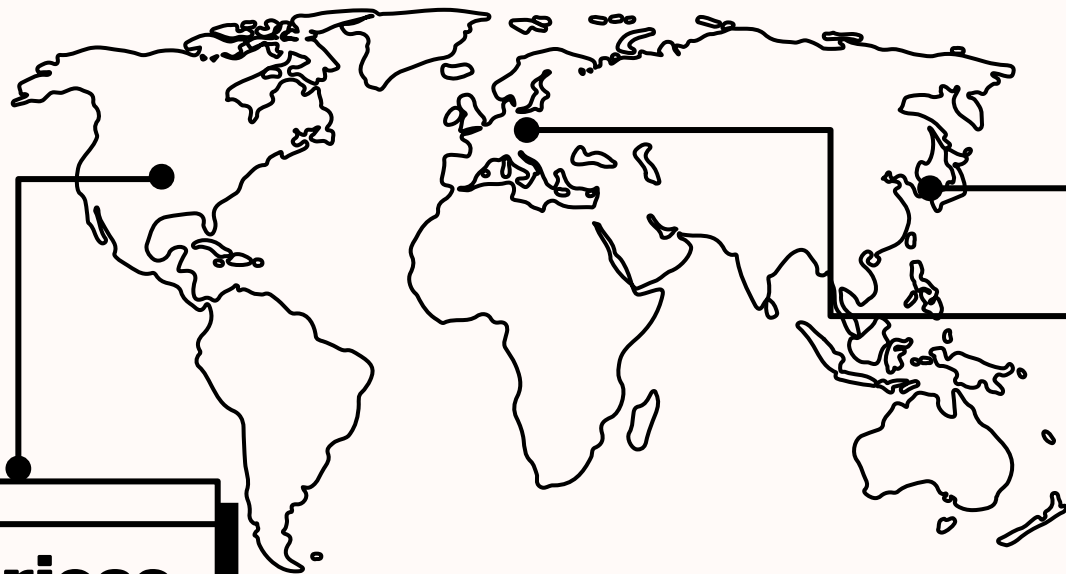
# I want you



# to watch TV

and streaming services

# I was (involved) here



## Americas

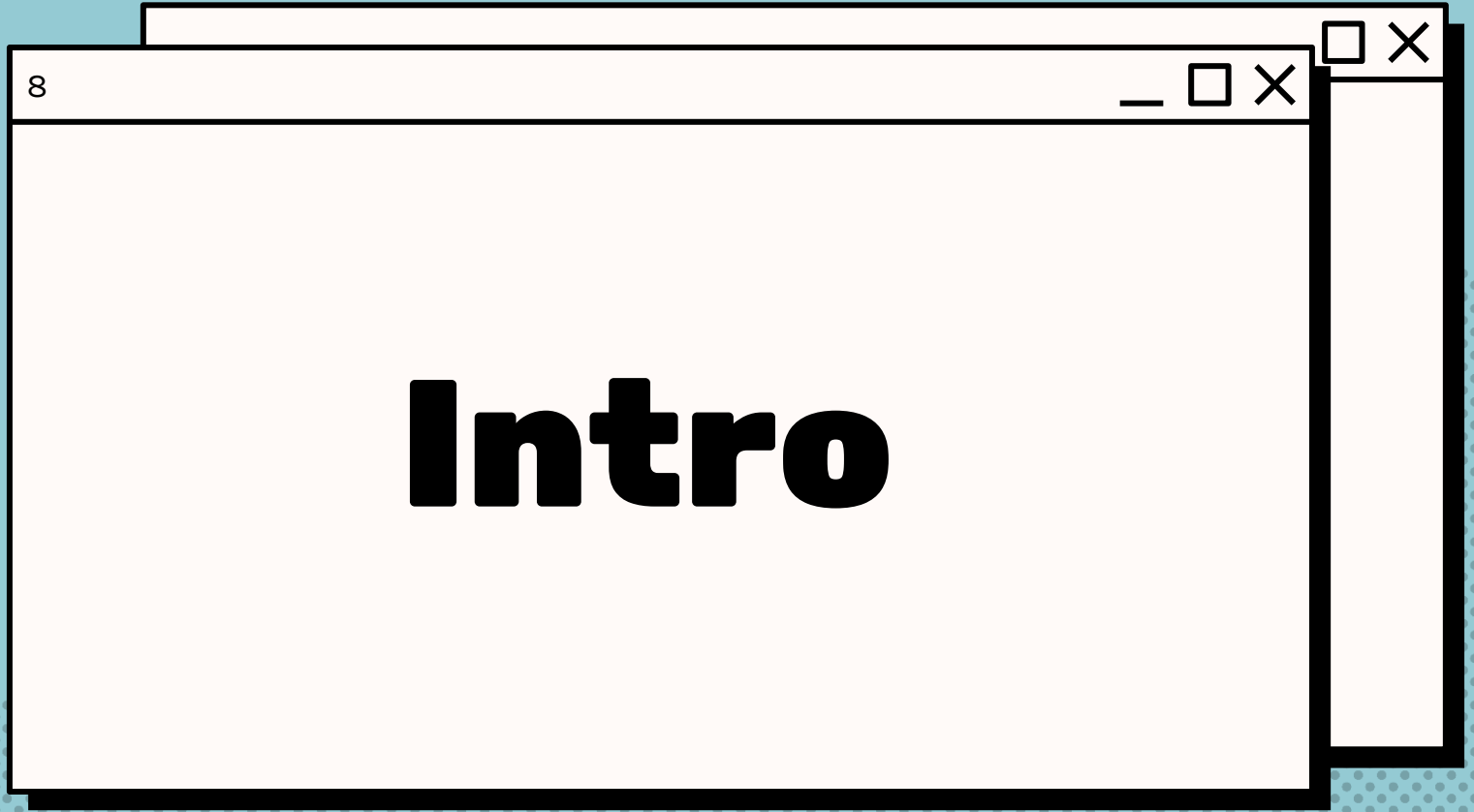
USA, (Mexico, Chile)

## Asia

South Korea

## Europe

Poland, Czechia,  
Germany, Austria,  
Netherlands, Belgium,  
Sweden, Norway, Italy,  
UK, Spain, ...





# Black-box vs. Glass-box



**What?**

happens



**Why?**

it happens



**How?**

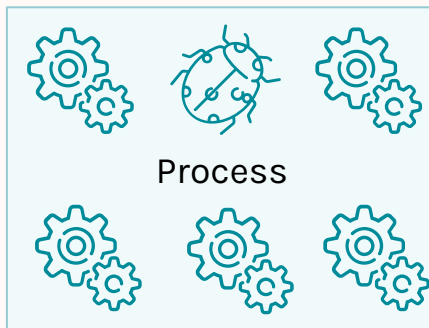
it happens  
(and how to fix it)

# All fine? Don't tease Pandora!

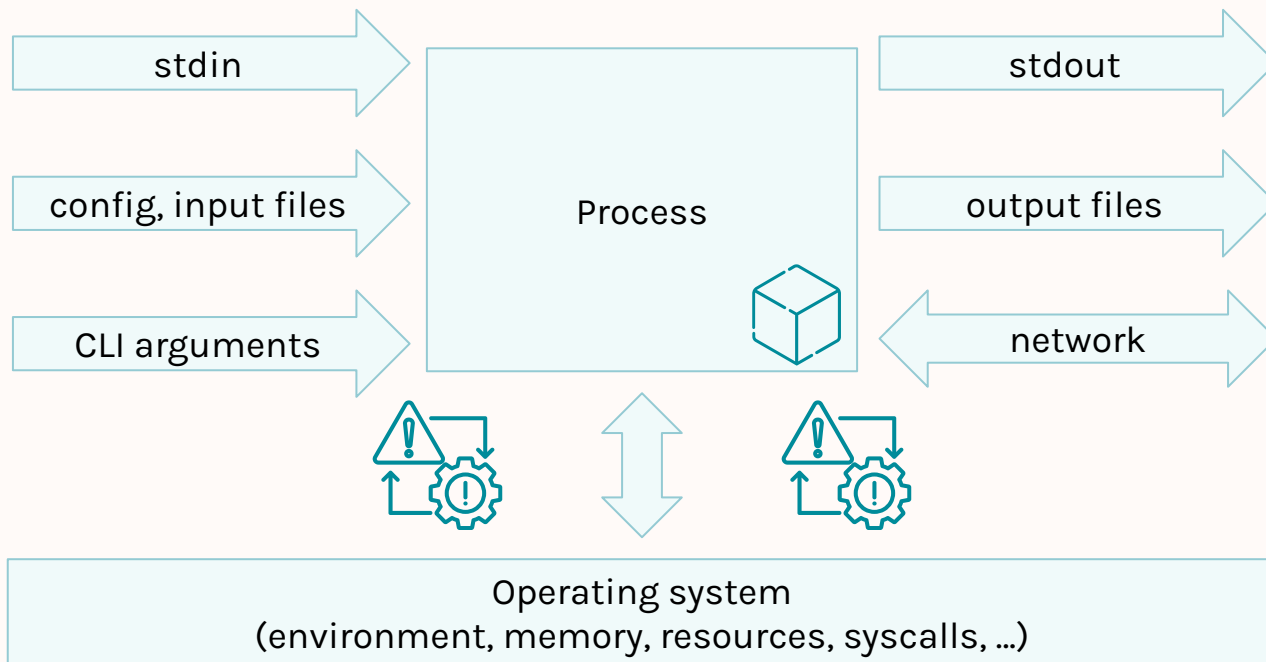
Process



# A problem? Perhaps inside?



# Or somewhere **HERE?**



# **outcome = function(input, ...)**

## **Outcome**

- what can be observed
- bug, or correct but unexpected behavior

## **Function**

- the program being analyzed

## **Parameters**

- input (files, arguments)
- external interactions
- past state

# Cherry-picked tricks

You're welcome 😊



15



# Binary runtime



# **strace / ltrace**

What is this app doing?





# LD\_PRELOAD

Call hijacking

# LD\_PRELOAD

Can do:

- change input and/or return
- apply additional logic
- change completely

Possible use cases:

- log calls - add missing printf()s
- unit tests
- fixups/customizations



# ldd

List dependencies (but not plugins)

```
LD_TRACE_LOADED_OBJECTS=1
```

```
objdump -x
```



# LD\_AUDIT

(LD\_PRELOAD on steroids)

`man ld.so`

`man rtld-audit`

21

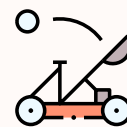


# Processes



# ps, pstree

ps aux vs. ps -ef  
pstree -catApulTsg





# **/proc/N/...**

## **status**

(uid, gids, RAM, namespaces, ...)



# **/proc/N/...**

**cmdline, environ**

```
xargs -0  
tr '\0' '\n'
```





# **/proc/N/...**

**fd, fdinfo**



# **/proc/N/...**

root



# \*top

top, htop, btop, atop  
iotop, iftop



# mount

**-o noexec**



# PID namespace

`fork()`

exiting parent, init process

30



# Networking



# **netstat, ss**

(list all sockets)

# Wiretapping

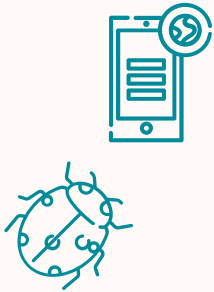
(legal)





# Network communication

**Client**



**Switch/Router**



**Server**



# Wiretapping plain HTTP

**Client**



tcpdump

**Switch/Router**



tcpdump  
port mirroring

**Server**



out of our control

# Wiretapping TLS (HTTPS) v1

**Client**



**Switch/Router**



**Server**



Diffie-Hellman  
RSA, AES, PKI

out of our control

# Wiretapping TLS (HTTPS) v1

**Client**



tcpdump  
+ self-reporting

**Switch/Router**



**Server**



out of our control

# self-reporting



```
export SSLKEYLOGFILE=foo.log
```



# SSLKEYLOGFILE support

	support	(lib)curl	wget	Chromium, Firefox	glib-networking (e.g. WebKit)
OpenSSL BoringSSL ...	API	✓	✗	✓	✗ (DM me)
mbedtls	API	✗	✗	?	?
GnuTLS	library ✓	✓	✓	✓	✓
rustls	library +API (?)	-	-	-	-



# Wireshark

with TLS decryption  
(live and post-capture)

# Wiretapping TLS (HTTPS) v2

**Client**



CA budle ++  
(I trust you!)

**Switch/Router**



MITM proxy  
(trust me!)

**Server**



out of our control

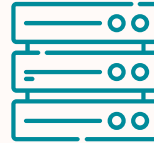


# PKI 101

**Client**

**Switch/Router**

**Server**



I trust: 

\*.foo.com

chain of trust

# PKI 101

**Client**



I trust:



≠



\*.foo.com

(trust me!)

**Switch/Router**



**Server**








# PKI 101

**Client**

**Switch/Router**

**Server**



I trust:   =    \*.foo.com  
(trust me!)

# MITM vs PITM

(man vs. person in the middle)  
euphemism treadmill



# CA bundle + mitmproxy

	file / dir / option
OpenSSL BoringSSL ...	<code>openssl version -a</code> # e.g. <code>/usr/lib/ssl/certs/</code> (+ iptables or DNS)
curl	<code>--proxy http://127.0.0.1:8080</code> (or iptables or DNS) <code>--cacert &lt;file&gt;</code> <code>--capath &lt;dir&gt;</code>
wget	<code>-e https_proxy=127.0.0.1:8080</code> (or iptables or DNS) <code>--ca-certificate &lt;file&gt;</code> <code>--ca-directory &lt;dir&gt;</code>



# mitmproxy

Let's do it! 🐈

47



# Multi- threading

# MT challenges



## Data races

- Mutex missing entirely
- 2 atomic vars  $\neq$  1 atomic pair
- MT-safe? Check you man page!
- (out of scope today)



## Deadlocks

- 2 or more mutexes
- ABBA (and more)
- detection
- prevention (out of scope today)





# ABBA

## Available

- lockdep (in Linux kernel)
- valgrind --tool=helgrind
- liblksmith (10y old)
- GDB scripts (x2)



## Much lamented

- liblockdep (user-space)



(2013-2021)





# ABBA

valgrind      lksmith

GDB scripts

# ABBA - GDB scripts



by Damian Ziobro



/DamZiobro/gdb-automatic-deadlock-detector



by Adam Szaj



/adam-szaj/gdb-scripts (find\_deadlock)

52



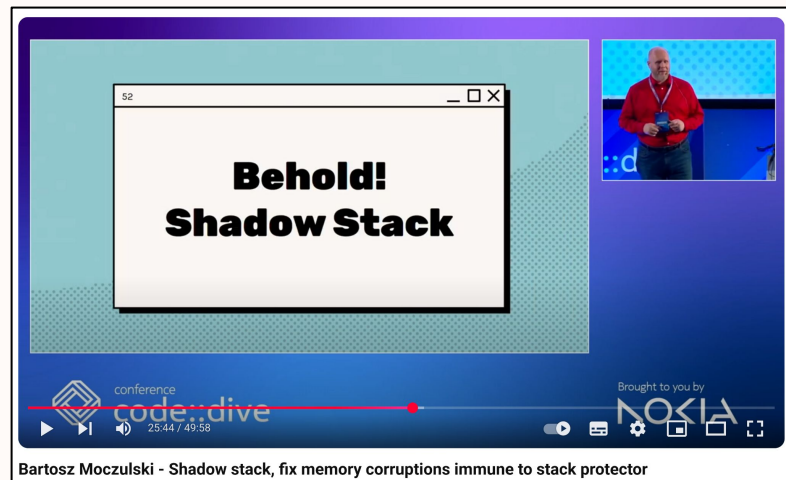
# Advanced tools

# Shadow Stack

Finding stack corruptions

<https://youtu.be/AG2yb9GRDoQ>

<https://github.com/bmoczulski/shadow-stack/>





# Valgrind

A must-know tool



- Memcheck (memory)
- Helgrind (multithreading)
  - deadlocks, data races
- and more ...

Powerful but ...

- x100 overhead - time and RAM (often impractical)

## RTFM



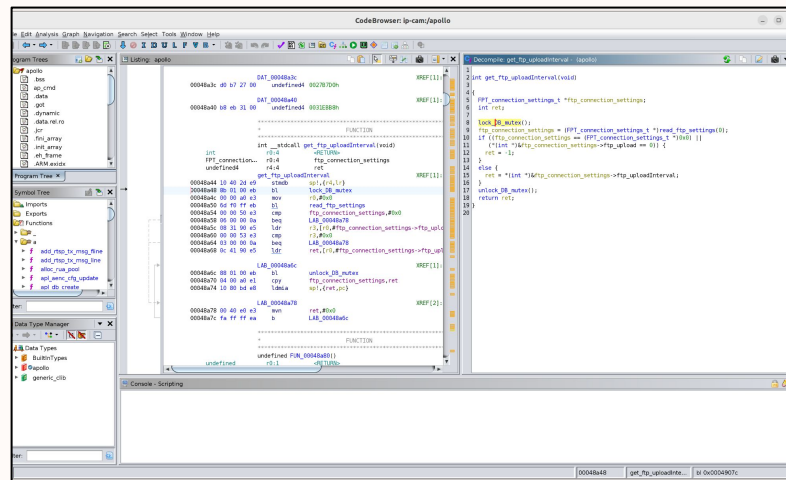
- May require code annotations
- Calling printf() may mess things up
- Blind to conditional variables (sic!)
- Linux futex and atomics = no no
- Memcheck + Helgrind =  + 



# Ghidra

Reverse engineering and  
disassembling can never  
be wrong, right?

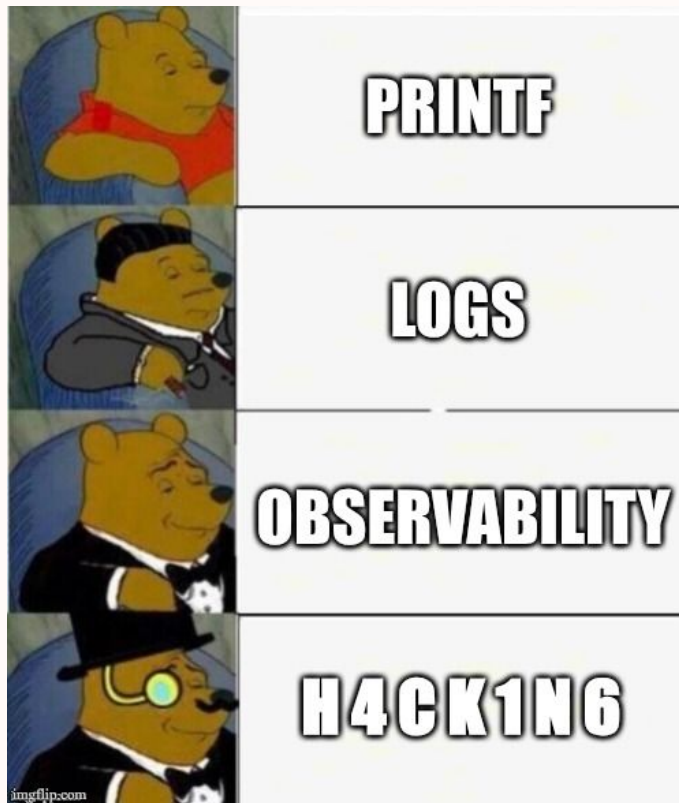
But check your EULA!





# TL;DR

“The evolution of debugging techniques”



58



# Thank you

Bartosz Moczulski  
will return

# Q & A

<https://bartosz.codes>