Kelompok:
2501981270 - AGUSTINUS LEONARDO DWITAMA
2501989550 - MUHAMAD DWI APRIYANTO
2502001864 - RAVI DEEVAN SATYAKI
2501965052 - KEVIN MORRIS ARMANDO
2501983723 - NATANAEL FRANSISCO

The Evolution of Volatile Memory Forensics

Introduction

1. Intro

   Akuisisi dan analisis memori volatil merupakan metpde mengidentifikasi ancaman siber terhadap malware tanpa berkas.proses pengumpulan data dari memori komputer, dengan tujuan memperoleh snapshot status memori pada titik waktu tertentu. Semakin meningkat angka populasinya, sulit dideteksi, dan dapat menyebabkan kerusakan yang signifikan. Meskipun data di RAM memiliki nilai forensik tinggi, terutama dengan adanya enkripsi penuh disk.

   Memori volatil menyimpan informasi penting seperti berkas terenkripsi, daftar proses, dan koneksi jaringan. Karena kegunaannya baanyak maka diperlukan pengawasan untuk mengatasi tantangan terhadap memori volatil, memanipulasi informasi data, kinerja lambat, dan subversi malware.

2. Literature Review

   Akuisisi Memori

   Dalam metode akuisisi dan analisis untuk Windows. Dengan akuisisi melalui kedua metode perangkat lunak dan perangkat keras serta, mengakuisisi sistem melalui perangkat lunak dan perangkat keras. Cara operasi malware tanpa berkas dan menguji alat akuisisi memori serta memberikan tinjauan metode akuisisi.

   Analisis Memori Volatil

   Tidak ada tinjauan yang menyeluruh tentang metode analisis memori volatil. Sanjay et al. membahas jenis-jenis malware tanpa berkas dan metode forensik memori. Case dan Richard mencatat kelemahan teknik forensik memori saat ini, termasuk kurangnya metodologi untuk malware userland dan perangkat baru seperti iOS dan IoT.

3. Memory Acquisition

   kualitas memory dump sangat berperan penting untuk menganalisis volatile memory. image memori dianggap benar apabila snapshot atau salinan yang didapati hanya berisi nilai-nilai

yang ada di dalam memori saat salinan tersebut diambil. salinan memori juga dianggap berintegritas apabila nilai-nilai yang berada di dalam memori tidak berubah sejak waktu tertentu yang ditentukan oleh ahli forensik.

3.1. Taxonomy

taksonomi disini merupakan tingkat hierarki akses dimana alat memory acquisition bekerja. hierarkinya terdiri dari user level, kernel level, hypervisor level, synchronous management level (SML), and asynchronous device level (ADL).

3.2. Acquisition Techniques

3.2.1. User Level

Menggunakan software emulator. software emulator menimbulkan beban kinerja yang tinggi dan memerlukan tingkat akses yang rendah sehingga sangat rentan terhadap serangan. emulator harus diimplementasikan saat sebelum kejadian karena menjalankan program sambil menyalin data. Emulator termasuk ke tool yang non-terminating.

3.2.2. Kernel Level

Terdapat tiga teknik akuisisi, pertama mengimplementasikan tools sebagai driver kernel, kedua membuat dump crash atau hibernation file, ketiga debugger yang berjalan di atas executables.

Tools biasanya diimplementasikan setelah kejadian. WinPmem, LiMe, ProcDump, WinKD merupakan tools yang diimplementasikan sebagai driver kernel. tools tersebut dapat memantau suatu peristiwa, seperti crash atau hang window.

Hibernation file merupakan kemampuan bawaan yang dapat digunakan untuk akuisisi memori. Ketika komputer mengalami hibernasi, sebagian besar memori fisiknya ditulis ke disk dalam file hibernasi untuk penyimpanan saat daya tidak tersedia. Di Windows, misalnya, file hibernasi berada di C:\hiberfil.sys.

Debugger dapat diimplementasikan sebelum kejadian dengan meluncurkan debuggee dari debugger, atau setelah kejadian dengan melampirkan debugger ke proses yang sedang berjalan. GNU Project Debugger (GDB) umum digunakan untuk sistem UNIX dan WinDbg atau Visual Studio untuk sistem Windows.

3.2.3. Hypervisor Level

Karena rootkit dan serangan lainnya mampu meningkatkan hak akses ke level kernel, alat memory acquisation yang berjalan pada level kernel masih rentan terhadap subversion. Dalam beberapa kasus, dibutuhkan untuk mengambil memori dari level akses hierarki yang lebih tinggi. Beberapa alat virtualisasi seperti VMware dan LibVMI memiliki fungsionalitas terintegrasi yang mampu mengambil memori guest dari level hypervisor. Alat-alat ini dapat diakses melalui GUI, baris perintah, dan pustaka dengan API yang dapat diakses, meskipun alat-alat ini biasanya harus diterapkan sebelum insiden terjadi. ada beberapa pengecualian seperti HyperSleuth, Vis, dan Cheng et al. Alat-alat ini tidak tersedia untuk analisis lebih lanjut tetapi menawarkan arsitektur yang bergantung pada lapisan virtualisasi tipis yang terletak di luar, bahkan sistem operasi host. Mesin virtual dapat dihentikan sementara, dan memori dikumpulkan melalui alat yang disediakan oleh vendor. VMware

menyediakan vmss2core.exe, dan serupa dengan itu LibVMI dikemas dengan dump-memory.

3.2.4. System Management Level

System Management Level adalah mode operasi yang hadir pada arsitektur dari sistem yang independen dari semua operasi sistem normal. Level manajemen sistem hanya bisa menangani pada operasi tingkat rendah seperti Basic Input/Output System dan Unified Extensible Firmware Interface (BIOS/UEFI), bukan sistem operasi atau aplikasi pengguna, dan memiliki hak akses yang lebih tinggi daripada hypervisor. Karena sifat independen dari sistem operasi dan mesin virtual, pengambilan memori pada level ini merupakan sebuah pilihan yang menarik. Namun, ada sedikit implementasi alat pengambilan pada level ini. SmmBackdoor adalah contoh langka dari alat tersebut, tetapi proses instalasinya kompleks dan cukup spesifik untuk model komputer tertentu. Untuk menggunakan SmmBackdoor, user menyediakan software langsung ke sistem UEFI untuk "menginfeksi" Modul Manajemen Sistem dan mengambil memori. Pembuat alat ini bermaksud bahwa alat ini dijadikan sebagai contoh eksploitasi dibandingkan sebagai alat forensik. Smm Backdoor terlimitasi di scope memory ke System Management RAM yang berada di Modul Manajemen Sistem

3.2.5. Asynchronous Device Level

Memory acquisition pada level ini dibantu oleh hardware atau bisa dibilang acquisition tingkat ADL, membutuhkan hardware untuk menangkap memory. PCILeech dan Inception adalah framework akses memori langsung (Direct Memory Access / DMA) yang dapat diterapkan pasca insiden dan bersifat non-terminating. Mereka menggunakan perangkat keras eksternal untuk mengakses memori melalui bus sistem, seperti peripheral component interconnect express (PCIe). Alat-alat ini membutuhkan penyediaan driver untuk hardware penangkap dan software alat itu sendiri untuk berinteraksi dengan hardware tersebut. Setelah hal diatas telah ada, PCILeech dapat diaktifkan melalui command line, contohnya :

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

Karena sistem berjalan tanpa adanya gangguan, mustahil untuk menangkap snapshot memory atomik. Alat lainnya, Snipsnap, menggunakan hardware Thread Control Block (TCB) dan kernel driver yang untrusted yang dapat mengambil memory dalam OS target. Metode ini butuh sedikit modifikasi pada on-chip memory controller dan CPU register file, yang mana membuat metode ini non-atomic. Namun, metode ini menawarkan isolasi performa bagi aplikasi yang berjalan pada sistem target. Cara hardware-based lainnya adalah cold boot technique, yang memanfaatkan sifat DRAM yang tidak akan dihapus setelah reset atau power cut. Untuk melakukan dump pada frozen memory saat cold boot, sangat mungkin untuk menggunakan dumping software menggunakan Preboot Execution Environment (PXE), untuk boot dari drive USB, atau untuk menyertakan rutin dumping dalam BIOS/UEFI.

4. Memory Analysis

Setelah memory dump, memory akan di parse dengan *Volatility* atau *Rekall.* Setelah itu akan dianalisis dengan beberapa metode:

- 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 menggunakannya untuk melatih algoritma classifier ML.

4.1. Tooling

Tools untuk parsing memory:

(Open Source)

- Volatility
- Rekall

(Commercial)

- Cellebrite Inspector
- FireEye Redline
- Magnet AXIOM
- WindowsSCOPE

Fitur tambahan milik commercial tools: Enterprise-level remote endpoint management with additional analysis (Memberikan kemampuan untuk mengatur dan menganalisis memory beberapa device dari satu sentral)

4.1.1. Volatility

Tools forensic memory dump analysis berbasis framework python.

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.*

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.

4.1.2. 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.

Perbedaan dengan *Volatility:*

- Bawaanya ada GUI
- Metode analisis yang memiliki support lebih bagus untuk tiap OS

4.1.3. Discussion

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

4.2 Traditional Memory Forensic Approaches

Traditional forensic methods ada 2 macam yaitu, scanning dan dynamic analysis.

4.2.1 Scanning Methods

Scanning method adalah cara untuk mencari bukti infeksi dengan mencari files, memory dumps, process lists, network connections. Ini dianggap sebagai metode efektif karena prosesnya cepat dan dalam banyak kasus malware yang dihadapi biasa sudah ada sebelumnya. Scanning method ada 2 jenis yaitu Signature Scanning dan Heuristic Scanning.

Signature Scanning

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

Metode de facto untuk scanning sekarang adalah dengan YARA database yang membandingkan memory dump dengan suatu tipe malware/keluarga malware.

Scanning techniques:

- Scan virtual address space atau memory address langsung.
- Pakai Windows Page Frame Number (PFN) untuk identifikasi pemilik tiap physical page.

Problem: memerlukan banyak storage untuk menyimpan memory dump.

Solutions:

- Gunakan kompresi untuk memori.
- Brengel and Rossow *MemScrimper* method.(lebih efisien dari kompresi)

1. Memory snapshot dari sistem bersih
2. Memory snapshot dari sistem terinfeksi
3. Mencari perbedaan dari memori snapshot
4. Simpan dan kompres perbedaanya

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.

4.2.2. Dynamic Analysis within a 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. Sandbox adalah aspek penting dari teknik forensik dinamis, karena dapat melindungi sistem analis dari infeksi yang tidak

diinginkan. Proses bare-metal, malware yang berjalan di luar sandbox, secara praktis tidak dapat diskalakan karena memerlukan instalasi ulang sistem secara penuh setiap selesai menjalankannya. malware dijalankan untuk memulihkan sistem analis ke keadaan semula. Ada dua pendekatan utama yang digunakan untuk menyiapkan sandbox, virtualization dan emulation, dan mereka berbeda dalam cara menciptakan lingkungan terkendali.

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. Karena emulator mengimplementasikan fungsinya melalui perangkat lunak, emulator ini sangat fleksibel. Emulator dapat dibuat untuk menjalankan program tamu pada arsitektur CPU perangkat keras yang benar-benar berbeda dari yang dirancang untuknya. 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.

Sandbox Tools

Beberapa alat sandbox sumber terbuka dan gratis mencakup Cuckoo, DRAKVUF, Sandboxie, dan SpeakEasy. Hampir semua upaya penelitian deteksi malware berbasis perilaku menggunakan semacam sistem sandbox. Antarmuka pengguna dengan sistem ini—menyediakan program dan konfigurasi untuk menguji dan mengambil data untuk dianalisis—dalam berbagai cara, mulai dari utilitas baris perintah hingga permintaan web (melalui GUI web atau API HTTP) dan GUI lokal. Misalnya saja tipikal permintaan pembuatan tugas untuk instance Cuckoo yang berjalan di “<server>” mengambil bentuk berikut:

curl -H "Task Name" -F file=@/program http://<server>/tasks/create/file

4.3. Machine Learning Approaches

Penggunaan pendekatan berbasis Machine Learning (ML) dalam deteksi malware telah meluas dalam beberapa tahun terakhir. Hal ini tidak mengherankan karena kesuksesan besar

yang dicapai algoritma tersebut dalam masalah klasifikasi di berbagai domain. Di bagian ini, kami mengeksplorasi upaya penggunaan algoritma ML dalam bidang deteksi malware, khususnya dengan forensik memori volatil. Pendekatan ML ini dapat diklasifikasikan menjadi dua kelompok, pendekatan rekayasa fitur dan pendekatan berbasis visi komputer.

Dalam bidang forensik memori volatile berbasis machine learning, berbagai pendekatan rekayasa fitur dan visi komputer telah dieksplorasi untuk mendeteksi malware. Rekayasa fitur seringkali melibatkan eksekusi malware di kotak pasir, mendapatkan dump memori, dan mengekstraksi fitur menggunakan alat seperti Volatilitas. Sebuah studi menggunakan pengklasifikasi seperti optimasi minimal sekuensial, random forest, decision tree, naïve Bayes, dan pengklasifikasi berbasis instans dengan 130 fitur yang diekstraksi dari pohon VAD, pemetaan file, dan informasi registry, sehingga mencapai keberhasilan dalam deteksi malware.

Pendekatan lain berfokus pada mengekstraksi fitur yang terkait dengan panggilan API, injeksi DLL, perubahan registry, dan koneksi jaringan. Algoritma klasifikasi seperti naïve Bayes, pengelompokan vektor dukungan, K-nearest neighbours, regresi logistik, decision tree, random forest, dan analisis diskriminan linier digunakan, menghasilkan akurasi lebih dari 93% dalam mendeteksi proses berbahaya ketika jaringan saraf berulang diterapkan.

Ekstraksi fitur dari dump memory juga telah digunakan untuk mengidentifikasi keberadaan ransomware menggunakan algoritma klasifikasi XGBoost, sehingga mencapai akurasi 89% terhadap kumpulan data terbatas.

Selain itu, teknik visi komputer diterapkan dengan merepresentasikan data sebagai gambar atau struktur mirip gambar. Sebuah penelitian menggunakan histogram tipe akses memori acak, yang mencapai akurasi hampir 100% dalam mendeteksi rootkit kernel dan akurasi 88% dalam memprediksi malware yang merusak memori tingkat pengguna, dengan tingkat positif palsu yang rendah.

Pendekatan inovatif lainnya menjadikan dump memori sebagai gambar dan menerapkan deskriptor fitur visual (GIST dan HOG) untuk menghitung fitur. Klasifikasi menggunakan random forest, XGBoost, linear SVM, optimasi minimal sekuensial, dan J48 mencapai akurasi hingga 96% dalam memprediksi keluarga malware. Namun, penelitian ini memiliki kumpulan data terbatas yang hanya mencakup 10 keluarga malware berbeda.

Secara keseluruhan, studi-studi ini menyoroti potensi machine learning dalam forensik memori volatile, dengan rekayasa fitur dan teknik visi komputer yang menjanjikan dalam mendeteksi berbagai jenis malware.

5. Conclusion

Penelitian akuisisi dan analisis volatile memory berkembang dengan cepat karena telah terbukti menjadi metode forensik yang berharga. Berbagai alat akuisisi memori telah diciptakan untuk sistem operasi utama, tetapi alat ini memiliki akurasi yang berbeda-beda, kecepatan, dan kepraktisan.

Volatilitas telah memantapkan dirinya sebagai alat ekstraksi memori terkemuka dan digunakan bersama dengan sebagian besar metode forensik memori oleh para peneliti. Metode forensik memori dapat diklasifikasikan sebagai analisis dinamis dari dalam sandbox, metode pemindaian, atau pendekatan machine learning. Metode yang menggunakan sandbox mengkarakterisasi perilaku malware lebih baik daripada metode pemindaian, namun mungkin tidak efektif terhadap malware yang berisi kode penghindaran kotak pasir. Metode pemindaian cepat diterapkan dan sudah banyak digunakan dalam perangkat lunak komersial, namun gagal mengidentifikasi malware dan malware yang sebelumnya tidak terlihat hanya memberikan gambaran terbatas tentang perilaku malware. Penggunaan algoritma klasifikasi machine learning untuk forensik volatile memory menunjukkan hasil yang menjanjikan, tetapi dalam banyak kasus, hasil ini memerlukan verifikasi pada kumpulan data yang lebih besar.