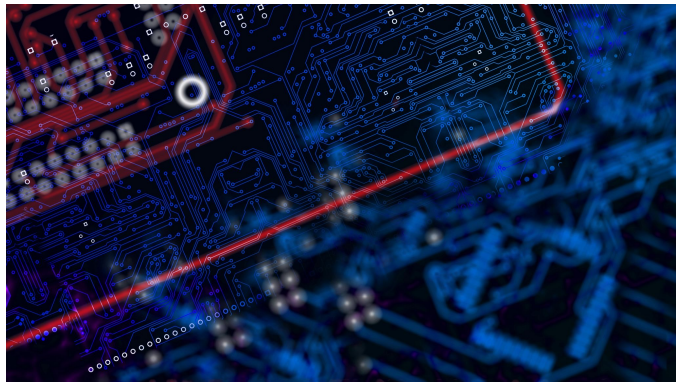
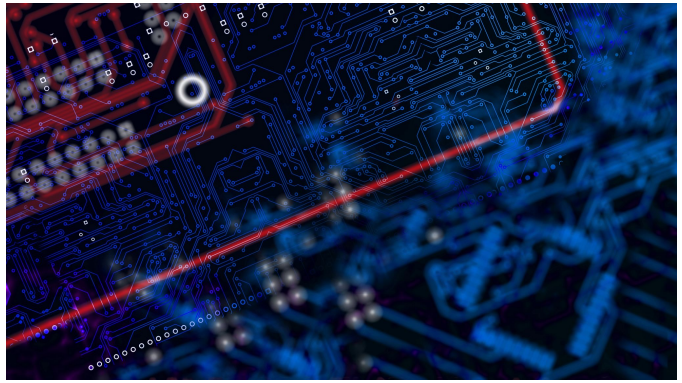

Memfini - System-wide
memory event Monitoring
interface for Linux

About us



Shubham Dubey

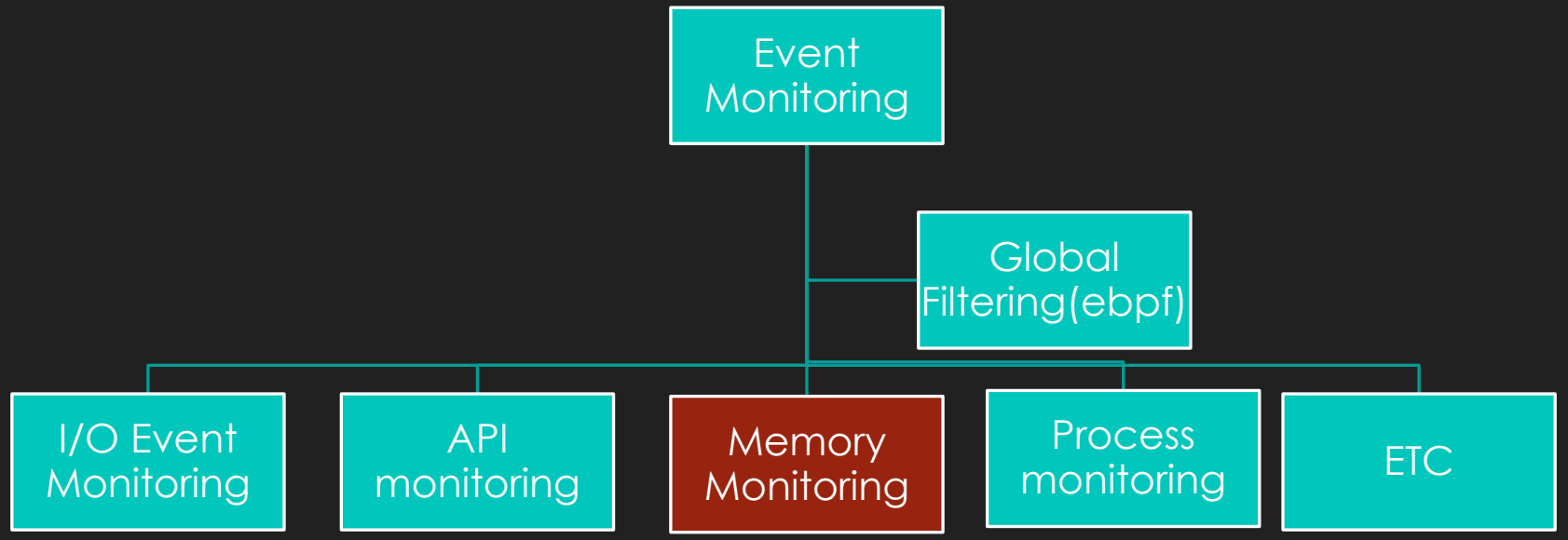
Security Researcher - Low level security expert
@nixhacker
https://nixhacker.com



Rishal Dwivedi

Security Researcher - AppSec, Malware
https://www.linkedin.com/in/rishaldwivedi/

Motivation

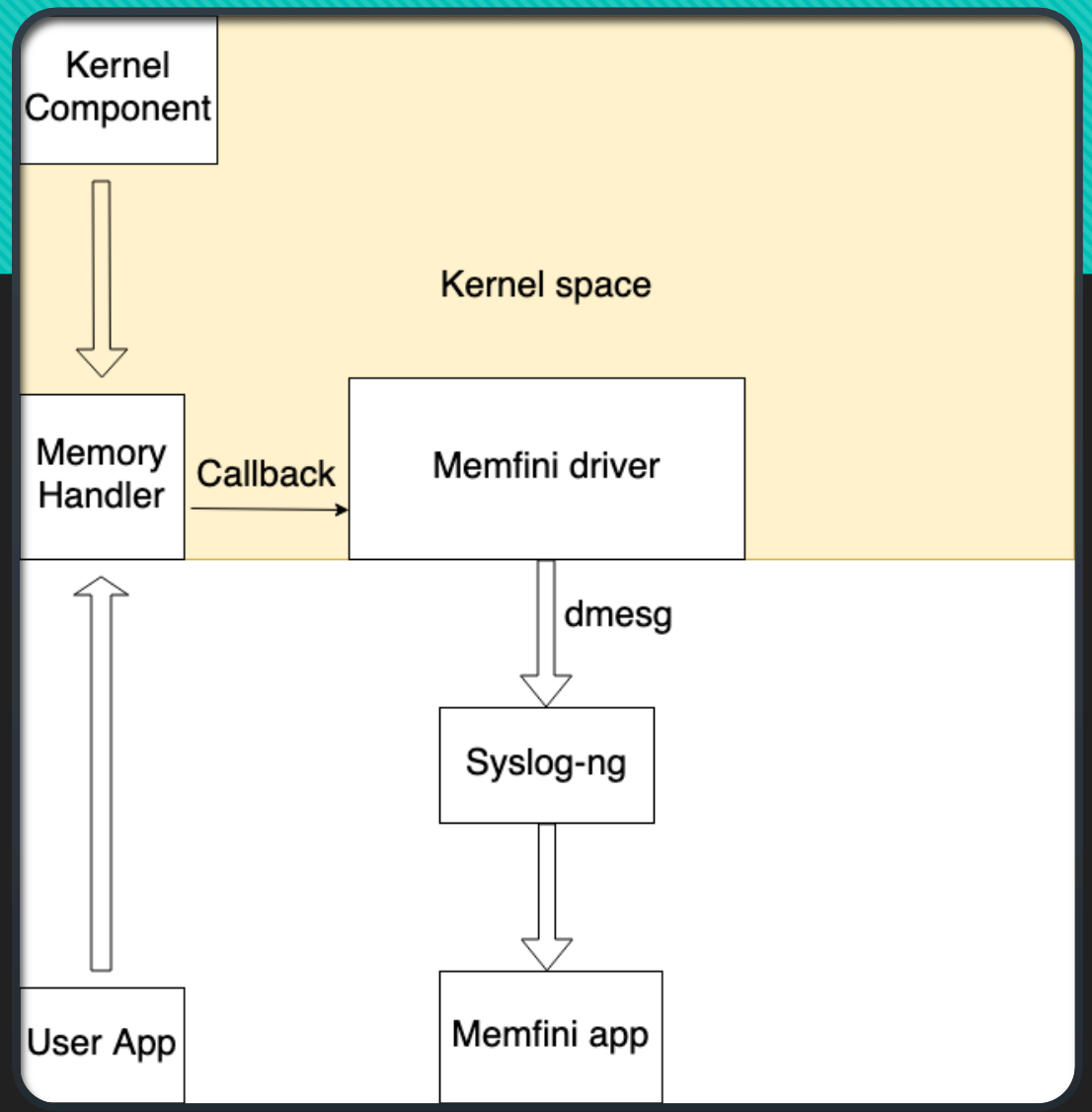

Event Monitoring
Global Filtering(ebpf)
I/O Event Monitoring
API monitoring
Memory Monitoring
Process monitoring
ETC

Motivation

- Trying to fill the gap

- Although, it may look like it has niche usage compared to others but this functionality can be quite useful for security researcher, forensic experts or even developers.

- At same time we are trying to make it as generic as possible so that users can use it in whatever way they want.



TL;DR

- It's an interface/Tool that can be used to Log memory related events generated Systemwide (Userspace and Kernelspace)
- Behind the scene, It has a kernel module that register callbacks for all memory related function calls and send the data to userspace application. Userspace memfini application is a command line tool that you can use to view and filter data according to the usage.

Targeted audiences

Security Researcher: Helps in finding indicators of incidents. Detecting Process injections, IPC using Shared memory, Vulnerability research focusing on memory corruption, Shellcode execution performed by malwares.

Developers: For Tracing Memory allocation, Debugging Allocation Errors Or memory Leaks. Memory Profiling at userspace or kernel space

Geeks: Since they love stats/Logs.

Usage

```
------------------ MEMFINI v0.1 - System-wide Memory Monitoring Interface -------------
usage: memfini [-h] [--start] [--stop] [--pid PID] [--pname PNAME] [--foreign] [--shared] [--inmemory] [--kernel] [--error]

optional arguments:
  -h, --help     show this help message and exit
  --start        Start Memfini
  --stop         Stop Memfini
  --pid PID      Filter by Process ID
  --pname PNAME  Filter by Process Name
  --foreign      Foreign Process access
  --shared       Shared memory events
  --inmemory     In-memory File Creation
  --kernel       Kernel memory events
  --error        Failed events
```

Screenshots

```
parallels@parallels-Parallels-Virtual-Platform:~/memfini$ memfini --foreign
------------------ MEMFINI v0.1 - System-wide Memory Monitoring Interface -------------
Aug 12 17:08:26 Process inject(44374) performed memory allocation in foreign process PID:44010 of size 4 bytes at address 0x7f7b269b1fd2

Aug 12 17:08:26 Process inject(44374) performed memory allocation in foreign process PID:44010 of size 4 bytes at address 0x7f7b269b1fd6

Aug 12 17:08:26 Process inject(44374) performed memory allocation in foreign process PID:44010 of size 4 bytes at address 0x7f7b269b1fda

Aug 12 17:08:26 Process inject(44374) performed memory allocation in foreign process PID:44010 of size 4 bytes at address 0x7f7b269b1fde

Aug 12 17:08:26 Process inject(44374) performed memory allocation in foreign process PID:44010 of size 4 bytes at address 0x7f7b269b1fe2

Aug 12 17:08:26 Process inject(44374) performed memory allocation in foreign process PID:44010 of size 4 bytes at address 0x7f7b269b1fe6

Aug 12 17:08:26 Process inject(44374) performed memory allocation in foreign process PID:44010 of size 4 bytes at address 0x7f7b269b1fea

Aug 12 17:08:26 Process inject(44374) performed memory allocation in foreign process PID:44010 of size 4 bytes at address 0x7f7b269b1fee
```

Screenshots

```
parallels@parallels-Parallels-Virtual-Platform:~/memfini$ memfini --kernel |head -100
------------------ MEMFINI v0.1 - System-wide Memory Monitoring Interface -------------
Aug 12 17:11:25 Kernel Memory allocation perfomed of size 8192 and flags 0xdc0. Caller details: alloc_ftrace_hash+0x54/0x80

Aug 12 17:11:25 Kernel Memory allocation perfomed of size 8 and flags 0xdc0. Caller details: alloc_ftrace_hash+0x54/0x80

Aug 12 17:11:25 Kernel Memory allocation perfomed of size 49 and flags 0x400cd0. Caller details: __d_alloc+0x179/0x1f0

Aug 12 17:11:25 Kernel Memory allocation perfomed of size 49 and flags 0x400cd0. Caller details: __d_alloc+0x179/0x1f0

Aug 12 17:11:25 Kernel Memory allocation perfomed of size 49 and flags 0x400cd0. Caller details: __d_alloc+0x179/0x1f0

Aug 12 17:11:25 Kernel Memory allocation perfomed of size 49 and flags 0x400cd0. Caller details: __d_alloc+0x179/0x1f0

Aug 12 17:11:25 Kernel Memory allocation perfomed of size 49 and flags 0x400cd0. Caller details: __d_alloc+0x179/0x1f0

Aug 12 17:11:25 Kernel Memory allocation perfomed of size 8192 and flags 0xdc0. Caller details: alloc_ftrace_hash+0x54/0x80

Aug 12 17:11:25 Kernel Memory allocation perfomed of size 49 and flags 0x400cd0. Caller details: __d_alloc+0x179/0x1f0

Aug 12 17:11:25 Kernel Memory allocation perfomed of size 49 and flags 0x400cd0. Caller details: __d_alloc+0x179/0x1f0

Aug 12 17:11:25 Kernel Memory allocation perfomed of size 49 and flags 0x400cd0. Caller details: __d_alloc+0x179/0x1f0

Aug 12 17:11:25 Kernel Memory allocation perfomed of size 49 and flags 0x400cd0. Caller details: __d_alloc+0x179/0x1f0
```

Demo - Index

- Tracking allocations – Shellcode execution, Allocation errors etc
- Kernel memory monitoring
- Unpacking
- IPC
- Process injections
- Fileless execution
- Memory Profiling

Packer, Loader

- Packer helps hiding the malware by encrypting or compressing the executable code.
  - UPX packer
    - 60-70% Linux malwares are UPX packed
    - Custom packer
- Loader is responsible for delivering the main payloads into the system, executing staying evasive.
  - Ezuri loader
    - Decrypts the binary data & loads it in-memory for execution.
    - TeamTNT – Tsunami Botnet
- Shellcode execution - RWX

Fileless malware

- Why in-memory execution?
  - Malware is executed in memory (RAM), no physical presence on disk.
  - Evade detection, no footprint.
- Shmget/shmat or shm_open
  - Used to create or open a shared memory object
  - Used for IPC - Rotajakiro
  - Files residing in /dev/shm (tmpfs) are in virtual memory (creates mount point), no presence on disk – Xorddos, OrBit, LOKI2
- Memfd_create
  - Preceder of shm_open, introduced in kernel 3.17.
  - Create anonymous file in memory, no presence on disk, no mount points - Tsunami botnet, pupy, merlin

Process Injection

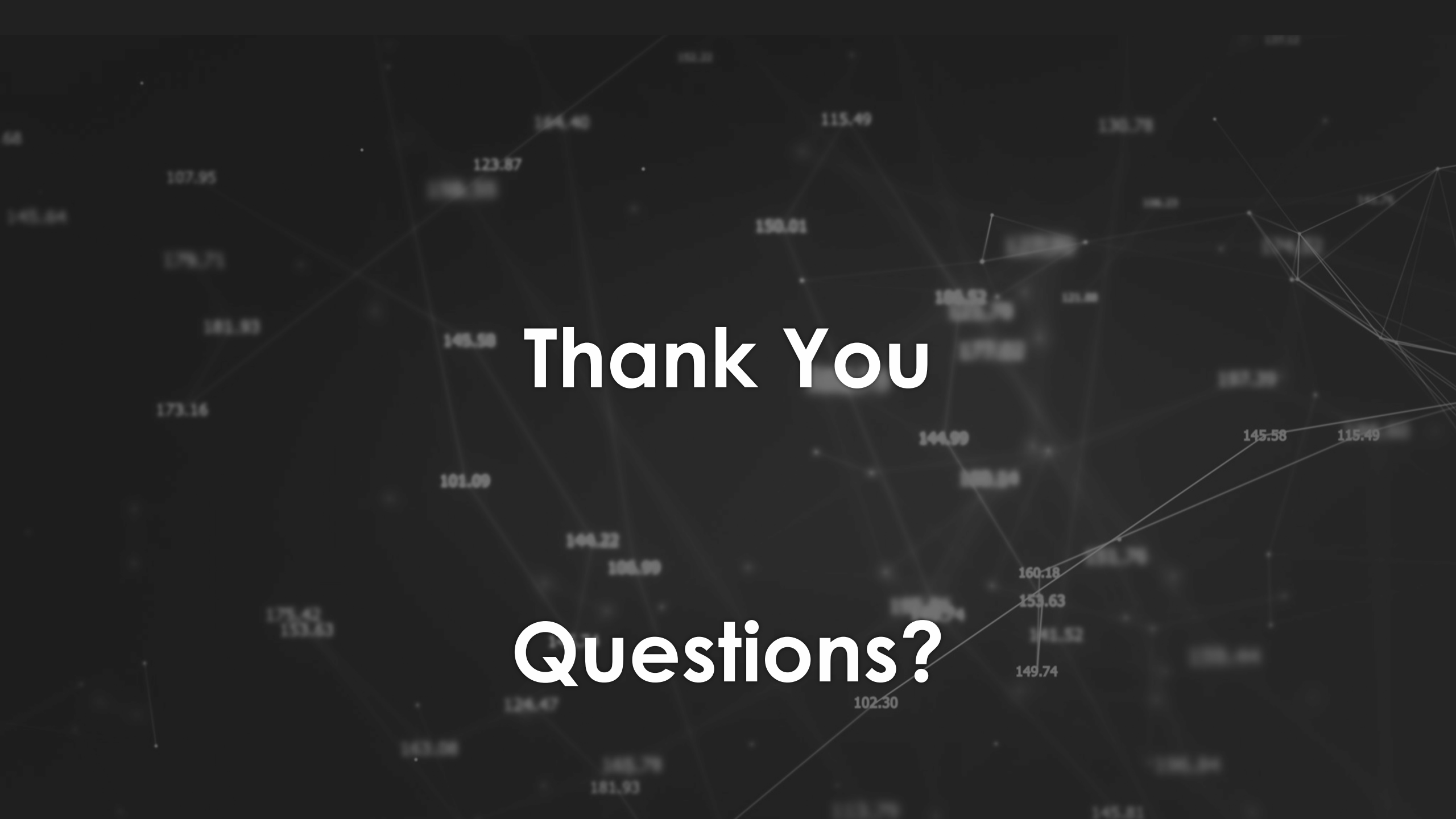
- Malware abusing it to inject into another process memory.
- LD_PRELOAD
- Ptrace
  - Useful for inspecting execution of other process. Inspecting, modifying process memory, tracing register values.
  - Widely used for debugging. Ex - gdb
  - Also, using it as Anti-debug technique.
- Process_vm_writev
  - Introduced in kernel > 3.2

DEMO

Memory Profiling

- Memfini can be used for memory profiling.
- Users can create a wrapper around memfini logs to profile systemwide/process specific memory usage.

```
codie@codie-VirtualBox:~/memfini$ sudo python3 profiler.py --pid 1520
---Memory profiler---
Toal memory usage: 36.61 Megabyte
Number of allocations: 619
Number of unallocations: 538
```


Thank You
Questions?