

The Evolution of **VOLATILE MEMORY FORENSICS**

Anggota



2501981270

AGUSTINUS
LEONARDO
DWITAMA



2501965052
KEVIN
MORRIS
ARMANDO



2502001864
RAVI
DEEVAN
SATYAKI



2501989550
MUHAMAD
DWI
APRIYANTO



2501983723
NATANAEL
FRANSISCO

Memory acquisition

66

Merupakan proses dalam forensik komputer yang melibatkan pengumpulan data dari memori komputer, terutama RAM, untuk menciptakan snapshot status memori pada titik waktu tertentu.

99



Memory acquisition

TUJUAN

Mendapatkan snapshot atau salinan dari status memori pada suatu titik waktu tertentu. Data yang diperoleh dapat mencakup informasi tentang proses yang berjalan, koneksi jaringan, dan objek memori lainnya.

Volatile Memory Analysis

66

Merupakan proses lanjutan dan befokus pada pemeriksaan dan interpretasi data yang telah diakuisisi untuk memahami aktivitas sistem komputer. Analisis memori volatil melibatkan penggunaan alat dan teknik khusus untuk mengidentifikasi jejak malware, menganalisis proses yang mencurigakan, dan memberikan wawasan terhadap apa yang terjadi.

99

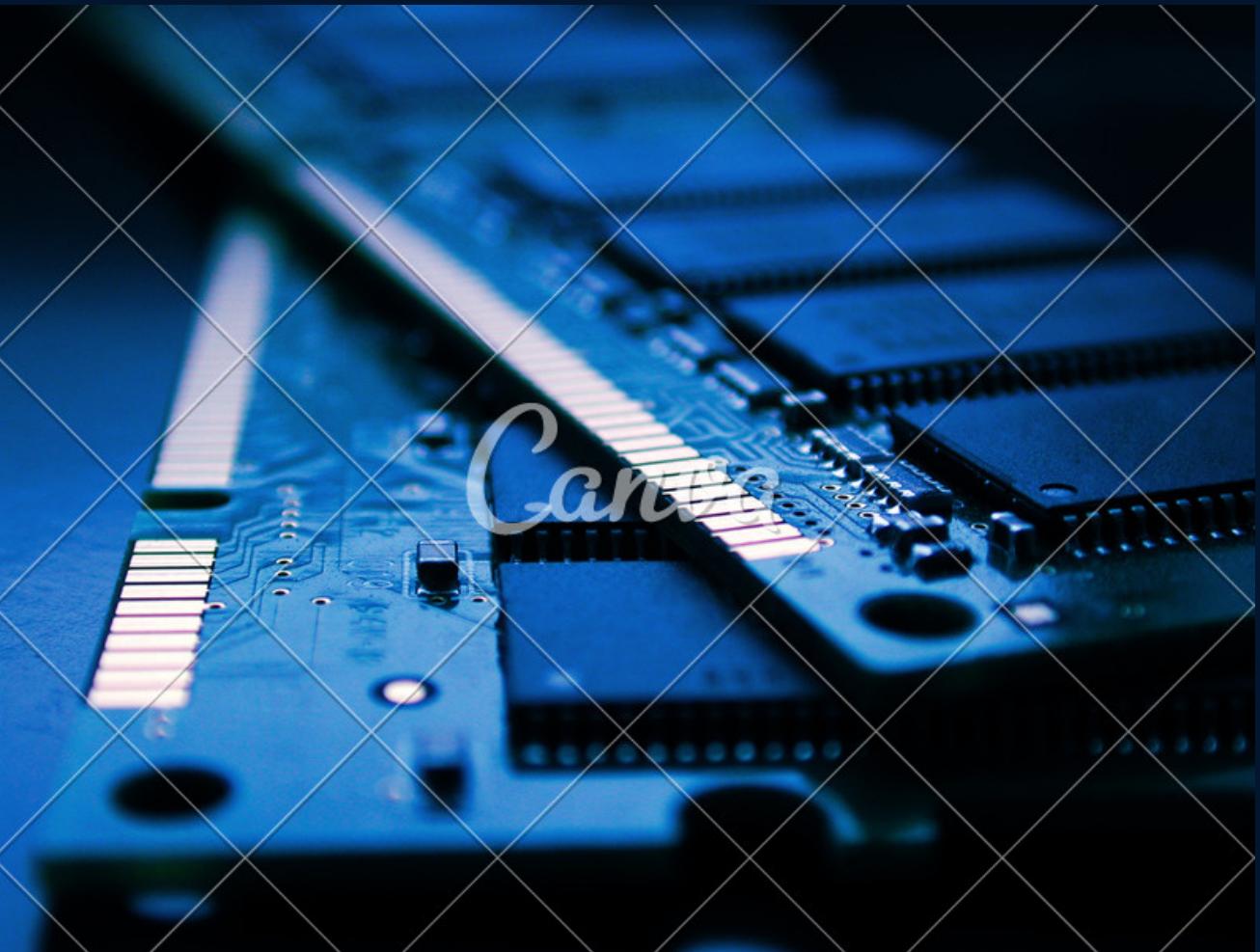


Volatile Memory Analysis

TUJUAN

Memahami aktivitas yang terjadi dalam sistem komputer, termasuk deteksi malware, pemahaman alur kejadian, dan identifikasi tindakan yang mencurigakan.

Memory Acquisition



Memory Acquisition melibatkan penyalinan memori yang volatil ke tempat penyimpanan yang non-volatile.



image memori dianggap benar apabila snapshot atau salinan yang didapati hanya berisi nilai-nilai yang ada di dalam memori saat salinan tersebut diambil.



image memori juga dianggap berintegritas apabila nilai-nilai yang berada di dalam memori tidak berubah sejak waktu tertentu yang ditentukan oleh ahli forensik.

Hierarchy



Time of Deployment



Pre-Incident

- Higher atomicity dump
- membutuhkan wawasan sebelum terjadinya incident.



Post-Incident

- dapat digunakan setelah terjadinya incident
- lower atomicity dump

Need for Termination



Non-Terminating

- Tidak mengganggu operating system
- Bukti forensic lebih berintegritas



Terminating

- Biasanya sudah terdapat di dalam operating system
- Tidak memerlukan konfigurasi tambahan

Acquisition Techniques



User Level

Menggunakan software emulator. software emulator menimbulkan beban kinerja yang tinggi dan memerlukan tingkat akses yang rendah sehingga sangat rentan terhadap serangan.

Benefit :

Emulator termasuk ke dalam tool yang non-terminating

Problem :

Emulator harus diimplementasikan saat sebelum kejadian karena menjalankan program sambil menyalin data.

Acquisition Techniques



Kernel Level

Terdapat tiga Acquisition Techniques pada kernel level

Mengimplementasikan tools sebagai driver kernel

Tools dapat memantau suatu peristiwa, seperti crash atau hang window. Tools termasuk ke Post-Incident time of deployment

Contohnya : WinPmem, LiMe, ProcDump, WinKD.

Hibernation File

Ketika komputer mengalami hibernasi, sebagian besar memori fisiknya ditulis ke disk dalam file hibernasi untuk penyimpanan saat daya tidak tersedia.

Di Windows, file hibernasi berada di C:\hiberfil.sys

Software Debugger

Debugger dapat diimplementasikan sebelum kejadian dengan meluncurkan debuggee dari debugger, atau setelah kejadian dengan melampirkan debugger ke proses yang sedang berjalan

Contohnya: GNU Project Debugger (GDB) umum digunakan untuk sistem UNIX dan WinDbg atau Visual Studio untuk sistem Windows.

Type 1 Hypervisor

Hypervisor Level

Rootkit dan attack lainnya masih dapat meningkatkan hak akses ke level kernel, sehingga alat akuisisi memori level kernel masih terbilang rentan. Maka dari itu, memory acquisition dari level yang lebih tinggi masih dibutuhkan.

Contoh alat virtualization yang bisa memperoleh guest memory dari level hypervisor :

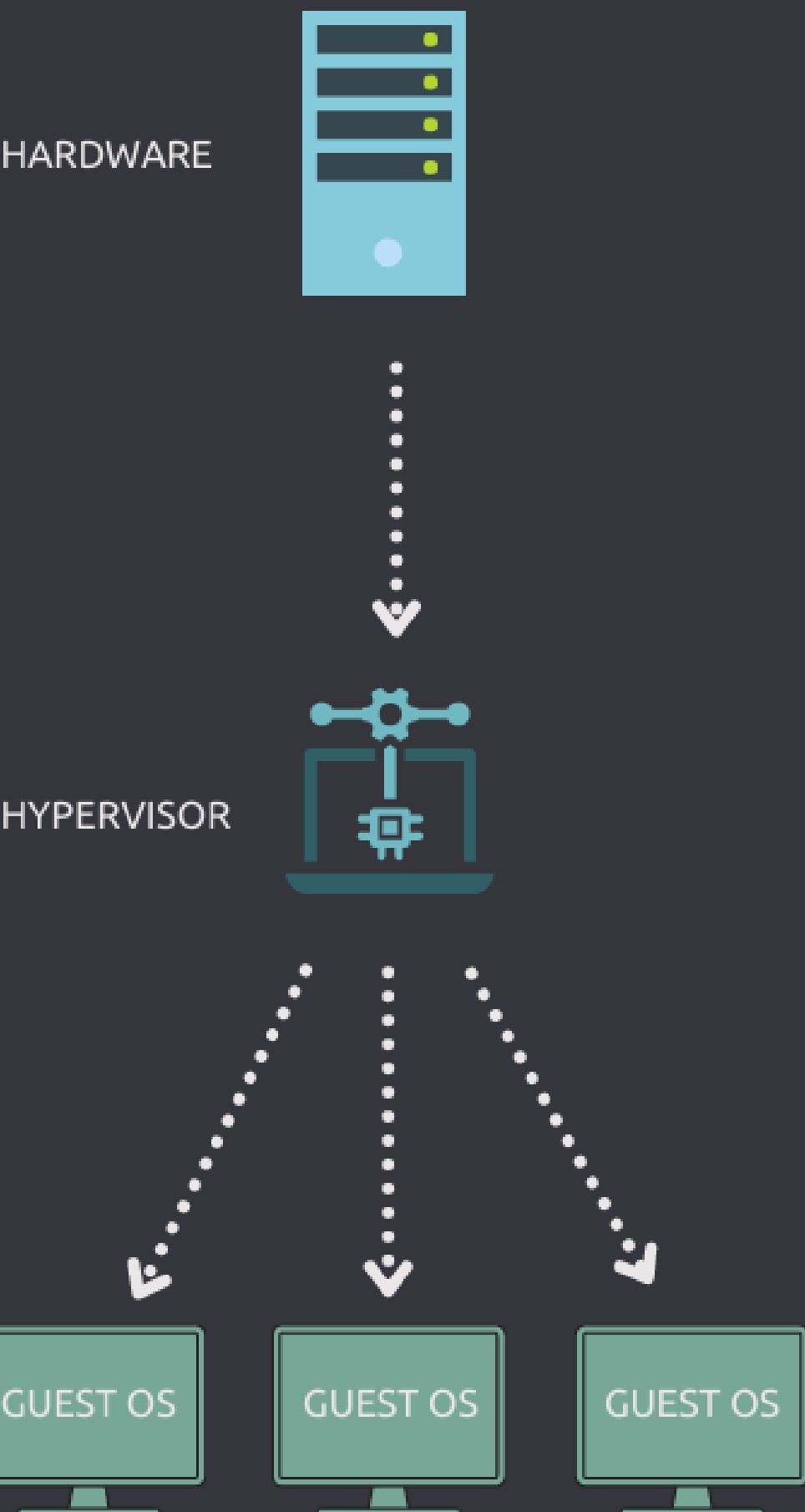
- VMware
- LibVMI

Tools diatas dibangun di tingkat hypervisor, sehingga mereka dapat memanfaatkan akses langsung hypervisor ke guest memory. Tools ini juga dapat menyediakan akuisisi yang lebih terpercaya yang tidak memberikan authorization pada manipulasi oleh sistem operasi guest, tidak hanya itu tools ini juga bisa digunakan untuk analisis lebih lanjut. Namun tools ini biasanya perlu diterapkan sebelum terjadi insiden.

Namun, ada beberapa alat yang bisa memberikan arsitektur yang dapat diterapkan pasca-insiden, seperti :

- HyperSleuth
- Vis
- Cheng et Al

Alat-alat tersebut tidak dapat menganalisis lebih lanjut, namun menggunakan lapisan virtualisasi "tipis" di luar, bahkan sistem operasi host dapat melakukan penjedaan VM dan memory acquisition. Hal ini menyediakan lapisan isolasi tambahan untuk akuisisi memori yang lebih handal.



System Management Level

```
SmmBackdoor.c(591) : *****
SmmBackdoor.c(592) :
SmmBackdoor.c(593) : UEFI SMM access tool
SmmBackdoor.c(594) :
SmmBackdoor.c(595) : by Oleksiuk Dmytro (aka Cr4sh)
SmmBackdoor.c(596) : cr4sh0@gmail.com
SmmBackdoor.c(597) :
SmmBackdoor.c(598) : *****
SmmBackdoor.c(599) :
SmmBackdoor.c(617) : Started as infector payload
SmmBackdoor.c(620) : Image base address is 0xd7024200
SmmBackdoor.c(630) : Resident code base address is 0xd613f00
SmmBackdoor.c(380) : BackdoorEntryResident0 : Started
SmmBackdoor.c(406) : Protocol notify handler is at 0xd613f61
SmmBackdoor.c(640) : Previous calls count is 1
SmmBackdoor.c(657) : Running in SMM
SmmBackdoor.c(681) : SMM system table is at 0xd70069e0
SmmBackdoor.c(536) : SMM protocol notify handler is at 0xd7024200
SmmBackdoor.c(503) : Max. SW SMI value is 0xEF
SmmBackdoor.c(514) : SW SMI handler is at 0xd7024b80
SmmBackdoor.c(369) : ProtocolNotifyHandler0 : Protocol ready
-
```

System Management Level (SML) adalah mode operasi khusus yang ada pada arsitektur/desain sistem komputer, yang independen (terpisah) dari semua operasi sistem normal.

- SML merupakan mode operasi tingkat rendah pada sistem komputer, berjalan paralel dengan sistem operasi utama. SML juga memiliki hak akses lebih tinggi daripada hypervisor dan hanya bisa menangani operasi tingkat rendah seperti BIOS/UEFI, bukan sistem operasi atau aplikasi pengguna, serta independen dan terisolasi dari OS bahkan virtual machines. SML punya akses penuh terhadap akses ke hardware yang tidak dimiliki OS utama atau Hypervisor.

Contoh tools yang digunakan untuk memory acquisition di level ini adalah **SmmBackdoor**.

Berikut penjelasan tentang SmmBackdoor :

- Tool ini memiliki proses instalasi yang kompleks dan hanya spesifik untuk model komputer tertentu.
- Penggunaannya adalah, user menyediakan software langsung ke UEFI untuk "menginfeksi" Modul Manajemen Sistem dan melakukan memory acquisition.
- Tool ini terbatas pada System Management RAM yang berada di Modul Manajemen Sistem



Asynchronous Device Level

Memory acquisition pada level ini dibantu oleh hardware atau bisa dibilang acquisition tingkat ADL, membutuhkan hardware untuk menangkap memory.

Berikut merupakan sifat dari ADL :

- Tidak menggunakan sistem operasi atau CPU dari komputer target, sehingga lebih aman dari manipulasi atau pemalsuan oleh malware.
- Mengakses langsung memori fisik komputer menggunakan Direct Memory Access (DMA) melalui koneksi peripheral seperti PCI Express.
- Memerlukan perangkat keras tambahan eksternal yang terhubung ke slot PCIe target. Contoh perangkat Keras ADL adalah PCILeech dan Inception.
- Bekerja secara asynchronous (tidak sinkron) terhadap sistem operasi dan CPU dari komputer target.
- Memungkinkan akuisisi memori fisik sistem pada kondisi pasca-insiden atau hidup tanpa menghentikan sistem target.

Contoh tools dan metode yang digunakan untuk memory acquisition pada level ini adalah :

- PCILeech
- Inception
- Snipsnap
- Cold Boot

PCI Leech dan Inception

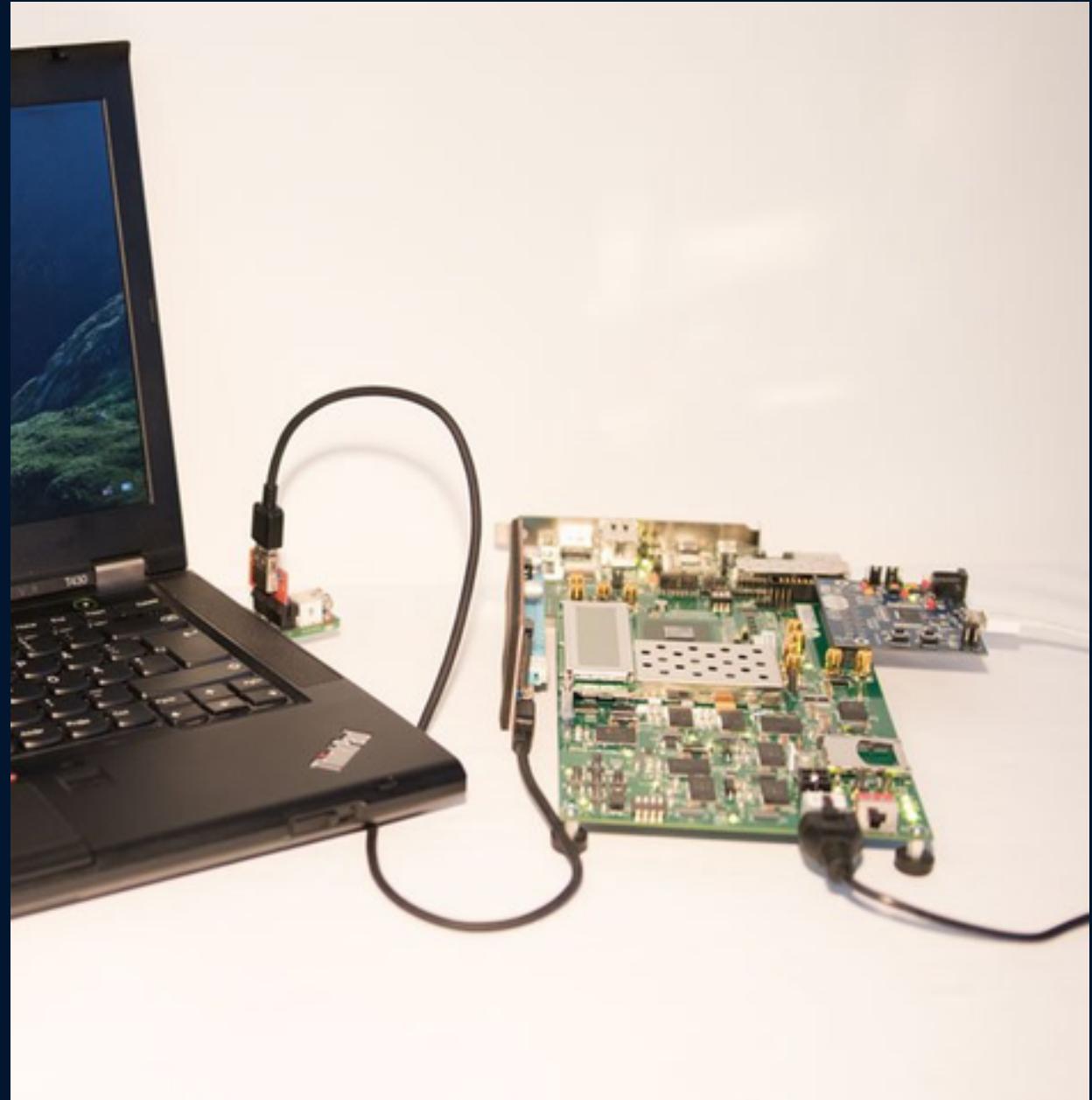
PCI Leech dan Inception merupakan framework Direct Memory Access (DMA) yang dapat diterapkan pasca-insiden, bersifat non-terminating. Mereka membutuhkan hardware untuk mengakses memori melalui bus sistem seperti PCIe.

Contoh command line PCILeech :

```
pcileech.exe dump -force -device usb3380://usb=2
```

Command line diatas akan menjalankan tool PCILeech, Melakukan dumping memori fisik sistem menggunakan perangkat keras USB3380 secara paksa walaupun terdapat error, dan mengidentifikasi perangkat keras yang terhubung ke port USB tertentu.

Kekurangan dari tool PCILeech adalah mustahil untuk menangkap snapshot memory atomik.



Snipsnap merupakan salah satu alat akuisisi memori yang bekerja dengan menggunakan hardware Thread Control Block (TCB) di CPU sistem tersebut secara langsung untuk membaca dan menyalin konten memori sistem. Tool ini membutuhkan kernel driver yang tidak aman (untrusted) untuk membantu akses memori level kernel. Snipsnap memerlukan modifikasi kecil pada memory controller internal CPU dan register file untuk memungkinkan akses memori eksternal. Berikut sifat lain dari Snipsnap :

- Non-atomic
- Performance isolation (tidak memperlambat sistem tersebut)
- Dapat diterapkan pasca-insiden



Cold Boot

Merupakan teknik yang memanfaatkan DRAM karena memiliki kebiasaan tidak akan menghapus data walau telah reset atau power cut.

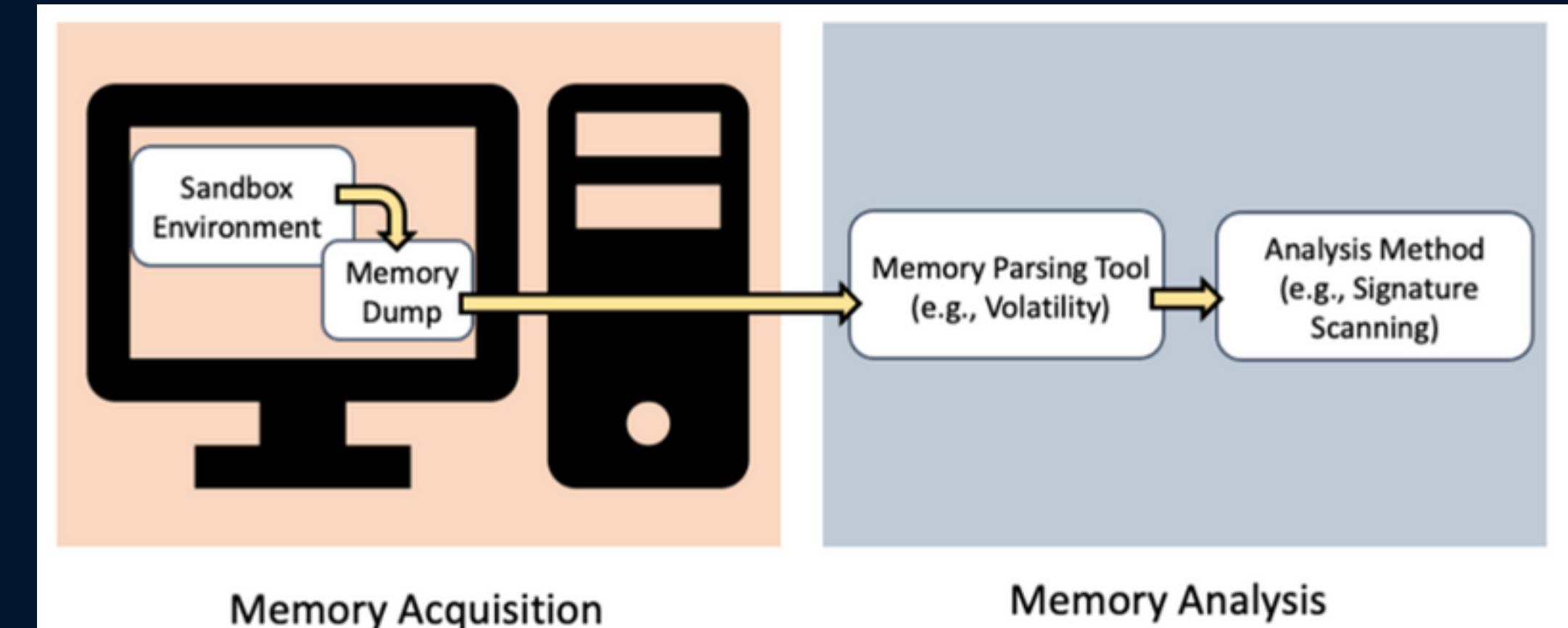
Data yang masih tersimpan saat reset atau power cut adalah frozen memory. Saat cold boot, sangat mungkin untuk menggunakan dumping software seperti **Preboot Execution Environment (PXE)** untuk boot dari USB, atau untuk menambahkan fitur auto-dump memori ke dalam program awal yang berjalan saat booting (BIOS/UEFI) untuk melakukan memory acquisition.

```
GRUB4DOS 0.4.6a 2019-12-30, Mem: 638K/3518M/1483M, End: 3689D4

▶Dump the ram (64-bit Halt)
Dump the ram (64-bit Reboot)
--- INFORMATION: 64-bit CPU ---
Dump the ram - max. 4GB (32-bit Halt)
Dump the ram - max. 4GB (32-bit Reboot)
commandline
reboot
halt
```

Memory Analysis

Setelah memory dump, memory akan di parse dengan *Volatility* atau *Rekall*



Metode Memory Analysis



Traditional (signature scanning / heuristic scanning)



Dynamic (sandboxing i.e. environment terkendali yang nanti sistem terinfeksi malware akan diberikan characteristic untuk mengidentifikasi malware tersebut.)



Machine Learning Techniques ⇒ Mengambil data characteristic dari Dynamic method dan menggunakan untuk melatih algoritma classifier ML.



TOOLS

Memory Analysis (Memory Parsing)

Open Source



Volatility



Rekall

Commercial



Cellebrite Inspector



FireEye Redline



Magnet AXIOM



WindowsSCOPE

Extra Feature: Enterprise-level remote endpoint management with additional analysis

Volatility

“A Python-based framework to analyze memory dumps.”



Pros:

- Kompatibel dengan Windows, Linux, atau Macintosh memory dumps
- Support banyak format memory dump
- Dapat menggunakan GUI dengan plugin Autopsy



Cons:

- Biasanya hanya dijalankan melalui command line.

Contoh:

python3 vol.py -f <dumpfile> windows.pslist

Volatility akan mengekstrak Windows Process List saat dilakukan memory image dump.

Volatility

“A Python-based framework to analyze memory dumps.”



Performance:

Klaim performa Volatility berasal dari developernya sendiri seperti:

- Lebih efisien dari Rekall
- Volatility2 bisa “list[ing] kernel modules from an 80 GB system in just a few seconds”
- Volatility3 beta, developers mengklaim ada banyak perkembangan performance dari Volatility2 dan Rekall dalam berbagai aspek.



Capabilities:

- Ekstrak penanganan (handles) saat ini dan sebelumnya yang sedang menjalankan sistem. (DLLs, command from console shell, memory resident pages, process executables)
- Ekstrak VAD nodes (addresses, tag, flags, control flags, name of the memory mapped file)
- Mendapatkan driver kernel yang dimuatkan (in files or physical memory)
- Mencari process thread objects
- System connections via active TCP connections, listening sockets for any protocol, residual data & artifacts from previous sockets, and network artifacts including TCP endpoints, TCP listeners, UDP endpoints, and UDP listeners.

Rekall

“Awalnya adalah fork dari Volatility di 2011 yang berkembang menjadi framework advance untuk forensic dan incident response. Toolnya memiliki kemiripan dengan Volatility dimana toolsnya dapat menganalisis memory dump dari banyak OS seperti Volatility dan memberikan informasi yang sama. tool ini sudah tidak ada support dari developernya.”

Differences with Volatility



Bawaannya ada GUI



Metode analisis yang memiliki support lebih bagus untuk tiap OS

Secara keseluruhan, hasil dari forensic tools yang disebutkan result perlu diinterpretasikan oleh operator toolsnya yang juga dibantu dengan automation.

Traditional Memory Forensic Approaches

Scanning



Signature Scanning

Scanning method bekerja dengan mencari kesamaan dari memory dump dengan malware sudah ada di database. Pembanding bisa byte/string pattern.

Problem:

Memerlukan banyak storage untuk menyimpan memory dump.

Solutions:

- Gunakan kompresi untuk memori.
- Brengel and Rossow MemScripper method.(lebih efisien dari kompresi)
 - Memory snapshot dari sistem bersih
 - Memory snapshot dari sistem terinfeksi
 - Mencari perbedaan dari memori snapshot
 - Simpan dan kompres perbedaanya

Traditional Memory Forensic Approaches

Scanning



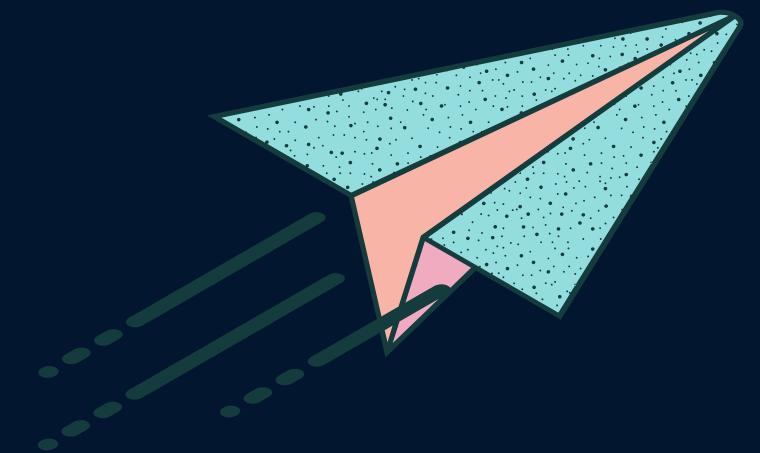
Heuristic Scanning

Heuristik Scanning dilakukan dengan menggunakan rules/algoritma untuk mencari commands/instruksi yang menunjukkan kegiatan malicious. Ini digunakan bersamaan dengan signature scanning.

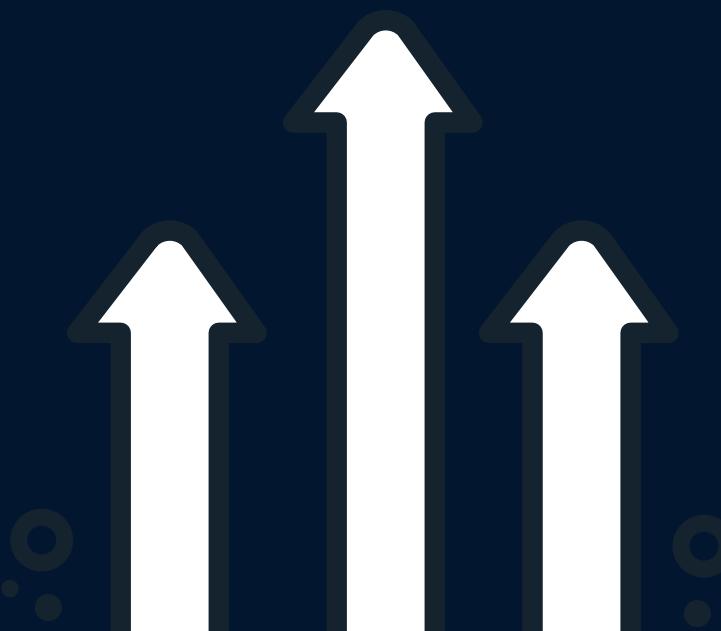
Example:

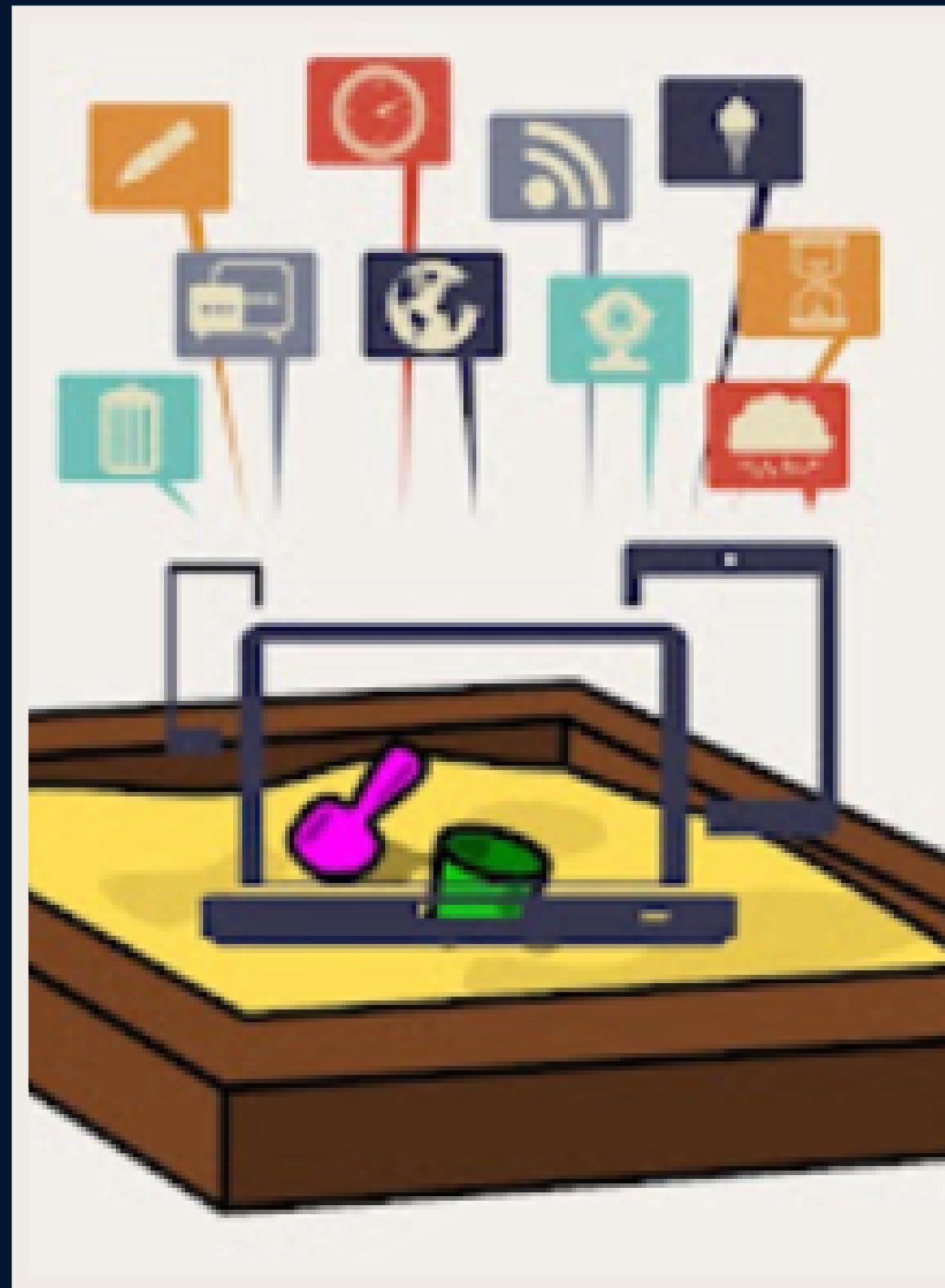
USIM toolkit, bekerja dengan:

- Mencari abstraksi dari OS (namespaces, filesystems, networking and communication channels, environment variables, runtime linkers/loaders, and virtual memory management).
- Hasil abstraksi dikumpulkan menjadi graph yang diatur dengan rules untuk menunjukkan bila ada deviasi dari normal run-time behavior melalui violation of invariants.



Dynamic Analysis dalam Sandbox





Sandboxing dapat memberikan pendekatan dinamis terhadap cyber forensics. Malware diperbolehkan untuk mengeksekusi dalam lingkungan terkendali, yang disebut sandbox, dan perilaku serta karakteristiknya, termasuk informasi tentang memori volatilnya, dapat direkam. Analisis dari informasi yang dikumpulkan di lingkungan sandbox dapat digunakan untuk membantu mengidentifikasi ancaman yang akan datang.

Virtualized Environments

Lingkungan tervirtualisasi, yang disebut mesin virtual, dikendalikan oleh hypervisor. Perangkat lunak hypervisor mengontrol akses berbagai program ke perangkat keras yang mendasarinya, sehingga mesin virtual dapat diisolasi dari mesin virtual atau program lain pada perangkat keras yang sama. Namun, hypervisor dan mesin virtual tidak dapat dijalankan secara bersamaan, sehingga sulit mengumpulkan data detail mengenai eksekusi program. Kehadiran hypervisor juga sulit disembunyikan dari malware, dan pembuat malware diketahui menggunakan teknik pengkodean mengelak yang mengidentifikasi keberadaan hypervisor dan selanjutnya mengubah perilaku program.

Software Emulators

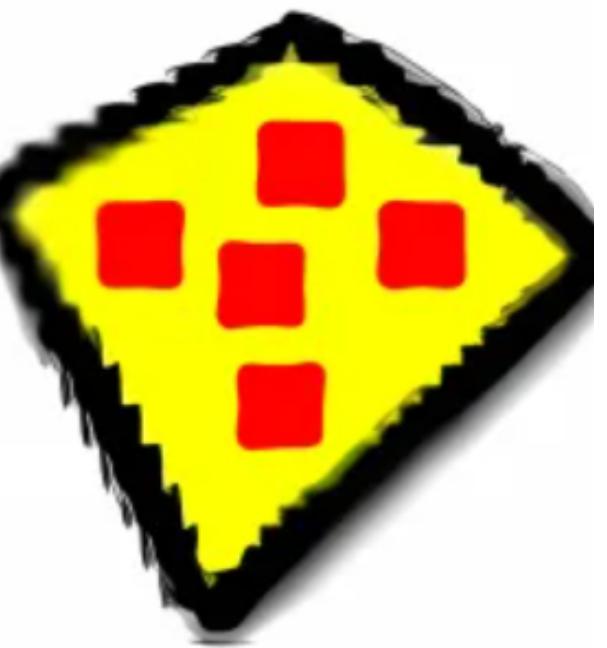
Emulator adalah program perangkat lunak yang mensimulasikan fungsionalitas suatu program atau perangkat keras. Emulator dapat menyimulasikan sistem operasi, namun karena kompleksitas sebagian besar versi modern, seringkali lebih mudah untuk meniru perangkat keras yang mendasarinya. Selain itu, saat program tamu berjalan, analis dapat memperoleh tampilan instruksi demi instruksi tentang apa yang dilakukan malware. Namun, salah satu kelemahan umum emulator eksekusi adalah penalti performa signifikan yang timbul karena penambahan level perangkat lunak. Selain itu, mirip dengan hypervisor, emulator dapat dideteksi dan dilewati oleh malware melalui teknik pengelakan.



Cuckoo Sandbox

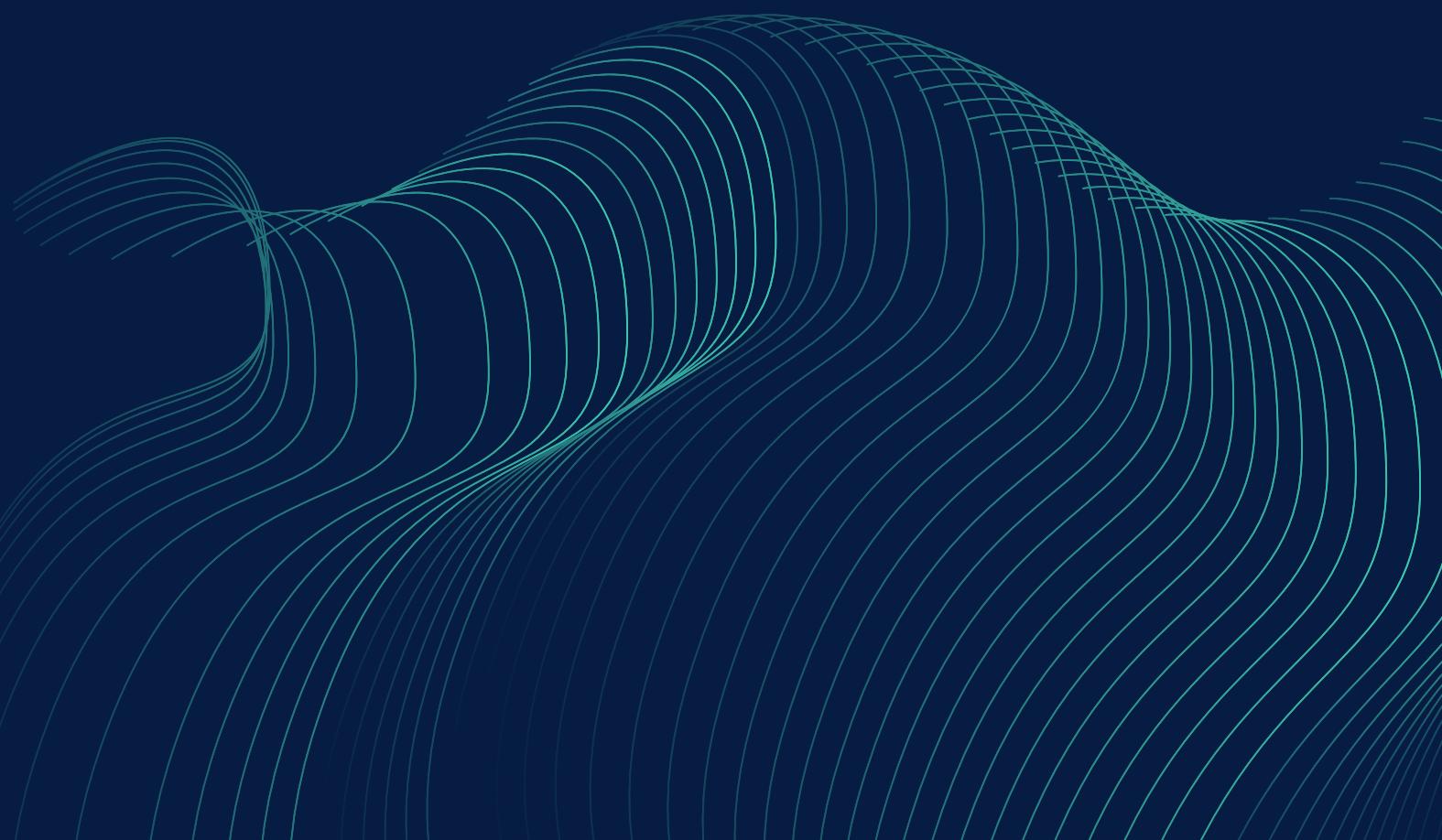


DRAKVUF



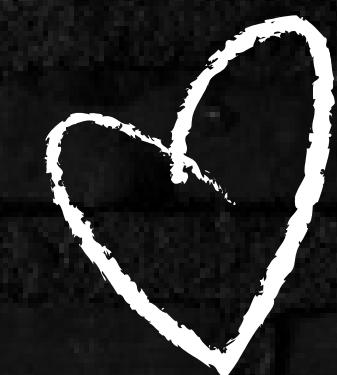
Sandboxie

SANDBOX TOOLS





Conclusions



CONCLUSION 1

Akuisisi memori, sebagai langkah awal, merupakan proses penting dalam mendapatkan snapshot status memori pada titik waktu tertentu. Dengan menggunakan berbagai alat perangkat lunak atau perangkat keras, akuisisi memori memberikan data mentah yang menjadi dasar bagi langkah selanjutnya.

CONCLUSION 2

Analisis memori volatil fokus pada aktivitas sistem komputer, termasuk identifikasi malware, pemahaman alur kejadian, dan deteksi tindakan mencurigakan. Melibatkan alat dan teknik khusus, analisis memori volatil memberikan wawasan mendalam tentang konteks dan implikasi dari data akuisisi.

THANK
YOU