UNIVERZITA PAVLA JOZEFA ŠAFÁRIKA V KOŠICIACH
PRÍRODOVEDECKÁ FAKULTA

AUTOMATIZÁCIA FORENZNEJ ANALÝZY OPERAČNEJ PAMÄTE

2024 **Adam URBAN**

UNIVERZITA PAVLA JOZEFA ŠAFÁRIKA V KOŠICIACH
PRÍRODOVEDECKÁ FAKULTA

AUTOMATIZÁCIA FORENZNEJ ANALÝZY OPERAČNEJ PAMÄTE

BAKALÁRSKA PRÁCA

Študijný program:	Aplikovaná informatika
Pracovisko (katedra/ústav):	Ústav informatiky
Vedúci bakalárskej práce:	RNDr. Eva Marková
Konzultant bakalárskej práce:	doc. RNDr. JUDr. Pavol Sokol, PhD.

Košice 200924

Adam URBAN

Univerzita P. J. Šafárika v Košiciach
Prírodovedecká fakulta

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta:	Adam Urban
Študijný program:	aplikovaná informatika (jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor:	Informatika
Typ záverečnej práce:	Bakalárska práca
Jazyk záverečnej práce:	slovenský
Sekundárny jazyk:	anglický

Názov: Automatizácia forenznej analýzy operačnej pamäte

Názov EN: Automation of digital forensics of operational memory

Cieľ: (1) Porovnanie aktuálnych prístupov k forenznej analýze operačnej pamäte.
(2) Analýza a spracovanie forenzných artefaktov obsiahnutých v operačnej pamäti.
(3) Návrh, implementácia a overenie nástroja na automatizáciu forenznej analýzy operačnej pamäte.

Literatúra:
(1) Latzo, Tobias, Ralph Palutke, and Felix Freiling. "A universal taxonomy and survey of forensic memory acquisition techniques." Digital Investigation 28 (2019): 56-69.
(2) Nyholm, Hannah, et al. "The Evolution of Volatile Memory Forensics." Journal of Cybersecurity and Privacy 2.3 (2022): 556-572.
(3) Case, Andrew, and Golden G. Richard III. "Memory forensics: The path forward." Digital investigation 20 (2017): 23-33.
(4) Iqbal, Salman, and Soltan Abed Alharbi. "Advancing automation in digital forensic investigations using machine learning forensics." Digital Forensic Science. IntechOpen, (2019).

Vedúci:	RNDr. Eva Marková
Konzultant:	doc. RNDr. JUDr. Pavol Sokol, PhD.
Oponent:	RNDr. Richard Staňa
Ústav :	ÚINF - Ústav informatiky
Riaditeľ ústavu:	doc. RNDr. Ondrej Krídlo, PhD.

Dátum schválenia: 14.05.2024

Poďakovanie

Rád by som nesmierne poďakoval svojej vedúcej práce RNDr. Eve Markovej, za veľkú pomoc pri vytváraní tejto práce, či už formou pripomienok, usmernení, motivácie, ale aj trpezlivosti a podpore. Taktiež by som veľmi rád poďakoval konzultantovi práce, doc. RNDr. JUDr. Pavlovi Sokolovi, PhD. za odborné usmernenia a postupy počas tvorby tejto práce.

Abstrakt v štátnom jazyku

Forenzná analýza operačnej pamäte (RAM) je dôležitou súčasťou vyšetrovania kybernetických trestných činov. RAM môže obsahovať cenné informácie o aktivitách, ktoré sa uskutočnili v počítači, ako sú spustené procesy, otvorené súbory a navštívené webové stránky. Analýza RAM môže pomôcť pri identifikácii páchateľov, pri odhaľovaní prípadov a pri zhromažďovaní dôkazov. V našej práci sme sa zamerali na zjednodušenie a automatizáciu forenznej analýzy, ktorou vieme pomôcť pri analýze podozrivých artefaktov so zameraním na operačný systém Windows. Ako modelové prípady používame dva druhy zaistených pamätí z portálu MemLabs, ďalším modelovým obrazom operačnej pamäte bol prípad ukradnutej Sečuánskej omáčky. Našim posledným obrazom bol nami vytvorený obraz operačnej pamäte nášho zariadenia. Na týchto obrazoch operačnej pamäte sme testovali nami vytvorený nástroj na automatizáciu forenznej analýzy operačnej pamäte.

Pri tvorení nástroja sme vyberali najvhodnejšie technológie na zaistenie operačnej pamäte a analýzu operačnej pamäte.

Kľúčové slová: artefakt, forenzná analýza, operačná pamäť

Abstrakt v cudzom jazyku

Forensic analysis of operational memory (RAM) is an important part of cybercrime investigations. RAM can contain valuable information about the activities that have taken place on a computer, such as running processes, open files, and websites visited. Analyzing RAM can help in identifying perpetrators, solving cases, and gathering evidence. In our work, we have focused on simplifying and automating forensic analysis that can help in analyzing suspicious artifacts with a focus on Windows operating system. We use two types of seized memory from MemLabs as model cases; another model operating memory image was the case of the stolen Szechuan Sauce. Our last image was an image we created of our device's operating memory. We tested the tool we created to automate forensic analysis of RAM on these RAM images.

In creating the tool, we selected the most appropriate technologies to acquire the RAM and analyze the RAM.

Key words: artifact, forensic analysis, operational memory

Obsah

Zoznam ilustrácií

Zoznam tabuliek

Zoznam skratiek a značiek

RAM	Random Access Memory, pamäť s náhodným prístupom
OS	operačný systém
SSL	Secure Sockets Layer, technológia na šifrovanie informácií
BIOS	Basic Input Output System, základný program počítača
EEPROM	Electrically Erasable Programmable Read-Only Memory, elektricky mazateľná pamäť ROM
NVRAM	Non-Volatile Random Access Memory, energeticky nezávislá pamäť s priamym prístupom
PCI	Peripheral Component Interconnect, štandard pre zbernicu počítača k pripojeniu periférnych zariadení k matičnej doske
iSCSI	Internet Small Computer System Interface, sieťový protokol, ktorý pripája úložný priestor pomocou počítačovej siete
XML	eXtensible Markup Language, rozšíriteľný značkovací jazyk
RC4	kryptografický algoritmus pre prenos dát
USB	Universal Serial Bus, univerzálna sériová zbernica
DNS	Domain Name System, prekladá názvy domén na IP adresy
ID	Identity Document, identifikačné číslo
TCP	Transmission Control Protocol, protokol pre prenos informácií medzi počítačmi
UDP	User Datagram Protocol, protokol pre prenos informácií medzi počítačmi
MFT	Managed File Transfer, technológia pre bezpečný prenos údajov
NTFS	New Technology File System, súborový systém
ICMP	Internet Control Message Protocol, protokol pre posielanie chybových správ operačným systémom
CSV	Comma-Separated Values, súborový formát na ukladanie tabuľkových dát

Úvod

V dnešnej digitálnej ére je počítačová kriminalita na vzostupe, čo kladie vysoké nároky na forenzné tímy a ich schopnosť efektívne získavať a analyzovať digitálne dôkazy. Jednou z kľúčových oblastí digitálnej forenznej analýzy je operačná pamäť, ktorá obsahuje cenné informácie o stave systému, bežiacich procesoch a dočasných údajoch, tzv. forenzných artefaktoch, ktoré môžu byť kľúčové pre vyšetrovanie. Automatizácia forenznej analýzy operačnej pamäte prináša potenciál zrýchliť a zefektívniť procesy, čím umožňuje forenzným analytikom rýchlejšie a presnejšie identifikovať kritické informácie.

Cieľom tejto záverečnej práce bolo preskúmať súčasné nástroje a techniky na forenznú analýzu operačnej pamäte a navrhnúť nový nástroj, ktorý by tieto procesy automatizoval. Práca sa skladá z niekoľkých kapitol, ktoré postupne predstavia teoretické základy forenznej analýzy, detailne popíšu súčasné nástroje, a nakoniec sa zamerajú na návrh a vyhodnotenie nového nástroja na automatizáciu.

V prvej kapitole tejto práce sme sa zamerali na základné princípy a metodológie forenznej analýzy s osobitným dôrazom na operačnú pamäť. Rozobrali sme rôzne techniky zaisťovania operačnej pamäte a prebrali nástroje, ktoré sa v tomto procese používajú.

Druhá kapitola ponúkla prehľad a porovnanie najpopulárnejších nástrojov, ako sú Volatility 2, Volatility 3 a Rekall. V tejto časti sme sa zamerali na ich schopnosti, výhody a nevýhody, a porovnávame ich účinnosť v rôznych situáciách.

V tretej kapitole, sme sa sústredili na praktický návrh nového nástroja na automatizáciu analýzy operačnej pamäte. Popísali sme použité dáta a technológie, ktoré sme zvolili na implementáciu, a vysvetlili sme, ako boli integrované funkcie generovania CSV súborov a vizualizácia výsledkov.

Touto prácou sme prispeli k rozvoju oblasti forenznej analýzy tým, že práca ponúka nový prístup k automatizácii procesov, čo môže zlepšiť nielen efektivitu, ale aj spoľahlivosť výsledkov. Veríme, že naše zistenia a navrhované riešenia budú hodnotným prínosom pre forenzných analytikov v oblasti kybernetickej bezpečnosti.

1 Forenzná analýza

Forenzná analýza predstavuje detailné vyšetrovanie a dokumentáciu udalostí s cieľom objektívne identifikovať vinníkov, ich motivácie, priebeh a dôsledky bezpečnostných incidentov či porušení zákonov. Ide o precízne ladený proces, v ktorom forenzný analytik systematicky analyzuje stopy, aby jasne určil, kto, ako a kedy sa podieľal na danom priestupku alebo trestnom čine. Táto disciplína úzko spolupracuje s právnymi procesmi, najmä v oblasti trestného práva, a využíva široké spektrum technologických prostriedkov a postupov na zhromažďovanie a interpretáciu stôp.

V praxi slúži forenzná analýza na získanie stôp týkajúcich sa trestných činov, zneužitia právomocí, porušenia zákonov alebo interných pravidiel, a pomáha overiť identitu osôb, pravosť dokumentov a údajov. Používa sa aj na riešenie bezpečnostných incidentov a následné uplatnenie nárokov na náhradu škody. Táto analytická metóda nachádza uplatnenie v rôznych odvetviach, od kriminalistiky po interne vyšetrovanie vo firmách, pričom každé odvetvie má svoje špecifické metódy, vrátane analýzy účtovníctva, vyšetrovania počítačových sietí, forenznej analýzy operačnej pamäte, ktorej sa budeme v našej práci venovať alebo mobilných zariadení, ako aj audio a video forenznej analýzy. proces môžu realizovať interní forenzní analytici alebo externé špecializované firmy, pričom sa opiera o znalosti z oblasti forenznej vedy a riadenia bezpečnosti a zohráva kľúčovú úlohu pri zabezpečovaní spravodlivosti a ochrane proti rôznym formám podvodov a zneužitia [1].

Digitálna forenzná analýza je dôležitou súčasťou reakcie na incidenty. Práve aplikácia metód digitálnej forenznej analýzy často umožňuje pracovníkom zodpovedným za riešenie bezpečnostných incidentov získať jasnú predstavu o postupnosti udalostí, ktoré viedli k škodlivému konaniu, ako je napríklad kompromitácia servera alebo iné narušenie ochrany údajov. V prípade iných incidentov, ako sú interné podvody alebo škodlivé aktivity zainteresovaných osôb, môže digitálna forenzná analýza pomôcť poukázať na vinníka. Pred podrobným skúmaním nástrojov a techník, ktoré majú k dispozícii pracovníci reagujúci na incidenty, je veľmi dôležité zaoberať sa základnými prvkami digitálnej forenznej analýzy. Tieto prvky poskytujú nielen kontext konkrétnych činností, ale aj metódu na zabezpečenie užitočnosti stôp, ktoré sú súčasťou vyšetrovania incidentu [2].

1.1 Forenzná analýza operačnej pamäte

Forenzná analýza operačnej pamäte je proces skúmania digitálnych stôp v operačnej pamäti zariadení, ktorého cieľom je odhaliť relevantné informácie, identifikovať postup možného páchateľa a vykonštruovať udalosti súvisiace s vyšetrovaním. Tento proces je kľúčovou súčasťou boja proti kybernetickým hrozbám, pretože umožňuje identifikáciu, zmiernenie a odstránenie potenciálnych hrozieb z pohľadu operačnej pamäti. Okrem toho sa forenzná analýza operačnej pamäte osvedčuje aj v poskytovaní dôležitých informácií orgánom činným v trestnom konaní po kybernetickom útoku.

Forenzná analýza operačnej pamäte (RAM) je pre forenzné vyšetrovanie dôležitá z niekoľkých dôvodov. Operačná pamäť je základným zdrojom informácií pri vyšetrovaní digitálnej kriminalistiky. Obsahuje údaje, ktoré nie sú k dispozícii v iných zdrojoch, ako sú obrazy disku alebo sieťové zábery. Ďalšou pridanou hodnotou forenznej analýzy výpisov pamäte RAM je poskytnutie dôležitých informácií o správaní malvéru, prípadne iného škodlivého softvéru. Analýzou obsahu pamäte môžu forenzní analytici určiť činnosti, ktoré vykonal škodlivý softvér, a to napríklad súbory, ktoré vytvoril alebo upravil, sieťové spojenia, ktoré nadviazal, a údaje, ktoré infiltroval.

Forenzná analýza pamäte RAM sa môže napokon využívať na vyšetrovanie vnútorných hrozieb. Útočník vo vnútri systému môže na vykonanie svojho útoku použiť operačnú pamäť, napríklad pre spustenie škodlivého kódu v pamäti alebo na krádež údajov z pamäte. Analýzou obsahu pamäte môžu forenzní analytici identifikovať aktivity útočníka vo vnútri systému a údaje, ku ktorým získal prístup, ktoré pozmenil, alebo ukradol [3].

Získavanie a analýza operačnej pamäte na identifikáciu kybernetických hrozieb je v súčasnosti aktívnou oblasťou výskumu v oblasti kybernetickej bezpečnosti. Význam, ktorý počítačové systémy zohrávajú v modernom živote, neustále rastie a spolu s ním aj kreativita a schopnosti tých, ktorí chcú k nim získať nezákonný prístup. Podniky v súčasnosti zažívajú o 50 % viac kybernetických útokov týždenne v porovnaní s rokom 2020 [4]. Najmä škodlivý softvér, ktorý využíva na infikovanie počítača legitímne programy a nezanecháva žiadnu stopu v súborovom systéme, je stále rozšírenejší a často sa dokáže vyhnúť antivírusovému softvéru. V skutočnosti sa predpokladá, že malvér, ktorý je priam neviditeľný v súborovom systéme je 10-krát úspešnejší ako iné typy malvéru pri vyhýbaní sa detekcii [5]. Z tohto dôvodu zostávame v nevedomosti o úplnom

rozsahu škôd, ktoré malvér bez súborov spôsobuje. Keďže je používanie bezsúborového malvéru rozsiahle, forenzná analýza pamäte môže aj naďalej predstavovať hlavný pilier forenzných metód. Operačná pamäť obsahuje artefakty obsahu zašifrovaných súborov, zoznamy spustených procesov a zoznamy sieťových spojení. Vzhľadom na rastúce používanie úplného šifrovania disku a ďalších ochranných opatrení je oveľa ťažšie a často nemožné získať takéto informácie zo súborového systému [6].

1.2 Zaisťovanie operačnej pamäte

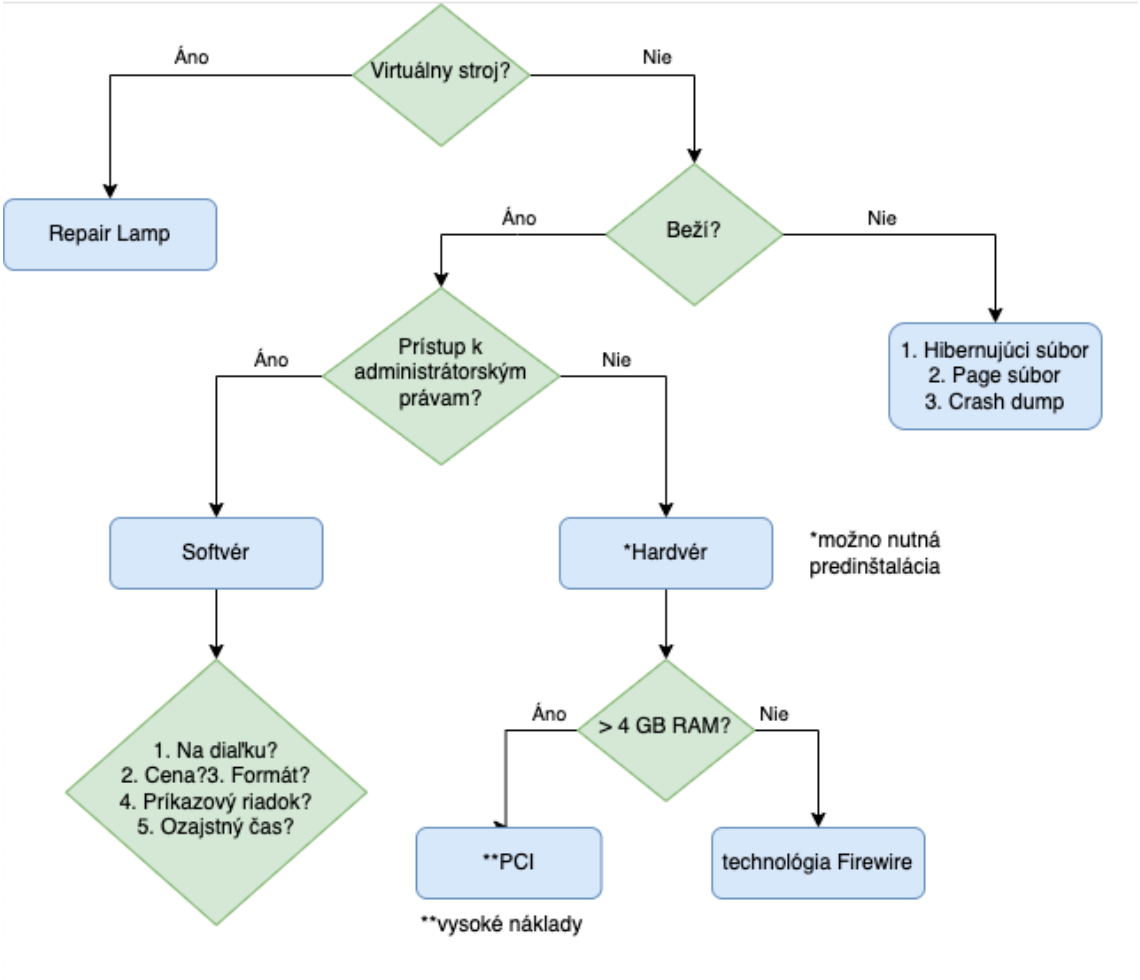
Zaisťovanie pamäte (i.e. capturing, dumping, sampling) zahŕňa kopírovanie obsahu volatilnej pamäte do nevolatilnej pamäte. Je to pravdepodobne jeden z najdôležitejších a taktiež aj neistých krokov v procese forenznej analýzy pamäte, keďže je vysoké riziko poškodenia obrazov pamäte, zničenia stôp a obmedzenia možností analýzy pamäte, ak sa obrazy pamäte nestratia úplne, čo je tiež možnosťou pri nesprávnom zaisťovaní operačnej pamäte. V našej práci sa zameriavame na zaisťovanie pamäte operačného systému Windows z dôvodu využívania nástrojov zameraných na operačný systém Windows.

Pre analýzu operačnej pamäte a prístupu k dátam z operačnej pamäte, potrebujeme pamäť najprv zaistiť, a to vytvorením obrazu pamäte pre prístup k bežiacim a ukončeným procesom, otvoreným portom, sieťovým spojeniam, súborom mapovaným v pamäti, skrytým údajom a ďalším dátam, ktoré sú uložené v operačnej pamäti v čase jej zaistenia.

Základom zaisťovania pamäte je postup kopírovania obsahu fyzickej pamäte do iného pamäťového zariadenia pre účely uchovania neporušenosti dát, získanie stôp bez zmeny ich stavu. Konkrétne metódy a nástroje, často závisia od cieľov vyšetrovania a vlastnostiach systému, ktorý je vyšetrovaný. Forenzný analytik sa snaží zachovať stav digitálneho prostredia spôsobom, ktorý forenznému analytikovi umožní dospieť analýzou k spoľahlivým záverom. Uložené údaje na disku a v pamäti RAM predstavujú dve najdôležitejšie zložky tohto prostredia.

Koreláciou údajov z viacerých zdrojov (disk, sieť, pamäť, atď.) v rámci digitálneho prostredia možno často lepšie pochopiť, čo sa stalo v systéme, než je obmedzená perspektíva, ktorú poskytuje len obsah diskového úložiska. Aby sme mohli zahrnúť tieto alternatívne zdroje, musíme akceptovať, že všetky získavané údaje, vrátane tradičných postupov získavania z disku, budú mať za následok určité skreslenie digitálneho prostredia. Analytici si musia byť vedomí toho, ako by tieto skreslenia mohli ovplyvniť

ich analýzu a v akom poradí musia zbierať údaje, aby tento vplyv znížili. Postup sa často uprednostňuje na základe poradia klesajúcej volatility (t. j., stopy, ktoré sa menia rýchlejšie, sa získavajú pred stopami, ktoré sú stabilnejšie). Najprv je potrebné získať nestále stopy z pamäte [7]. Hoci proces získavania vzorky fyzickej pamäte môže zvýšiť neistotu vo fáze zberu, dodatočné informácie, ktoré prináša, môžu viesť k väčšej dôvere v analýzu forenzného analytika a menšiemu skresleniu skutočných faktov vyšetrovania. Zaisťovanie operačnej pamäte je náročnejší proces, ktorý závisí od viacero faktorov, s ktorými v spojitosti so zaisťovaním operačnej pamäte pracujeme. Je potrebný všestranný súbor nástrojov a schopnosť prispôsobiť techniky na základe špecifík každého prípadu a prostredia, s ktorým sa analytik stretne. Obrázok 1 znázorňuje zjednodušený rozhodovací strom, ktorý zachytáva niektoré, avšak nie všetky podrobné faktory, s ktorými je potrebné vysporiadať sa pri zaisťovaní pamäte [8].



Obrázok 1 Schéma niektorých faktorov potrebných na zváženie pred zaistením pamäte [8]

1.2.1 Nástroje na zaisťovanie operačnej pamäte

Existuje niekoľko rôznych typov forenzného softvéru, ktoré môže forenzný analytik používať. Prvým z nich sú forenzné aplikácie. Tieto aplikácie sú účelovo určené na vykonávanie rôznych digitálnych forenzných úloh. Často sú komerčne dostupné a vo veľkej miere sa používajú v komunitách orgánov činných v trestnom konaní a vo vláde, ako aj v súkromnom priemysle [2].

Všetky softvérové nástroje na získavanie údajov postupujú pri získavaní pamäte podľa podobného postupu. Tieto nástroje pracujú najmä tak, že načítajú modul jadra, ktorý mapuje požadované fyzické adresy do virtuálneho adresného priestoru úlohy spustenej v systéme. V tomto okamihu môžu pristupovať k údajom z virtuálneho adresového priestoru a zapisovať ich do požadovanej nevolatilnej pamäte. Softvér na zaisťovanie využíva spôsob na mapovanie virtuálnych adries na fyzické, a to prístupom, ktorý využíva väčšina komerčne dostupných nástrojov.

V nasledujúcej časti sa pozrieme na niektoré nástroje s cieľom zaistenia operačnej pamäte. Na zozname sú uvedené bežne používané nástroje na zaisťovanie pamäte, a to bez určenia poradia. Cieľom nasledujúceho vymenovania nástrojov nie je poskytnúť zoznam funkcií, ktoré príslušné nástroje poskytujú, ale informatívne priblížiť nástroje, ktoré sú k dispozícii.

GMG Systems, Inc., KnTTools [9] - Medzi hlavné vlastnosti tohto nástroja patrí vzdialené nasadenie moduly, kryptografické kontroly integrity, zhromažďovanie stôp cez protokol SSL, kompresiu výstupu, voliteľné obmedzovanie šírky pásma, automatické zhromažďovanie údajov o stave používateľa v reálnom čase na krížové porovnávanie, zachytávanie súborov stránok, robustné zaznamenávanie chýb a dôkladné testovanie a dokumentácia. Môže tiež získavať ROM/EEPROM/NVRAM z BIOS-u a pamäte periférnych zariadení (PCI, grafická karta, sieťový adaptér).

F-Response [10] - Balík produktov od spoločnosti F-Response predstavil prelomovú novú schopnosť v oblasti forenznej analýzy pamäte - schopnosť skúmať živé systémy z vzdialeného miesta prostredníctvom pripojenia iSCSI určeného len na čítanie. F-Response poskytuje pohľad na fyzickú pamäť a pevné disky cieľového systému bez ohľadu na výrobcu a operačný systém, čo znamená, že k nim môžeme pristupovať z operačných systémov Windows, Mac OS X alebo z analýzy systému Linux a spracovať ich pomocou akéhokoľvek nástroja.

Mandiant Memoryze [11] - Nástroj, ktorý môžeme ľahko spustiť z vymeniteľného média, a ktorý podporuje získavanie z väčšiny populárnych verzií systému Microsoft Windows. Môžete importovať výstup XML z programu Memoryze do programu Mandiant Redline na grafickú analýzu objektov vo fyzickej pamäti.

HBGary FastDump [12] - Nástroj, ktorý tvrdí, že zanecháva čo najmenšiu schopnosť získať súbory stránok a fyzickú pamäť do jedného výstupného súboru (HPAK), a schopnosť skúmať pamäť procesu (potenciálne invazívna operácia, ktorá vyžaduje výmenu stránok späť do pamäte RAM pred získaním údajov).

MoonSols Windows Memory Toolkit [13] - MWMT obsahuje win32dd, win64dd a najnovšiu verziu nástroja DumpIt, ktorý kombinuje 32- a 64-bitové nástroje na získavanie obrazu pamäte do spustiteľného súboru, ktorý vyžaduje na ovládanie jediné kliknutie. Nie je potrebná žiadna ďalšia interakcia. Ak však potrebujete pokročilejšie možnosti, ako napríklad výber medzi typmi výstupných formátov, zapnutie šifrovania RC4 alebo skriptovanie vykonávania na viacerých počítačoch, môžeme to urobiť tiež.

AccessData FTK Imager [14] - Tento nástroj podporuje získavanie mnohých typov údajov, vrátane pamäte RAM. Spoločnosť AccessData predáva aj vopred nakonfigurovanú sadu nástrojov USB s priamou odozvou, ktorá okrem záznamov chatu, sieťových pripojení atď. získava aj fyzickú pamäť.

EnCase/WinEn [15] - Tento nástroj na získavanie údajov od spoločnosti Guidance Software dokáže vypisovať pamäť v komprimovanom formáte a zaznamenávať metadáta v hlavičkách (ako napr. názov prípadu, analytik atď.). Verzia EnCase Enterprise využíva podobný kód vo svojom agentovi ktorý umožňuje vzdialený prieskum živých systémov.

Belkasoft Live RAM Capturer [16] - Nástroj, ktorý inzeruje možnosť výpisu pamäte aj v prípade, že sú prítomné agresívne mechanizmy proti ladeniu a antidumpingu. Podporuje všetky hlavné 32- a 64-bitové verzie systému Windows a možno ho spustiť z USB flash disku.

ATC-NY Windows Memory Reader [17] - Tento nástroj dokáže uložiť pamäť v surovom stave alebo pri havárii a obsahuje rôzne možnosti hashovania integrity. Pri použití z prostredia podobného UNIXu, ako je MinGW alebo Cygwin, môžete ľahko odoslať výstup do vzdialeného poslucháča netcat alebo cez šifrovaný tunel SSH.

Winpmem [18] - Jediný open-source nástroj na získavanie pamäte pre systém Windows. Obsahuje možnosť výstupu súborov vo formáte raw alebo crash dump, výber medzi rôznymi metódami získavania (vrátane veľmi experimentálnej techniky PTE

remapping), a vystaviť fyzickú pamäť prostredníctvom zariadenia na živú analýzu miestneho systému.

Nástroj	Platformy	Cena
GMG Systems, Inc., KnTTools	Windows	zadarmo
F-Response	Windows, Linux, macOS	zadarmo
Mandiant Memoryze	Windows	platená
HBGary FastDump	Windows	platená
MoonSols Windows Memory Toolkit	Windows	zadarmo
AccessData FTK Imager	Windows	zadarmo
EnCase/WinEn	Windows	zadarmo
Belkasoft Live RAM Capturer	Windows	zadarmo
ATC-NY	Windows, Linux	zadarmo
Winpmem	Windows, Linux	zadarmo

Tabuľka 1 Porovnanie nástrojov na zaistenie operačnej pamäte

Z veľkého množstva voľne dostupných nástrojov na zaistenie operačnej pamäte sme využili softvér AccessData FTK Imager, z ktorého sme našu zaistenú pamäť uloženú ako mem súbor využili na analýzu a následné vytváranie výstupov vzťahov dát našej operačnej pamäte. Podrobnejšie sa na použité dáta pozrieme v tretej kapitole tejto práce.

1.3 Analýza operačnej pamäte

Forenzná analýza pamäte sa ukázala ako jeden z najvšestrannejších a najúčinnejších spôsobov analýzy počítačových systémov. Stala sa každodennou súčasťou postupov reakcie na incidenty, ako aj hnacou silou proaktívnej analýzy prostredí na prítomnosť škodlivých aktivít.

Za posledné desaťročie prispelo niekoľko faktorov k zvýšenému záujmu o techniky forenznej analýzy pamäte, ktoré umožňujú analyzovať operačnú pamäť systému na účely

forenzných artefaktov. Medzi tieto faktory patrí obrovský nárast veľkosti analyzovaných dát, väčšie spätné záznamy o prípadoch, keďže čoraz viac trestnej činnosti zahŕňa používanie počítačových systémov, využívanie forenzných techník pri reakcii na incidenty v boji proti škodlivému softvéru a trendy vo vývoji škodlivého softvéru, keď škodlivý softvér v súčasnosti bežne nezanecháva stopy na nie volatilných pamäťových zariadeniach, čo sú zariadenia, ktoré uchovávajú dáta aj po vypnutí zariadenia.

Dôležité je, že techniky forenznej analýzy pamäte môžu odhaliť značné množstvo nestálych stôp, ktoré by sa pri tradičných forenzných postupoch vypnutia zariadení úplne stratili. Tieto stopy zahŕňajú zoznamy spustených procesov, sieťové pripojenia, fragmenty nestálych údajov, ako sú správy v chate, a kľúčový materiál pre šifrovanie diskov [20].

1.3.1 Forenzné artefakty v operačnej pamäti

V tejto podkapitole sa bližšie pozrieme čo sú to forenzné artefakty, s akými sa stretneme v operačnej pamäti systému Windows a taktiež sa zameriame na tie, ktoré sú pre našu prácu relevantné, ktorým sa budeme bližšie venovať a budeme sledovať vzťahy medzi nimi. Pojem artefakt nemá v súčasnom ponímaní kybernetickej, respektíve digitálnej forenznej analýze formálnu definíciu. Termín bol vo všeobecnosti prijatý v téme kybernetickej forenznej oblasti pre objekty, ktoré napomáhajú pri napredovaní vyšetrovaní [23].

Analýza operačnej pamäte (RAM) je kľúčovou súčasťou forenznej analýzy aj systémov Windows. Poskytuje cenné informácie o prebiehajúcich procesoch, aktivitách užívateľov a stave systému v čase zaisťovania pamäte. Forenzný analytik môže z RAM extrahovať rôzne typy artefaktov, ktoré slúžia ako dôkazy v kriminálnych prípadoch alebo pri riešení bezpečnostných incidentov. Táto analýza umožňuje zachytiť volatilné artefakty, ktoré by boli po vypnutí systému stratené, a poskytuje informácie o prebiehajúcich aktivitách v čase zaisťovania pamäte. Taktiež môže pomôcť pri identifikácii malvéru a iných bezpečnostných hrozieb a poskytuje cenné informácie pre rekonštrukciu incidentu. Avšak, táto analýza vyžaduje špecializované nástroje a znalosti a môže byť náročná na čas a zdroje. Je dôležité si uvedomiť, že nie všetky artefakty sú vždy dostupné v RAM a volatilné artefakty sa po vypnutí systému stratia.

Medzi typické forenzné artefakty z RAM patrí zoznam bežiacich procesov vrátane názvu procesu, ID procesu (PID), rodičovského procesu a príkazového riadka, kde

v ukončených procesoch je možné zaznamenať aj čas spustenia procesu a čas ukončenia procesu. Informácie o vláknach v rámci procesov, vrátane ID vlákna, stavu vlákna a zásobníka vlákna. Záznamy o sieťovej aktivite, vrátane pripojení, sieťových paketov, DNS záznamov a webovej histórie. Kľúče a hodnoty z registra Windows, ktoré obsahujú konfiguráciu systému a nastavenia užívateľov. Stopy po malvére, ako sú súbory, procesy a sieťová aktivita spojená so škodlivým kódom. Rôzne artefakty z prehliadača ako história prehliadania, súbory cookie, uložené heslá a iné informácie z webových prehliadačov.

Okrem týchto typických artefaktov je možné z RAM extrahovať aj rôzne špecializované artefakty v závislosti od konkrétneho prípadu a cieľov analýzy. Forenzný analytik musí mať hlboké znalosti operačného systému Windows a forenzných techník, aby dokázal správne interpretovať a analyzovať extrahované artefakty [24].

1.4 Podobné práce

Tobias Latzo, Ralph Palutke, Felix Freiling skúmali využitie forenznej analýzy na získanie šifrovacích kľúčov a analýzu škodlivého softvéru, ktorý sa nachádza výlučne v pamäti RAM. Keďže pamäť sa zvyčajne získava pred samotnou analýzou, preskúmali rôzne techniky a nástroje na realizáciu tejto úlohy. V článku [19] definovali taxonómiu akvizičných metód založenú na dobre definovanom čiastkovom poradí, ktoré zovšeobecňuje koncept kruhového oddelenia oprávnení. Ich taxonómia nám umožnila poskytnúť komplexný prehľad najmodernejších techník zaisťovania pamäte, ktorý je nezávislý od použitého operačného systému a hardvérovej architektúry.

Vďaka výskumom autorov článku, sme dokázali vybrať a porovnať rôzne techniky na zaisťovanie pamäte.

Andrew Case a Golden G. Richard III. v článku [20] pojednávali o význame skúmania prchavej pamäte pre digitálnu forenznú analýzu. V minulosti sa digitálna forenzná analýza zameriavala na pamäťové zariadenia, no v súčasnosti je čoraz dôležitejšia forenzná analýza pamäte. Foren zná analýza pamäte umožňuje vyšetrovateľom nájsť dôkazy, ktoré by sa pri použití tradičných metód stratili.

Vývoj nástrojov na forenznú analýzu pamäte je náročný, ale ich prínos je značný a pomocou autorov sme dokázali hlbšie pochopiť dôležitosť zaisťovania a analýzy operačnej pamäte, pre nás špecificky pre operačný systém Windows.

Lucideus píše v článku [21] o získavaní prchavej pamäte systému Windows a forenznej analýze. Zameriava sa na to, čo je volatilná pamäť a prečo je dôležitá pre forenzné vyšetrovanie. Článok tiež podrobne opisuje kroky na získanie prchavej pamäte z počítača so systémom Windows. Niektoré z dôležitých bodov tohto článku sú, že prchavá pamäť sa stratí, keď sa zariadenie reštartuje alebo vypne, a že získavanie prchavej pamäte môže byť riskantné, ale môže tiež poskytnúť vyšetrovateľom dôležité informácie. Autor nás bližšie voviedol do témy forenznej analýzy, a to hlbším ozrejmením problematiky na zaisťovanie operačnej pamäte so systémom Windows, ktorému sme sa v našej práci venovali.

Hannah Nyhlom a spol. sa v článku [6] zaoberajú zaisťovaním operačnej pamäte na forenznú analýzu. Preskúmavajú rôzne techniky zaisťovania a kategorizujú ich na základe úrovne hierarchie prístupu. Tiež diskutujú o kladoch a záporoch každej techniky. Prínosom tohto článku bolo najmä argumentovanie výhod a nevýhod jednotlivých techník pre zaistenie operačnej pamäte.

Khaleque Md Aashiq Kamal a spol. sa zameriavajú na porovnávanie rôznych nástrojov na forenznú analýzu operačnej pamäte. V článku [22] sú analyzované nástroje na základe času spracovania a množstva zanechaných artefaktov v pamäti. Testované nástroje zahŕňajú FTK Imager, Pro Discover, Nigilant32, Helix3(dd), OSForensics a Belkasoft RAM Capturer. Štúdia tiež zistila, že zväčšenie veľkosti pamäte neznamená proporcionálne zvýšenie času spracovania nástrojov.

V rôznorodosti prínosu tém týkajúcich sa forenznej analýze operačnej pamäte a jej zaisťovania, prispeli svojimi poznatkami aj autori tohto článku, ktorí priblížili iné nástroje na zaistenie, ale aj analýzu operačnej pamäte.

2 Nástroje na analýzu operačnej pamäte

Ako sme v prechádzajúcej kapitole spomínali, po zaistení operačnej pamäte môžeme následne pokračovať na jej analýzu. Analýzou zaistenej operačnej pamäte vieme na jednej strane získať artefakty z pamäte, ktoré sa v terminológii bezpečnosti označujú aj ako forenzné artefakty, zároveň ich vieme vyhodnocovať, analyzovať a hľadať vzťahy medzi artefaktmi, ku ktorým pristúpime rôznymi spôsobmi.

Bolo vytvorených niekoľko nástrojov na analýzu pamäte, ktoré umožňujú používateľovi analyzovať výpisy pamäte a hľadať užitočné artefakty. Medzi príklady takýchto nástrojov s otvoreným zdrojovým kódom patria Volatility a Rekall a medzi komerčné nástroje patria Cellebrite Inspector, FireEye Redline, Magnet AXIOM a WindowsSCOPE. Takmer všetky výskumné metódy využívajú softvér Volatility a väčšina komerčných riešení využíva Volatility v rámci svojho produktu. V nasledujúcich kapitolách sa bližšie povenujeme niektorým z týchto nástrojov [6].

2.1 Volatility 2

Volatility 2 je open-source framework naporgramovaný v jazyku Python 2 pre forenznú analýzu pamäte, ktorý umožňuje skúmať obsah obrazov pamäte, dumpov, z rôznych operačných systémov, primárne Windows. Poskytuje širokú škálu pluginov na analýzu rôznych aspektov pamäte, ako sú procesy, vlákna, ovladače zariadení, sieťové aktivity a registre [25].

Nástroj funguje na princípe analýzy pamäťových snímok, ktoré sú statickými kópiami obsahu pamäte v danom okamihu. Volatility 2 sa používa v rôznych oblastiach forenznej informatiky, napríklad na vyšetrovanie kybernetických incidentov, digitálne vyšetrovanie a podobne. Medzi jeho hlavné výhody patrí široká škála funkcií, open-source dostupnosť a aktívna komunita vývojárov a používateľov. Nevýhodou je náročnosť používania a obmedzená dostupnosť profilov pre niektoré verzie Windows.

Volatility 2 je výkonný nástroj pre forenznú analýzu pamäte, ktorý umožňuje skúmať rôzne aspekty pamäťových snímok z rôznych operačných systémov. Je však dôležité poznamenať, že používanie Volatility 2 môže byť náročné a vyžaduje si pochopenie štruktúry pamäte operačného systému.

Pre prístup k dátam z pamäti, je potrebné zadať profil pamäte, ktorý získame pluginom imageinfo, ktorý ako jeden z mála funguje bez zadania pamäte [6].

python vol.py -f <*výpis súboru*> imageinfo

Po získaní profilu sa pridáva k vyvolávaciemu príkazu aj názov profilu pamäte.

python vol.py -f <*výpis súboru*> --profile= <*profil pamäte*> pslist

2.2 Volatility 3

Volatility 3 rámec na analýzu operačnej pamäte bežiaci v Pythone 3 dokáže analyzovať výpisy pamäte z počítačov so systémami Windows, Linux alebo Macintosh, podporuje mnoho rôznych typov formátov výpisov súborov [26]. Má vynikajúce funkcie na vytváranie funkcií a je pomerne efektívne implementovaný. Volatility sa stal najväčším a najlepšie podporovaným frameworkom vďaka veľkej základni prispievateľov a nezávisle vytvorených zásuvných modulov, z ktorých mnohé sú zamerané na forenzné analýzy špecifických platforiem. Hlavnou nevýhodou Volatility je, že sa zvyčajne spúšťa z rozhrania príkazového riadka, čo ju robí pre niektorých potenciálnych používateľov nedostupnou. Hoci Volatility obsahuje mnoho rôznych režimov a príkazov, typické vyvolanie má nasledujúcu podobu:

python3 vol.py -f <*cesta k súboru*> windows.pslist

V tomto príklade Volatility extrahuje zoznam procesov systému Windows v čase výpisu obrazu pamäte. Vývojári Volatility tvrdia, že je efektívnejší ako iný forenzný softvér na analýzu pamäte vrátane Rekall. Okrem toho počas verejného vydania beta verzie Volatility3 vývojári tvrdili, že v mnohých prípadoch došlo k výraznému zlepšeniu výkonu oproti Volatility2 a Rekall.

Pomocou nástroja Volatility je možné získať aktuálne a predchádzajúce handle procesov bežiacich v systéme, načítané knižnice DLL (dynamic-link libraries) procesu, všetky príkazy, ktoré útočník zadal prostredníctvom konzolového shellu, rezidentné stránky v pamäti a spustiteľné súbory procesu. Napokon, Volatility dokáže získať informácie o pripojeniach, ktoré systém vytvoril. Konkrétne získava spojenia TCP, ktoré boli aktívne v čase získania pamäte, počúvajúce zásuvky pre akýkoľvek protokol, zvyškové údaje a artefakty z predchádzajúcich zásuviek a sieťové artefakty vrátane koncových bodov TCP, poslucháčov TCP, koncových bodov UDP a poslucháčov UDP [6].

2.3 Rekall

Rekall je platformou podobnou Volatility, vyvinutou spoločnosťou Google, ktorá tvrdí, že je to najkomplexnejší rámec na analýzu pamäte. Tento softvér je k dispozícii pre platformy Linux, macOS a Windows. Spoločnosť Google vydala aj nástroj na získavanie pamäte Pmem, navrhnutý tak, aby spolupracoval s rámcom Rekall a poskytoval jediný bod pre súbor nástrojov na získavanie a analýzu [27].

Rekall je otvorený softvér, ktorý je určený na extrakciu, analýzu a vizualizáciu dát z pamäte počítačových systémov. Pomocou Rekallu môžu forenzní analytici extrahovať široké spektrum informácií z operačnej pamäte, vrátane spustených procesov, otvorených súborov, prihlásených používateľov a dokonca aj pamäťových štruktúr jadra operačného systému. Tieto údaje môžu byť následne analyzované s cieľom identifikovať nezvyčajné aktivity, ako sú malware, zneužívanie oprávnení alebo neoprávnený prístup. Využíva v rôznych oblastiach, vrátane forenzných analýz operačnej pamäte, kde poskytuje komplexné prostredie na preskúmanie dát a nájdenie dôležitých stop. Medzi typické prípady použitia patrí vyšetrovanie kybernetických útokov, digitálne forenzné vyšetrovania a incidentné reakcie. Forenzní analytici môžu využívať jeho množstvo modulov a nástrojov na rôzne účely, ako je napríklad vyhľadávanie špecifických vzorov, analýza škodlivého softvéru alebo rekonštrukcia udalostí.

2.4 Porovnanie nástrojov

Pri výbere najvhodnejšieho nástroja pre analýzu operačných pamätí zavážilo množstvo faktorov. Najdôležitejší dopad bola najmä na aktuálnosť technológie. Tabuľka 2 zobrazuje porovnanie technológií na analýzu pamäte.

	Rekall	Volatility 2	Volatility 3
Vývoj	Google Security komunita	Tím Volatility	Tím Volatility
Udržiavanosť	Udržiavaný komunitou	Neudržiavaný	Aktívne udržiavaný
Verzia Pythonu	Python 3	Python 2	Python 3

Podporované verzie OS	Windows 10, Windows 11	Windows 7, Windows 10	Windows 10, Windows 11
Profil	Automatické zistenie profilu OS	Manuálne zadanie profilu OS	Automatické zistenie profilu OS
Update	December 2017	December 2015	Január 2024

Tabuľka 2 Porovnanie nástrojov na analyzovanie operačnej pamäte

Signifikantným rozdielom medzi technológiou Rekall a Volatility 3 je najmä aktuálnosť, kde je Volatility 3 aktuálne vyvíjaná a aktualizovaný tímom vývojárov a nie komunitou ako v prípade Rekall. Volatility 2 nie je vôbec aktualizovaný. Obidve Volatility technológie sú vyvinuté vývojovým tímom Volatility, zatiaľ čo Rekall je od spoločnosti Google. Aktuálne je Volatility 3 ako jediný pravidelne udržiavaný tímom vývojárov technológie, kým Rekall je udržiavaný iba otvorenou komunitou a Volatility 2 nie je vôbec udržiavaný. Rekall a Volatility 3 podporujú aktuálnu verziu Pythonu a Volatility 2 iba verziu nižšiu ako 3. Pri operačných systémoch, ktoré podporujú jednotlivé tehcnológie sme sa zamerali na Windows, kde najnovšiu verziu 11 podporuje Rekall a Volatility 3. Volatility 2 podporuje najvyššiu verziu Windows 10, niektoré pluginy podporovali iba Windows 7 a staršie. Pri spúšťaní pluginov Volatility 2 je potrebné manuálne vyhľadať a zadať profil operačného systému, kde pri Rekall a Volatility 3 je tento profil automaticky zistení pri spúšťaní príkazu. Najnovší update bol vykonaný pre technológiu Volatility 3, a to v januári tohto roku, Rekall bol aktualizovaný na konci roka 2017 a Volatility 2 bol aktualizovaný naposledy decembri 2015.

3 Návrh nástroja

V tejto kapitole našej práce sa pozrieme na nami navrhnutý nástroj, ktorý vie pomôcť forenzným analytikom zjednodušiť zobrazenie dát použitím pluginov. Zápisom, čítaním, parsovaním a vykresľovaním dát vieme získať jednoduchší prístup k dátam v CSV súboroch a unikátne vzťahy medzi dátami v operačnej pamäti pre hlbšiu analýzu artefaktov a stôp.

3.1 Popis použitých dát

V našej práci sme sa zamerali na dáta, ktoré získame fyzickým zaistením operačnej pamäte, kde sme proces opisovali v prvej kapitole tejto práce. Rôznorodosť dát sme zabezpečili prácou s viacerými zaistenými operačnými pamäťami. Presnejšie sme pracovali s našou samostatne zaistenou pamäťou, a to vytvorením mem súboru pomocou technológie AccessData FTK Imager na zaisťovanie pamätí, ktorý sme bližšie opisovali v prvej kapitole tejto práce. Zaistená pamäť je o veľkosti 9 GB a s najnovším operačným systémom Windows 11, kde sa jednalo o spustené bežné používateľské procesy s internetom, stiahnutými aplikáciami, powershellom, nastavení počítača a pod. Následne sme v práci využili dáta z voľne dostupných zaistených operačných pamätí z ponuky Memlabs, čo je súbor výziev v štýle Capture The Flag (CTF) pre študentov a výskumníkov. Presnejšie sme využili MemoryDump_Lab1.raw a MemoryDump_Lab2.raw súbory, ktoré sa využívajú na tréning, resp. na vyskúšanie si analýzy operačnej pamäte spolu s dokumentovaným postupom pri hľadaní podozrivých artefaktov v pamäti. Hlavným cieľom vytvorenia tohto úložiska bolo poskytnúť spoľahlivú platformu, kde sa jednotlivci môžu učiť, precvičovať a zlepšovať si svoje zručnosti v oblasti forenznej analýzy pamäte [28]. Tretím zdrojom dát bola zaistená pamäť so známym prípadom ukradnutého receptu sečuánskej omáčky, ktorý je voľne dostupný pre tréning analýzy pamäte so zaujímavými artefaktmi, úlohami na hľadanie konkrétnych forenzných artefaktov v pamäti, prípadne priame postupy ako analyzovať [29].

3.2 Výber pluginov

V tejto kapitole sa budeme venovať vybraným nástrojom a postupom, ktoré sme v našej práci zvolili. V prvom rade je potrebné podotknúť, že technológiu na analýzu

operačnej pamäte sme zvolili Volatility, ktorej sme viac venovali v 2. kapitole tejto práce. Kvôli nedostupnosti všetkých pluginov v aktuálnej verzii Volatility 3, sme využili aj Volatility 2, čo znamená, že bola potrebná aj práca s programovacím jazykom Python verzie nižšej ako tri a využili sme verziu 2.7 pre vôbec umožnenie rozbehnutia Volatility 2. Prostredím na získavanie dát nám najprv postačoval príkazový riadok, avšak pri neskorším náročnejších krokoch, sme zvolili programovacie prostredie prístupné pre jazyk Python, a to Jupyter Notebook [30]. Pre uľahčenie vytvárania výstupov v podobe CSV súborov sme zvolili dátovú štruktúru z knižnice pandas implementovateľnú v jazyku Python, kde sme si jednotlivé dáta vedeli rozdeliť do stĺpcov a riadkov a následne generovali výsledný CSV súbor, kde oddeľovač je čiarka.

Z dôvodu vizualizácie dát sme zvolili grafovú štruktúru orientovaných grafov pre zobrazenie vzťahov jednotlivých dát využitím balíka NetworkX implementovanom pre jazyk Python.

Postupnosť našej práce spočívala v získavaní dát technológiou Volatility, parsovaním dát pomocou jazyka Python a využitím dátovej štruktúry DataFrame sme sa dopracovali k prvým výsledkom, čo boli CSV súbory s dátami jednotlivých pluginov. Ďalej sme dáta obsahujúce proces ID upravili pre vizualizáciu, keďže analýzou výstupov pluginov sme dospeli k záveru, že je možné spájať iba informácie z pluginov iba na základe proces ID, ktoré je nestále a obsahuje ho viacero pluginov. Potrebovali sme dáta spojiť na základe proces ID, kde sme predchádzali duplicite jednotlivých dát. Po získaní relevantných dát pre vizualizáciu a spojenie ich do jedného CSV súboru, sme vytvorili orientovaný graf vzťahov medzi týmito dátami.

Nami navrhnutý nástroj je implementovaný v programovacom jazyku Python verzie 3 a využíva viacero potrebných knižníc, presnejšie pandas, CSV, Matplotlib a NetworkX. Pre využitie nástroja, sme použili dáta získané z troch rôznych zaistených pamätí, ktoré sú presnejšie popísané v podkapitole 3.1. Pre jednoduchosť spracovania dát, sme zvolili možnosť zápisu dát technológiou Volatility 3 do textového súboru spustením príkazu v príkazovom riadku nášho zariadenia:

!python3 vol.py -f <*cesta k obrazu pamäte*> **windows.**<*názov pluginu*> **>** <*cesta k súboru*>**.txt**

Následne nami implementovaným nástrojom v programovacom jazyku Python verzie 3 sme pristúpili k dátam zapísaným do textového súboru a využitím technológie

DataFrame z knižnice pandas, sme dokázali pristúpiť k dátam jednotlivo a vygenerovať súbor čitateľnejší pre analytikov, a to súbor CSV.

V prípade práce s Volatility 2 sme využili programovací jazyk Python 2.7, keďže táto technológia podporuje jedine Python do verzie 3. Pri zápise do súborov, sme potrebovali najprv zistiť profil pamäte, vďaka ktorému sme vedeli pristúpiť k presnejším dátam v operačnej pamäti.

!python vol.py -f <*cesta k obrazu pamäte*> **imageinfo**

Vo výpise sme získali Navrhované profily, ktoré sú podobné nášmu obrazu. Vďaka zadaniu profilu k pluginom sme mohli využiť ďalší prístup k dátam a ich následnému zápisu do textového súboru.

!python vol.py –profile==<získaný profil> **-f** <*cesta k obrazu pamäte*> <*názov pluginu*> **>** <*cesta k súboru*>**.txt**

V nasledujúcej časti sa pozrieme bližšie na jednotlivé pluginy technológií Volatility 2 a Volatility 3, ktoré sme v našej práci využili. Dôležité je podotknúť, že aj napriek neaktualizovaniu Volatility 2 sme s ňou pracovali z dôvodu niektorých pluginov, ktoré vo Volatility 3 neboli k dispozícii, z tohto dôvodu sme nedokázali využiť iba pluginy dostupné vo Volatility 3. Na začiatku sa zameriame na pluginy, ktoré sú dostupné v obidvoch verziách Volatility, kde rozdiel výstupov je zanedbateľný, a to napríklad iné usporiadanie stĺpcov, iné názvy označení častí dát a pod.

pslist – Plugin, ktorý zobrazuje zoznam bežiacich, alebo ukončených procesov. Poskytuje detaily o samotných procesoch ako je názov procesu, ID procesu (PID), ID nadradeného procesu (PPID), offset, počet vlákien a handle, časovú pečiatku začiatku a konca procesu. Pomáha analyzovať procesy a ich vzťahy v systéme. Pre získanie informácií o procesoch máme k dispozícii v obidvoch verziách technológií aj pluginy psscan a pstree, kde prvý menovaný môže získať všetky procesy v pamäti, vrátane aktívnych, ukončených a skrytých procesov a druhý v poradí usporiada procesy do stromovej štruktúry systémom nadradených a podradených procesov, deteguje to totožnou metódou ako pslist, preto sa neberú do úvahy skryté alebo neviazané procesy. Obrázok 2 je zobrazením výstupu niektorých dát z obrazu operačnej pamäte MemoryDump_Lab1.raw spusteného v prostredí Jupyter Notebook.

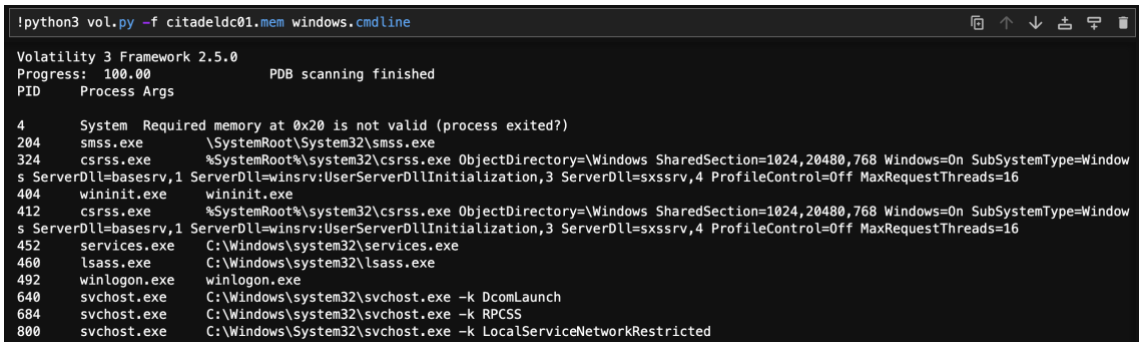
```
!python3 vol.py -f MemoryDump_Lab1.raw windows.pslist

Volatility 3 Framework 2.5.0
Progress:  100.00               PDB scanning finished
PID     PPID    ImageFileName   Offset(V)       Threads Handles SessionId       Wow64   CreateTime      ExitTime        File output

4       0       System  0xfa8000ca0040  80      570     N/A     False   2019-12-11 13:41:25.000000      N/A     Disabled
248     4       smss.exe        0xfa800148f040  3       37      N/A     False   2019-12-11 13:41:25.000000      N/A     Disabled
320     312     csrss.exe       0xfa800154f740  9       457     0       False   2019-12-11 13:41:32.000000      N/A     Disabled
368     360     csrss.exe       0xfa8000ca81e0  7       199     1       False   2019-12-11 13:41:33.000000      N/A     Disabled
376     248     psxss.exe       0xfa8001c45060  18      786     0       False   2019-12-11 13:41:33.000000      N/A     Disabled
416     360     winlogon.exe    0xfa8001c5f060  4       118     1       False   2019-12-11 13:41:34.000000      N/A     Disabled
424     312     wininit.exe     0xfa8001c5f630  3       75      0       False   2019-12-11 13:41:34.000000      N/A     Disabled
484     424     services.exe    0xfa8001c98530  13      219     0       False   2019-12-11 13:41:35.000000      N/A     Disabled
492     424     lsass.exe       0xfa8001ca0580  9       764     0       False   2019-12-11 13:41:35.000000      N/A     Disabled
500     424     lsm.exe 0xfa8001ca4b30  11      185     0       False   2019-12-11 13:41:35.000000      N/A     Disabled
588     484     svchost.exe     0xfa8001cf4b30  11      358     0       False   2019-12-11 13:41:39.000000      N/A     Disabled
652     484     VBoxService.ex  0xfa8001d327c0  13      137     0       False   2019-12-11 13:41:40.000000      N/A     Disabled
720     484     svchost.exe     0xfa8001d49b30  8       279     0       False   2019-12-11 13:41:41.000000      N/A     Disabled
816     484     svchost.exe     0xfa8001d8c420  23      569     0       False   2019-12-11 13:41:42.000000      N/A     Disabled
852     484     svchost.exe     0xfa8001da5b30  28      542     0       False   2019-12-11 13:41:43.000000      N/A     Disabled
876     484     svchost.exe     0xfa8001da96c0  32      941     0       False   2019-12-11 13:41:43.000000      N/A     Disabled
```

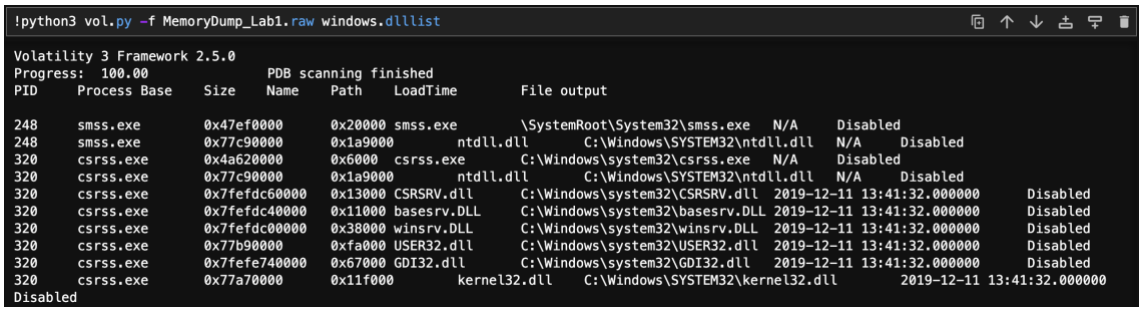
Obrázok 2 Ukážka výpisu pluginu pslist

cmdline – Plugin, ktorý slúži na zobrazenie príkazových riadkov procesov, ktoré bežali v systéme. Zobrazuje informácie ako ID procesu (PID), názov procesu, cestu k súboru s programom a argumenty príkazového riadka. Obrázok 3 zobrazuje výstup niektorých dát z dumpu operačnej pamäte citadelc01.mem, čo je pamäť zariadenia v spojitosti s ukradnutou sečuánskou omáčkou, spusteného v prostredí Jupyter Notebook.



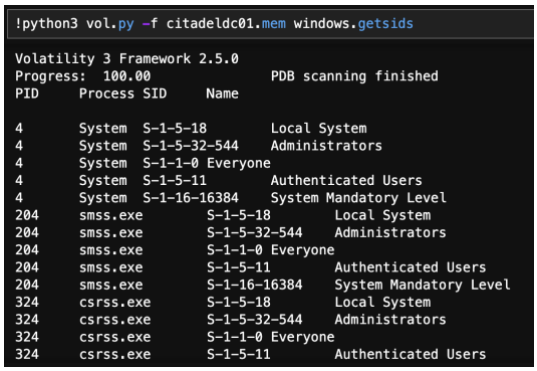
Obrázok 3 Ukážka výpisu pluginu cmdline

dlllist – Plugin, ktorý zobrazuje zoznam dynamických knižníc (DLL) načítaných do procesu. Zobrazuje informácie ako meno DLL, cestu k súboru s DLL, veľkosť DLL, dátum a čas načítania DLL a priečinok s výstupom. Plugin je užitočný pri forenznej analýze, pretože umožňuje identifikovať DLL, ktoré boli načítané do procesu, a zistiť, či s nimi bolo manipulované. Obrázok 4 zobrazuje výstup niektorých dát z obrazu operačnej pamäte MemoryDump_Lab1.raw spusteného v prostredí jupyter notebook.



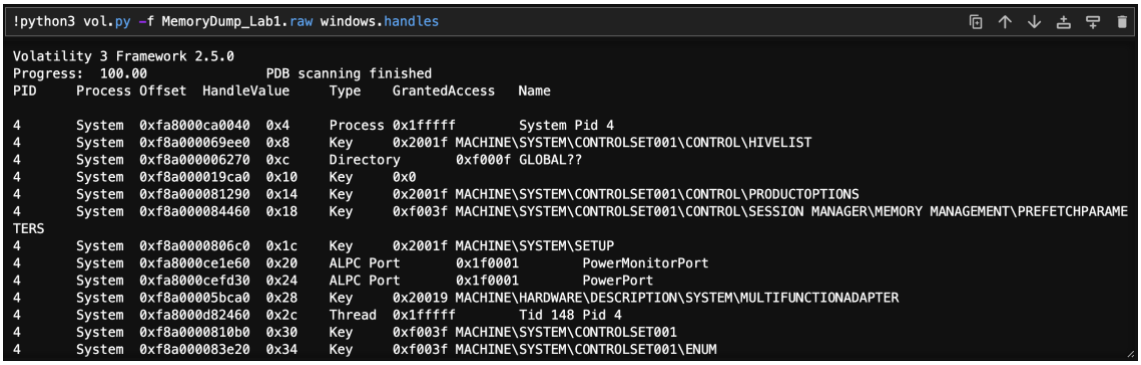
Obrázok 4 Ukážka výpisu pluginu dlllist

getsids - getsids je ďalší užitočný plugin pre Volatility, ktorý získava bezpečnostné identifikátory (SIDs) pre procesy. Poskytuje informácie o bezpečnostných identifikátoroch procesov, čo je dôležité pri analýze oprávnení a prístupových práv. Pomáha pri identifikácii procesov a ich prístupových práv v systéme, čo je dôležité pri forenzných analýzach a bezpečnostných auditoch. Vo výstupe zobrazuje štyri stĺpce, a to ID procesu, názov procesu, bezpečnostný identifikátor používateľa (security identificator – SID) a meno používateľa. Obrázok 5 zobrazuje výstup niektorých dát z obrazu operačnej pamäte citadelc01.mem spusteného v prostredí jupyter notebook.



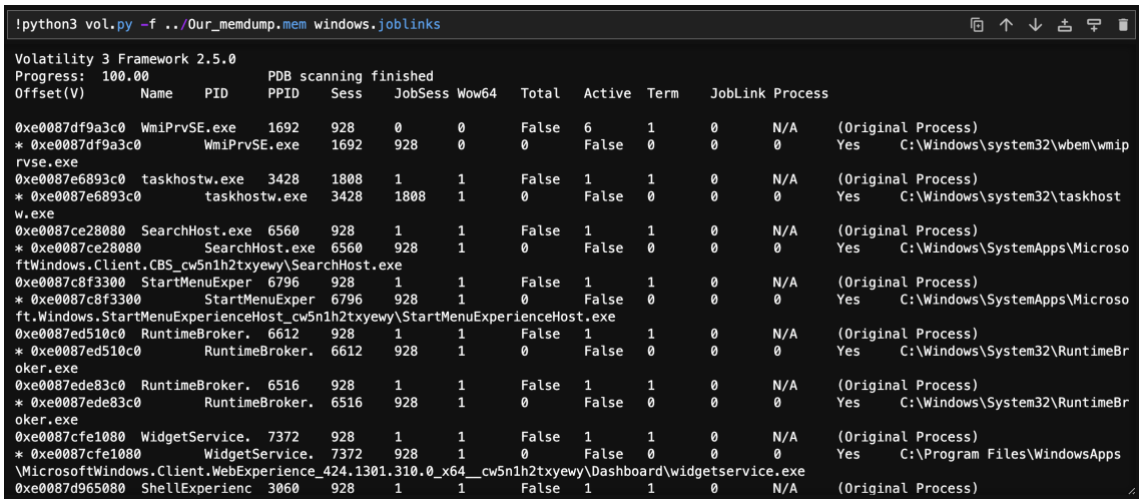
Obrázok 5 Ukážka výpisu pluginu getsids

handles - Plugin zobrazuje zoznam procesom otvorených súborov, adresárov a registrov. Zobrazuje informácie ako typ objektu (súbor, adresár, register), cestu k objektu, stav objektu (otvorený, čítaný, zapisovaný). Plugin je užitočný pri forenznej analýze, pretože umožňuje zistiť, aké súbory, adresáre a registre boli procesom otvorené. Obrázok 6 zobrazuje výstup niektorých dát z obrazu operačnej pamäte MemoryDump_Lab1.raw spusteného v prostredí jupyter notebook.



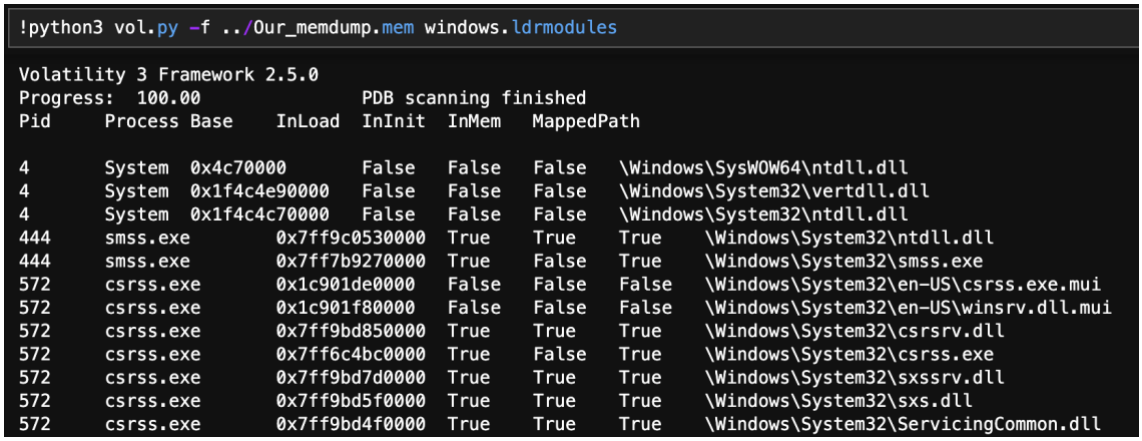
Obrázok 6 Ukážka výpisu pluginu handles

joblinks – Plugin, ktorý zobrazuje zoznam úloh (jobov) a procesov, ktoré k nim patria. Zobrazuje informácie ako offset, ID jobu, ID procesu, ID nadradeného procesu, stav jobu (beží, pozastavený, ukončený). Plugin je užitočný pri forenznej analýze, pretože umožňuje zistiť, aké procesy patria k danému jobu. Obrázok 7 zobrazuje výstup niektorých dát z obrazu operačnej pamäte Our_memdump.mem, čo je nami vytvorený dump pamäte z nášho zariadenia, spusteného v prostredí jupyter notebook.



Obrázok 7 Ukážka výpisu pluginu joblinks

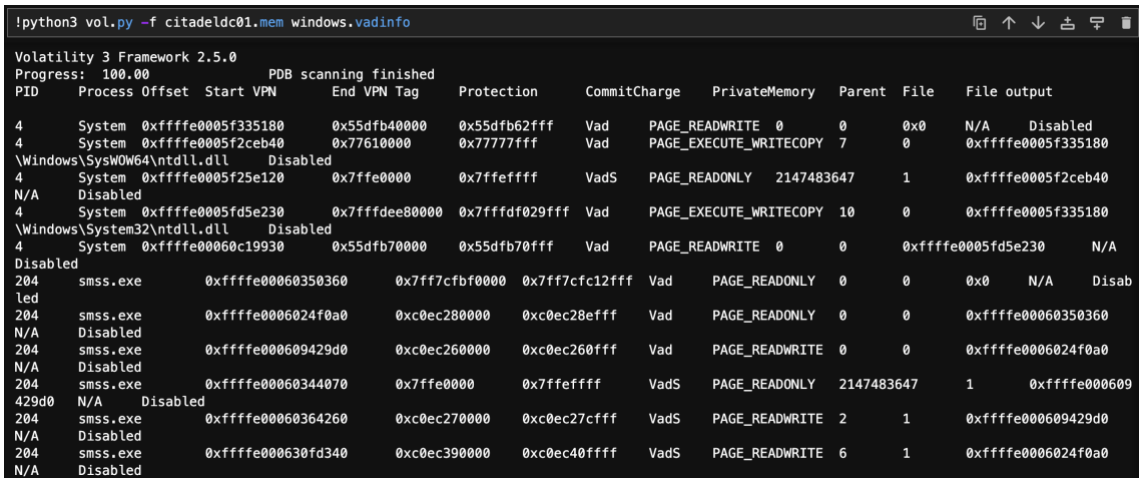
ldrmodules - Ldrmodules je ďalší plugin pre Volatility, ktorý identifikuje načítané moduly v procesoch. Poskytuje informácie o načítaných knižniciach, DLL súboroch a ich umiestnení v pamäti procesu. Umožňuje analyzovať, ktoré knižnice sú načítané v konkrétnych procesoch a pomáha pri identifikácii potenciálnych hrozieb. Obrázok 8 zobrazuje výstup niektorých dát z obrazu operačnej pamäte Our_memdump.mem, čo je nami vytvorený dump pamäte z nášho zariadenia, spusteného v prostredí jupyter notebook.



Obrázok 8 Ukážka výpisu pluginu ldrmodules

vadinfo - Plugin vadinfo zobrazuje informácie o virtuálnej pamäti procesu. Zobrazuje informácie ako typ pamäte (vyhradená, mapovaná), veľkosť pamäte, stav

pamäte (voľná, použitá). Tento plugin je užitočný pri forenznej analýze, pretože umožňuje zistiť, ako je využívaná virtuálna pamäť procesu. Obrázok 9 zobrazuje výstup niektorých dát z obrazu operačnej pamäte Our_memdump.mem, čo je nami vytvorený dump pamäte z nášho zariadenia, spusteného v prostredí jupyter notebook.



```
!python3 vol.py -f citadeldc01.mem windows.vadinfo

Volatility 3 Framework 2.5.0
Progress:  100.00               PDB scanning finished
PID     Process Offset  Start VPN       End VPN Tag     Protection      CommitCharge    PrivateMemory   Parent  File    File output

4       System  0xffffe0005f335180      0x55dfb40000    0x55dfb62fff    Vad     PAGE_READWRITE  0       0       0x0     N/A     Disabled
4       System  0xffffe0005f2ceb40      0x77610000      0x77777fff      Vad     PAGE_EXECUTE_WRITECOPY  7       0       0xffffe0005f335180
\Windows\SysWOW64\ntdll.dll     Disabled
4       System  0xffffe0005f25e120      0x7ffe0000      0x7fefffff      VadS    PAGE_READONLY   2147483647      1       0xffffe0005f2ceb40
N/A     Disabled
4       System  0xffffe0005fd5e230      0x7fffdee80000  0x7fffdf029fff  Vad     PAGE_EXECUTE_WRITECOPY  10      0       0xffffe0005f335180
\Windows\System32\ntdll.dll     Disabled
4       System  0xffffe00060c19930      0x55dfb70000    0x55dfb70fff    Vad     PAGE_READWRITE  0       0       0xffffe0005fd5e230      N/A
Disabled
204     smss.exe        0xffffe00060350360      0x7ff7cfbf0000  0x7ff7cfc12fff  Vad     PAGE_READONLY   0       0       0x0     N/A     Disab
led
204     smss.exe        0xffffe0006024f0a0      0xc0ec280000    0xc0ec28efff    Vad     PAGE_READONLY   0       0       0xffffe00060350360
N/A     Disabled
204     smss.exe        0xffffe000609429d0      0xc0ec260000    0xc0ec260fff    Vad     PAGE_READWRITE  0       0       0xffffe0006024f0a0
N/A     Disabled
204     smss.exe        0xffffe00060344070      0x7ffe0000      0x7fefffff      VadS    PAGE_READONLY   2147483647      1       0xffffe000609
429d0   N/A     Disabled
204     smss.exe        0xffffe00060364260      0xc0ec270000    0xc0ec27cfff    VadS    PAGE_READWRITE  2       1       0xffffe000609429d0
N/A     Disabled
204     smss.exe        0xffffe000630fd340      0xc0ec390000    0xc0ec40ffff    VadS    PAGE_READWRITE  6       1       0xffffe0006024f0a0
N/A     Disabled
```

Obrázok 9 Ukážka výpisu pluginu vadinfo

V nasledujúcej časti sa pozrieme na niektoré pluginy technológie Volatility 2, ktoré sme využili v našej práci a nie sú dostupné v novšej verzii Volatility 3, avšak sú pre nás zaujímavé. Presnejšie sa dotkneme pluginov, ktoré sú zamerané na súbory, sieťové spojenia a vlákna.

filescan - Plugin filescan slúži na vyhľadávanie používaných a historických súborov v pamäti. Výpis obsahuje offset, počet referencií na daný súbor, počet handles, prístup (čítanie, zapisovanie, spúšťanie), cestu k súboru. Plugin je užitočný pri forenznej analýze, pretože umožňuje vyhľadať súbory, ktoré boli v systéme otvorené, vytvorené alebo modifikované. Obrázok 10 je zobrazuje výstup niektorých dát z obrazu operačnej pamäte MemoryDump_Lab1.raw spusteného v prostredí jupyter notebook.

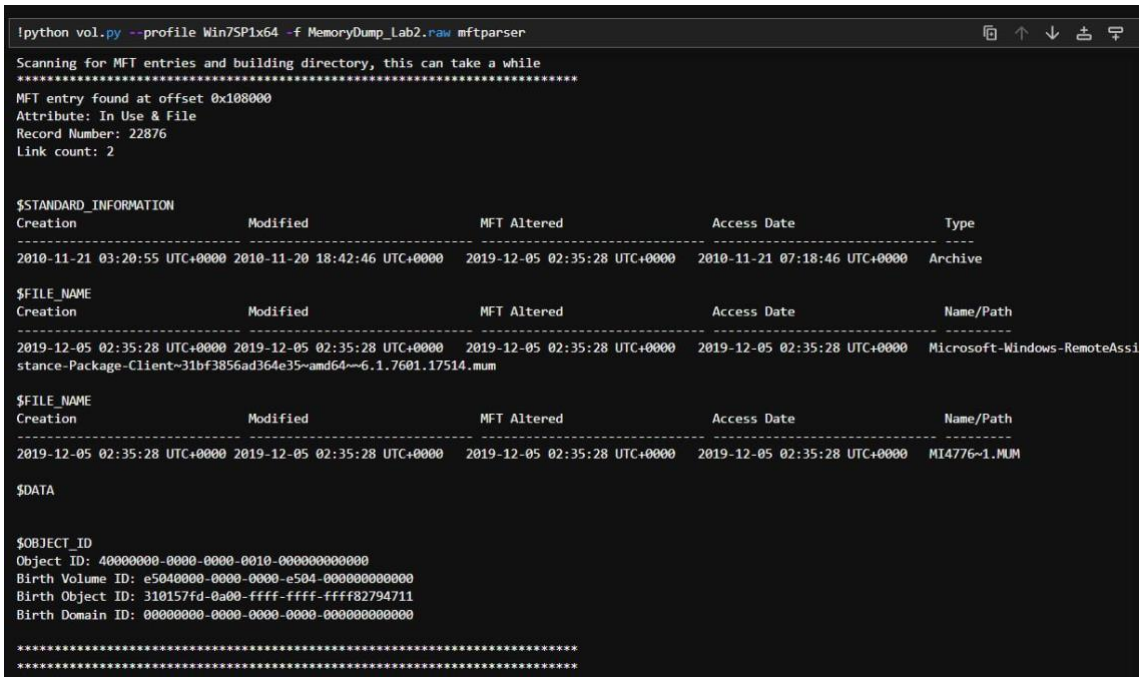
```
!python vol.py --profile Win7SP1x64 -f MemoryDump_Lab1.raw filescan |more +23

Offset(P)           #Ptr   #Hnd Access Name
------------------ ------ ------ ------ ----
0x000000003e801310      2      1 ------ \Device\NamedPipe\MsFteWds
0x000000003e809610      9      0 R--r-d \Device\HarddiskVolume2\Windows\System32\dot3api.dll
0x000000003e80b9f0      2      1 ------ \Device\Afd\Endpoint
0x000000003e80bf20      2      1 ------ \Device\Afd\Endpoint
0x000000003e80c070      9      0 R--r-d \Device\HarddiskVolume2\Windows\System32\eappcfg.dll
0x000000003e80fb00     15      0 R--r-d \Device\HarddiskVolume2\Windows\ehome\ehepgres.dll
0x000000003e8105e0      1      0 R--rwd \Device\HarddiskVolume2\Users\Alissa Simpson\Links\desktop.ini
0x000000003e811220      1      0 R--rwd \Device\HarddiskVolume2\Users\Alissa Simpson\Favorites\desktop.ini
0x000000003e811370      1      0 R--rwd \Device\HarddiskVolume2\Users\Alissa Simpson\Downloads\desktop.ini
0x000000003e811df0     16      0 R--r-d \Device\HarddiskVolume2\Windows\System32\SearchFolder.dll
0x000000003e813070      4      0 R--r-d \Device\HarddiskVolume2\Windows\System32\p2pcollab.dll
0x000000003e813260      6      0 R--r-d \Device\HarddiskVolume2\Windows\System32\ListSvc.dll
0x000000003e813bf0     13      0 R--r-d \Device\HarddiskVolume2\Windows\System32\mssitlb.dll
0x000000003e813d40      1      1 R--r-d \Device\HarddiskVolume2\Windows\System32\en-US\FirewallAPI.dll.mui
0x000000003e815f20      9      0 R--r-d \Device\HarddiskVolume2\Windows\System32\wlanhlp.dll
0x000000003e817070      1      0 R--rwd \Device\HarddiskVolume2\Users\Alissa Simpson\Favorites\Links for United States\desktop.ini
```

Obrázok 10 Ukážka výpisu pluginu filescan

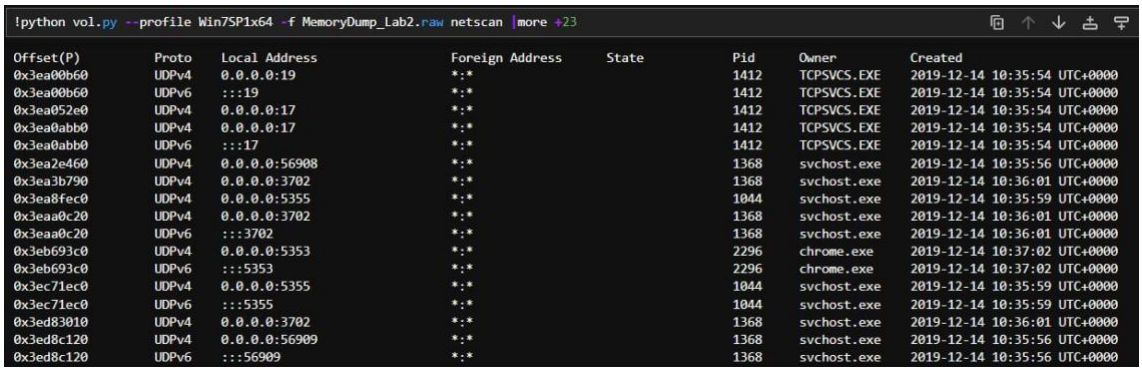
mftparser - Plugin mftparser slúži na analýzu tabuľky Master File Table (MFT), ktorá je dôležitou súčasťou súborového systému NTFS používaného v operačných systémoch Windows. Funguje ako centrálny index, ktorý sleduje všetky súbory a priečinky uložené na zväzku NTFS.

Zobrazuje informácie ako: názov súboru, cestu k súboru, veľkosť súboru, dátum a čas vytvorenia súboru, dátum a čas úpravy súboru a atribúty súboru. Umožňuje analyzovať informácie o súboroch uložených na NTFS diskoch. Obrázok 11 zobrazuje výstup niektorých dát z obrazu operačnej pamäte MemoryDump_Lab2.raw spusteného v prostredí jupyter notebook.



Obrázok 11 Ukážka výpisu Volatility 2 pluginu mftparser

netscan - Plugin netscan slúži na zobrazenie zoznamu sieťových pripojení a aktivít v systéme. Zobrazuje lokálnu IP adresu, vzdialenú IP adresu, protokol (TCP, UDP, ICMP), stav pripojenia (pripojené, odpojené) a porty. Umožňuje zistiť, s akými systémami a službami komunikoval systém. Obrázok 12 je zobrazuje výstup niektorých dát z obrazu operačnej pamäte MemoryDump_Lab2.raw spusteného v prostredí jupyter notebook.



Obrázok 12 Ukážka výpisu Volatility 2 pluginu netscan

psxview - Plugin psxview slúži na zobrazenie informácií o procesoch bežiacich v systéme, vrátane skrytých procesov a ich aktivít, napríklad pomocou pslistu a psscanu. Pomáha to pri detekcii skrytých a neviazaných procesov z jedného výstupného okna. Zobrazuje informácie ako ID procesu, meno procesu, stav procesu (beží, pozastavený, ukončený), prioritu procesu, cestu k súboru s programom, argumenty príkazového riadka a pamäťové využitie procesu. Umožňuje zistiť, aké procesy boli v systéme spustené. Obrázok 13 je zobrazuje výstup niektorých dát z obrazu operačnej pamäte MemoryDump_Lab1.raw spusteného v prostredí jupyter notebook.

```
!python vol.py --profile Win7SP1x64 -f MemoryDump_Lab1.raw psxview |more +23
```

Offset(P)	Name	PID	pslist	psscan	thrdproc	pspcid	csrss	session	deskthrd	ExitTime
0x000000003eea0580	lsass.exe	492	True	True	True	True	True	True	False	
0x000000003febb400	sppsvc.exe	1508	True	True	True	True	True	True	True	
0x000000003fd9a4e0	VBoxTray.exe	2304	True	True	True	True	True	True	True	
0x000000003fa48060	DumpIt.exe	796	True	True	True	True	True	True	True	
0x000000003ed58890	svchost.exe	1372	True	True	True	True	True	True	True	
0x000000003ea46960	explorer.exe	604	True	True	True	True	True	True	True	
0x000000003efa5b30	svchost.exe	852	True	True	True	True	True	True	True	
0x000000003e8199e0	svchost.exe	2368	True	True	True	True	True	True	False	
0x000000003fccea60	SearchFilterHo	1720	True	True	True	True	True	True	True	
0x000000003ef8c420	svchost.exe	816	True	True	True	True	True	True	True	
0x000000003ec1bb30	svchost.exe	472	True	True	True	True	True	True	True	
0x000000003fd4db30	dwm.exe	3004	True	True	True	True	True	True	True	
0x000000003fccbb30	winlogon.exe	2808	True	True	True	True	True	True	True	
0x000000003effa910	dwm.exe	1988	True	True	True	True	True	True	False	
0x000000003ef49b30	svchost.exe	720	True	True	True	True	True	True	True	
0x000000003fdff630	SearchProtocol	2524	True	True	True	True	True	True	True	

Obrázok 13 Ukážka výpisu Volatility 2 pluginu psxview

thrdscan - Plugin thrdscan slúži na zobrazenie zoznamu vlákien (threads) v systéme. Zobrazuje ID vlákna, ID procesu, ku ktorému vlákno patrí, stav vlákna (beží, pozastavené, ukončené), priorita vlákna a funkciu, v ktorej vlákno momentálne beží. Umožňuje zistiť, aké vlákna boli v systéme spustené a aké aktivity vykonávali. Obrázok 14 je zobrazuje výstup niektorých dát z obrazu operačnej pamäte MemoryDump_Lab2.raw spusteného v prostredí jupyter notebook.

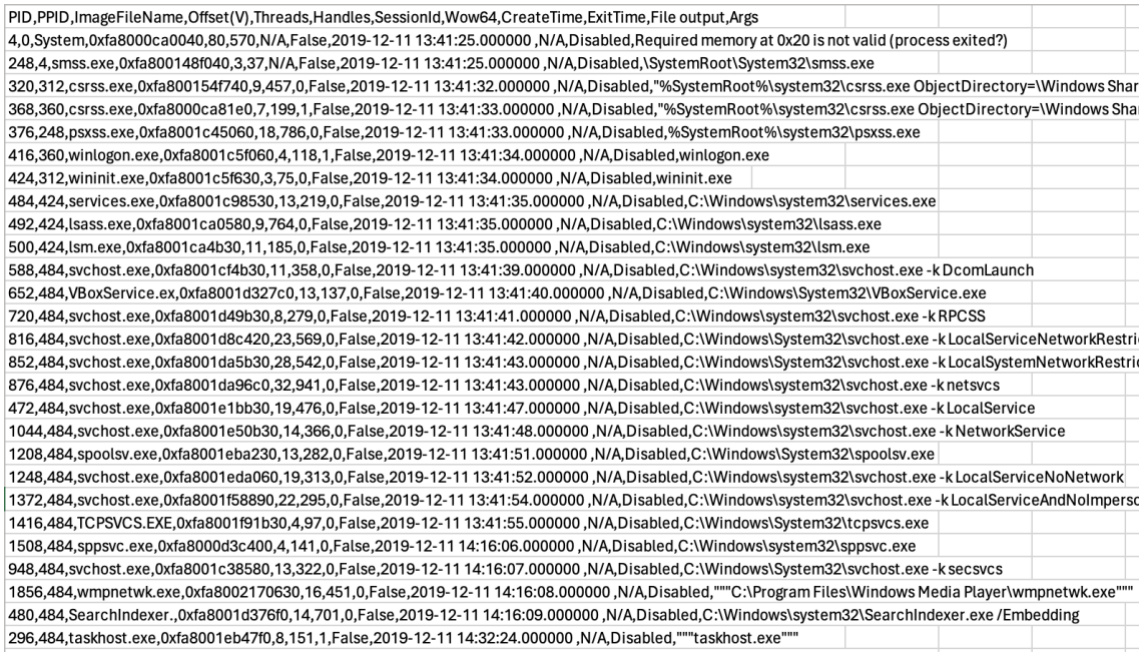
```
!python vol.py --profile Win7SP1x64 -f MemoryDump_Lab2.raw thrdscan
Offset(P)             PID    TID      Start Address Create Time                   Exit Time
------------------ ------ ------ ------------------ ------------------------------ ------------------------------
0x000000003e807060   1896   1740         0x778bc500 2019-12-14 10:36:15 UTC+0000
0x000000003e808060   1064   1696         0x778bc500 2019-12-14 10:36:15 UTC+0000
0x000000003e80a060   1896   1308         0x778bc500 2019-12-14 10:36:15 UTC+0000
0x000000003e80a660   1896   1172         0x778bc500 2019-12-14 10:36:15 UTC+0000
0x000000003e80ab60   1896   1176         0x778bc500 2019-12-14 10:36:15 UTC+0000   2019-12-14 10:36:30 UTC+0000
0x000000003e80b060   1896   1828         0x778bc500 2019-12-14 10:36:15 UTC+0000
0x000000003e80c060   1896   1572         0x778bc500 2019-12-14 10:36:15 UTC+0000
0x000000003e80c620   1896   1880         0x778bc500 2019-12-14 10:36:15 UTC+0000
0x000000003e80d990   1896   1904         0x778bc500 2019-12-14 10:36:15 UTC+0000
0x000000003e810300   1896    916         0x778bc500 2019-12-14 10:36:15 UTC+0000
0x000000003e818230   1076   1532         0x778bc500 2019-12-14 10:38:02 UTC+0000
0x000000003e81a320   1064   2216         0x778bc500 2019-12-14 10:36:20 UTC+0000
0x000000003e81c060   1064   2236         0x778bc500 2019-12-14 10:36:21 UTC+0000
0x000000003e81d9e0   2308   3004         0x778bc500 2019-12-14 10:36:50 UTC+0000
```

Obrázok 14 Ukážka výpisu Volatility 2 pluginu thrdscan

3.3 CSV súbory

Jednou z motivácie našej práce, bolo zjednodušiť analýzu dát pamäte forenzným analytikom, spôsobom lepšie zobraziteľných a jednoduchšie čitateľných dát z operačnej pamäte, výstup sme sa rozhodli zvoliť formát CSV súborov.

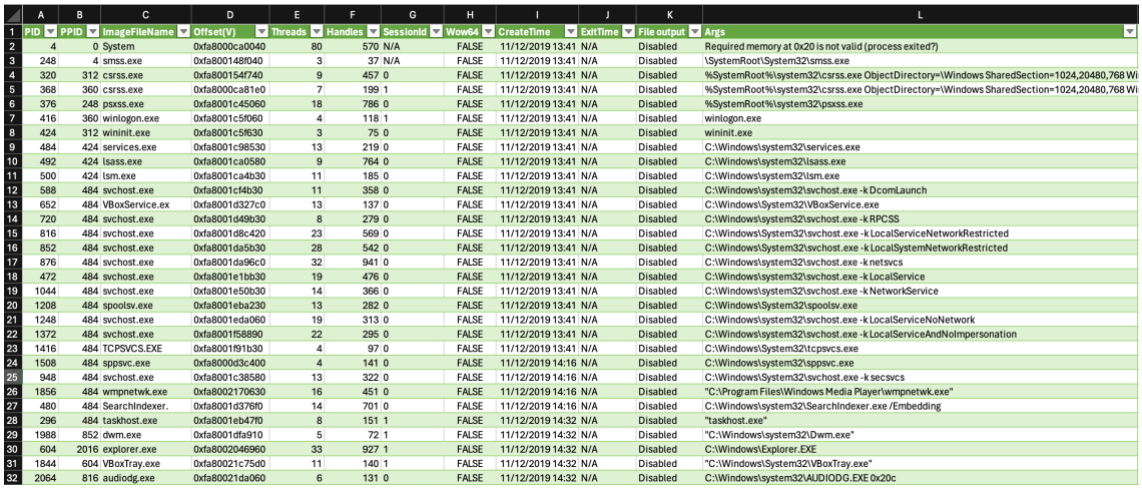
Prvým z výstupov boli parsované štruktúrované dáta, viď. Obrázok 15, získané využitím jednotlivých pluginov oboch verzií Volatility prehľadne zapisovaných do CSV súborov využitím čiarky ako oddeľovaču dát v jednom riadku.



```
PID,PPID,ImageFileName,Offset(V),Threads,Handles,SessionId,Wow64,CreateTime,ExitTime,File output,Args
4,0,System,0xfa8000ca0040,80,570,N/A,False,2019-12-11 13:41:25.000000 ,N/A,Disabled,Required memory at 0x20 is not valid (process exited?)
248,4,smss.exe,0xfa800148f040,3,37,N/A,False,2019-12-11 13:41:25.000000 ,N/A,Disabled,\SystemRoot\System32\smss.exe
320,312,csrss.exe,0xfa800154f740,9,457,0,False,2019-12-11 13:41:32.000000 ,N/A,Disabled,"%SystemRoot%\system32\csrss.exe ObjectDirectory=\Windows Shar
368,360,csrss.exe,0xfa8000ca81e0,7,199,1,False,2019-12-11 13:41:33.000000 ,N/A,Disabled,"%SystemRoot%\system32\csrss.exe ObjectDirectory=\Windows Sha
376,248,psxss.exe,0xfa8001c45060,18,786,0,False,2019-12-11 13:41:33.000000 ,N/A,Disabled,%SystemRoot%\system32\psxss.exe
416,360,winlogon.exe,0xfa8001c5f060,4,118,1,False,2019-12-11 13:41:34.000000 ,N/A,Disabled,winlogon.exe
424,312,wininit.exe,0xfa8001c5f630,3,75,0,False,2019-12-11 13:41:34.000000 ,N/A,Disabled,wininit.exe
484,424,services.exe,0xfa8001c98530,13,219,0,False,2019-12-11 13:41:35.000000 ,N/A,Disabled,C:\Windows\system32\services.exe
492,424,lsass.exe,0xfa8001ca0580,9,764,0,False,2019-12-11 13:41:35.000000 ,N/A,Disabled,C:\Windows\system32\lsass.exe
500,424,lsm.exe,0xfa8001ca4b30,11,185,0,False,2019-12-11 13:41:35.000000 ,N/A,Disabled,C:\Windows\system32\lsm.exe
588,484,svchost.exe,0xfa8001cf4b30,11,358,0,False,2019-12-11 13:41:39.000000 ,N/A,Disabled,C:\Windows\system32\svchost.exe -k DcomLaunch
652,484,VBoxService.ex,0xfa8001d327c0,13,137,0,False,2019-12-11 13:41:40.000000 ,N/A,Disabled,C:\Windows\System32\VBoxService.exe
720,484,svchost.exe,0xfa8001d49b30,8,279,0,False,2019-12-11 13:41:41.000000 ,N/A,Disabled,C:\Windows\system32\svchost.exe -k RPCSS
816,484,svchost.exe,0xfa8001d8c420,23,569,0,False,2019-12-11 13:41:42.000000 ,N/A,Disabled,C:\Windows\System32\svchost.exe -k LocalServiceNetworkRestri
852,484,svchost.exe,0xfa8001da5b30,28,542,0,False,2019-12-11 13:41:43.000000 ,N/A,Disabled,C:\Windows\System32\svchost.exe -k LocalSystemNetworkRestri
876,484,svchost.exe,0xfa8001da96c0,32,941,0,False,2019-12-11 13:41:43.000000 ,N/A,Disabled,C:\Windows\system32\svchost.exe -k netsvcs
472,484,svchost.exe,0xfa8001e1bb30,19,476,0,False,2019-12-11 13:41:47.000000 ,N/A,Disabled,C:\Windows\system32\svchost.exe -k LocalService
1044,484,svchost.exe,0xfa8001e50b30,14,366,0,False,2019-12-11 13:41:48.000000 ,N/A,Disabled,C:\Windows\system32\svchost.exe -k NetworkService
1208,484,spoolsv.exe,0xfa8001eba230,13,282,0,False,2019-12-11 13:41:51.000000 ,N/A,Disabled,C:\Windows\System32\spoolsv.exe
1248,484,svchost.exe,0xfa8001eda060,19,313,0,False,2019-12-11 13:41:52.000000 ,N/A,Disabled,C:\Windows\system32\svchost.exe -k LocalServiceNoNetwork
1372,484,svchost.exe,0xfa8001f58890,22,295,0,False,2019-12-11 13:41:54.000000 ,N/A,Disabled,C:\Windows\system32\svchost.exe -k LocalServiceAndNoImpers
1416,484,TCPSVCS.EXE,0xfa8001f91b30,4,97,0,False,2019-12-11 13:41:55.000000 ,N/A,Disabled,C:\Windows\System32\tcpsvcs.exe
1508,484,sppsvc.exe,0xfa8000d3c400,4,141,0,False,2019-12-11 14:16:06.000000 ,N/A,Disabled,C:\Windows\system32\sppsvc.exe
948,484,svchost.exe,0xfa8001c38580,13,322,0,False,2019-12-11 14:16:07.000000 ,N/A,Disabled,C:\Windows\System32\svchost.exe -k secsvcs
1856,484,wmpnetwk.exe,0xfa8002170630,16,451,0,False,2019-12-11 14:16:08.000000 ,N/A,Disabled,"""C:\Program Files\Windows Media Player\wmpnetwk.exe"""
480,484,SearchIndexer.,0xfa8001d376f0,14,701,0,False,2019-12-11 14:16:09.000000 ,N/A,Disabled,C:\Windows\system32\SearchIndexer.exe /Embedding
296,484,taskhost.exe,0xfa8001eb47f0,8,151,1,False,2019-12-11 14:32:24.000000 ,N/A,Disabled,"""taskhost.exe"""
```

Obrázok 15 Náhľad CSV súboru parsovaných dát

Tento prístup umožňuje jednoduché načítanie a manipuláciu s dátami bez potreby zložitých analytických nástrojov. Obrázok 16 zobrazuje načítané dáta z CSV súboru do tabuľky pomocou automatickej možnosti načitania dát, kde oddeľovač sme zvolili čiarku.



	A	B	C	D	E	F	G	H	I	J	K	L
1	PID	PPID	ImageFileName	Offset(V)	Threads	Handles	SessionId	Wow64	CreateTime	ExitTime	File output	Args
2	4	0	System	0xfa8000ca0040	80	570	N/A	FALSE	11/12/2019 13:41	N/A	Disabled	Required memory at 0x20 is not valid (process exited?)
3	248	4	smss.exe	0xfa800148f040	3	37	N/A	FALSE	11/12/2019 13:41	N/A	Disabled	\SystemRoot\System32\smss.exe
4	320	312	csrss.exe	0xfa800154f740	9	457	0	FALSE	11/12/2019 13:41	N/A	Disabled	%SystemRoot%\system32\csrss.exe ObjectDirectory=\Windows SharedSection=1024,20480,768 Wi
5	368	360	csrss.exe	0xfa8000ca81e0	7	199	1	FALSE	11/12/2019 13:41	N/A	Disabled	%SystemRoot%\system32\csrss.exe ObjectDirectory=\Windows SharedSection=1024,20480,768 Wi
6	376	248	psxss.exe	0xfa8001c45060	18	786	0	FALSE	11/12/2019 13:41	N/A	Disabled	%SystemRoot%\system32\psxss.exe
7	416	360	winlogon.exe	0xfa8001c5f060	4	118	1	FALSE	11/12/2019 13:41	N/A	Disabled	winlogon.exe
8	424	312	wininit.exe	0xfa8001c5f630	3	75	0	FALSE	11/12/2019 13:41	N/A	Disabled	wininit.exe
9	484	424	services.exe	0xfa8001c98530	13	219	0	FALSE	11/12/2019 13:41	N/A	Disabled	C:\Windows\system32\services.exe
10	492	424	lsass.exe	0xfa8001ca0580	9	764	0	FALSE	11/12/2019 13:41	N/A	Disabled	C:\Windows\system32\lsass.exe
11	500	424	lsm.exe	0xfa8001ca4b30	11	185	0	FALSE	11/12/2019 13:41	N/A	Disabled	C:\Windows\system32\lsm.exe
12	588	484	svchost.exe	0xfa8001cf4b30	11	358	0	FALSE	11/12/2019 13:41	N/A	Disabled	C:\Windows\system32\svchost.exe -k DcomLaunch
13	652	484	VBoxService.ex	0xfa8001d327c0	13	137	0	FALSE	11/12/2019 13:41	N/A	Disabled	C:\Windows\System32\VBoxService.exe
14	720	484	svchost.exe	0xfa8001d49b30	8	279	0	FALSE	11/12/2019 13:41	N/A	Disabled	C:\Windows\system32\svchost.exe -k RPCSS
15	816	484	svchost.exe	0xfa8001d8c420	23	569	0	FALSE	11/12/2019 13:41	N/A	Disabled	C:\Windows\System32\svchost.exe -k LocalServiceNetworkRestricted
16	852	484	svchost.exe	0xfa8001da5b30	28	542	0	FALSE	11/12/2019 13:41	N/A	Disabled	C:\Windows\System32\svchost.exe -k LocalSystemNetworkRestricted
17	876	484	svchost.exe	0xfa8001da96c0	32	941	0	FALSE	11/12/2019 13:41	N/A	Disabled	C:\Windows\system32\svchost.exe -k netsvcs
18	472	484	svchost.exe	0xfa8001e1bb30	19	476	0	FALSE	11/12/2019 13:41	N/A	Disabled	C:\Windows\system32\svchost.exe -k LocalService
19	1044	484	svchost.exe	0xfa8001e50b30	14	366	0	FALSE	11/12/2019 13:41	N/A	Disabled	C:\Windows\system32\svchost.exe -k NetworkService
20	1208	484	spoolsv.exe	0xfa8001eba230	13	282	0	FALSE	11/12/2019 13:41	N/A	Disabled	C:\Windows\System32\spoolsv.exe
21	1248	484	svchost.exe	0xfa8001eda060	19	313	0	FALSE	11/12/2019 13:41	N/A	Disabled	C:\Windows\system32\svchost.exe -k LocalServiceNoNetwork
22	1372	484	svchost.exe	0xfa8001f58890	22	295	0	FALSE	11/12/2019 13:41	N/A	Disabled	C:\Windows\system32\svchost.exe -k LocalServiceAndNoImpersonation
23	1416	484	TCPSVCS.EXE	0xfa8001f91b30	4	97	0	FALSE	11/12/2019 13:41	N/A	Disabled	C:\Windows\System32\tcpsvcs.exe
24	1508	484	sppsvc.exe	0xfa8000d3c400	4	141	0	FALSE	11/12/2019 14:16	N/A	Disabled	C:\Windows\system32\sppsvc.exe
25	948	484	svchost.exe	0xfa8001c38580	13	322	0	FALSE	11/12/2019 14:16	N/A	Disabled	C:\Windows\System32\svchost.exe -k secsvcs
26	1856	484	wmpnetwk.exe	0xfa8002170630	16	451	0	FALSE	11/12/2019 14:16	N/A	Disabled	"C:\Program Files\Windows Media Player\wmpnetwk.exe"
27	480	484	SearchIndexer.	0xfa8001d376f0	14	701	0	FALSE	11/12/2019 14:16	N/A	Disabled	C:\Windows\system32\SearchIndexer.exe /Embedding
28	296	484	taskhost.exe	0xfa8001eb47f0	8	151	1	FALSE	11/12/2019 14:32	N/A	Disabled	"taskhost.exe"
29	1988	852	dwm.exe	0xfa8001dfa910	5	72	1	FALSE	11/12/2019 14:32	N/A	Disabled	"C:\Windows\system32\Dwm.exe"
30	604	2016	explorer.exe	0xfa8002046960	33	927	1	FALSE	11/12/2019 14:32	N/A	Disabled	C:\Windows\Explorer.EXE
31	1844	604	VBoxTray.exe	0xfa80021c75d0	11	140	1	FALSE	11/12/2019 14:32	N/A	Disabled	"C:\Windows\System32\VBoxTray.exe"
32	2064	816	audiodg.exe	0xfa80021da060	6	131	0	FALSE	11/12/2019 14:32	N/A	Disabled	C:\Windows\system32\AUDIODG.EXE 0x20c

Obrázok 16 Náhľad načítaných dát v exceli

Taktiež sme si však uvedomili, že rovnako dôležité sú vzťahy medzi jednotlivými dátami. Na tento účel sme dáta v jednotlivých CSV súboroch spojili použitím identifikátora procesu (PID). Obrázok 17 zobrazuje spojené dáta z piatich pluginov, presnejšie sú to pslist, netscan, cmdline, getsids, dlllist a ldrmodules poprepájané na základe PID.

Tento krok spájania CSV súborov nám umožnil automatizovať získavanie výstupov bez nutnosti spúšťania všetkých pluginov naraz. Namiesto toho sme boli schopní získať výstupy z viacerých pluginov súčasne, čo viedlo k efektívnejšiemu spracovaniu a zvýšenej produktivite. Tento integrovaný prístup nielenže zjednodušuje prácu s dátami, ale aj zvyšuje celkovú efektivitu analytického procesu. Výstupy vo forme CSV súborov sú ľahko interpretovateľné a umožňujú rýchlejšie a efektívnejšie rozhodovanie na základe získaných údajov. Týmto spôsobom sme dosiahli cieľ vylepšenia prístupu k dátam z pluginov a optimalizovali proces ich analýzy a využitia.

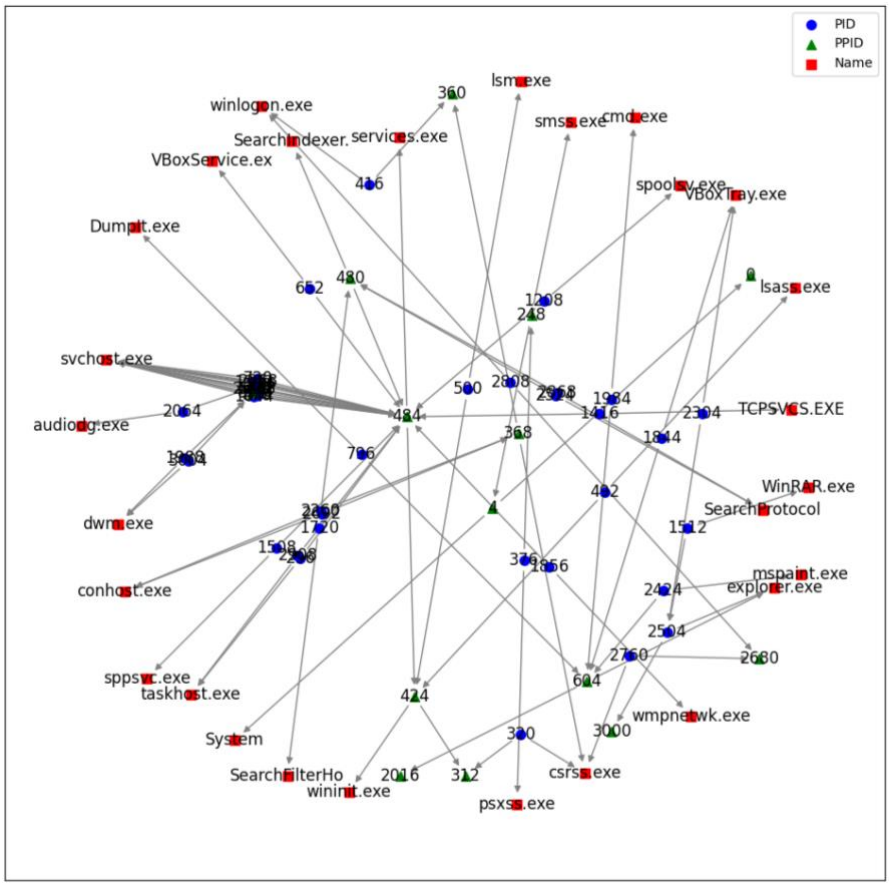
	A	B	C	D	E	F	G	H
1	PID	LocalAddress	ForeignAddress	Args	SID	Name	Path	MappedPath
2	4	0.0.0.0:17	*:*	Required memory at 0x20 is not valid (proc	S-1-5-18	Local System	\SystemRoot\System32\smss.exe	\Windows\System32\ntdll.dll
3	248	0.0.0.0:17	*:*	\SystemRoot\System32\smss.exe	S-1-5-32-544	Administrators	C:\Windows\SYSTEM32\ntdll.dll	\Windows\SysWOW64\ntdll.dll
4	320	:::17	*:*	%SystemRoot%\system32\csrss.exe Object	S-1-1-0	Everyone	C:\Windows\system32\csrss.exe	\Windows\System32\smss.exe
5	368	0.0.0.0:56908	*:*	%SystemRoot%\system32\csrss.exe Object	S-1-5-11	Authenticated Users	C:\Windows\SYSTEM32\ntdll.dll	\Windows\System32\ntdll.dll
6	376	0.0.0.0:3702	*:*	%SystemRoot%\system32\psxss.exe	S-1-16-16384	System Mandatory Level	C:\Windows\system32\CSRSRV.dll	\Windows\Fonts\vgasys.fon
7	416	0.0.0.0:5355	*:*	winlogon.exe	S-1-5-18	Local System	C:\Windows\system32\basesrv.DLL	\Windows\Fonts\cga40woa.fon
8	424	0.0.0.0:3702	*:*	wininit.exe	S-1-5-32-544	Administrators	C:\Windows\system32\winsrv.DLL	\Windows\Fonts\vgaoem.fon
9	484	:::3702	*:*	C:\Windows\system32\services.exe	S-1-1-0	Everyone	C:\Windows\system32\USER32.dll	\Windows\Fonts\dosapp.fon
10	492	0.0.0.0:5353	*:*	C:\Windows\system32\lsass.exe	S-1-5-11	Authenticated Users	C:\Windows\system32\GDI32.dll	\Windows\Fonts\ega40woa.fon
11	500	:::5353	*:*	C:\Windows\system32\lsm.exe	S-1-16-16384	System Mandatory Level	C:\Windows\SYSTEM32\kernel32.dll	\Windows\Fonts\cga80woa.fon
12	588	0.0.0.0:5355	*:*	C:\Windows\system32\svchost.exe -k Dcor	S-1-5-18	Local System	C:\Windows\system32\KERNELBASE.dll	\Windows\System32\csrss.exe
13	652	:::5355	*:*	C:\Windows\System32\VBoxService.exe	S-1-5-32-544	Administrators	C:\Windows\system32\LPK.dll	\Windows\System32\ntdll.dll
14	720	0.0.0.0:3702	*:*	C:\Windows\system32\svchost.exe -k RPCS	S-1-1-0	Everyone	C:\Windows\system32\USP10.dll	\Windows\System32\user32.dll
15	816	0.0.0.0:56909	*:*	C:\Windows\System32\svchost.exe -k Local	S-1-5-11	Authenticated Users	C:\Windows\system32\msvcrt.dll	\Windows\System32\kernel32.dll
16	852	:::56909	*:*	C:\Windows\System32\svchost.exe -k Local	S-1-16-16384	System Mandatory Level	C:\Windows\system32\sxssrv.DLL	\Windows\System32\gdi32.dll
17	876	0.0.0.0:0	*:*	C:\Windows\system32\svchost.exe -k netsv	S-1-5-18	Local System	C:\Windows\system32\sxs.dll	\Windows\System32\winsrv.dll
18	472	0.0.0.0:7	*:*	C:\Windows\system32\svchost.exe -k Local	S-1-5-32-544	Administrators	C:\Windows\system32\RPCRT4.dll	\Windows\System32\sxs.dll
19	1044	:::7	*:*	C:\Windows\system32\svchost.exe -k Netw	S-1-1-0	Everyone	C:\Windows\system32\CRYPTBASE.dll	\Windows\System32\cryptbase.dll
20	1208	0.0.0.0:7	*:*	C:\Windows\System32\spoolsv.exe	S-1-5-11	Authenticated Users	C:\Windows\system32\ADVAPI32.dll	\Windows\System32\sxssrv.dll
21	1248	0.0.0.0:9	*:*	C:\Windows\system32\svchost.exe -k Local	S-1-16-16384	System Mandatory Level	C:\Windows\SYSTEM32\sechost.dll	\Windows\System32\rpcrt4.dll
22	1372	:::9	*:*	C:\Windows\system32\svchost.exe -k Local	S-1-5-18	Local System	C:\Windows\system32\csrss.exe	\Windows\System32\csrsrv.dll
23	1416	0.0.0.0:9	*:*	C:\Windows\System32\tcpsvcs.exe	S-1-5-32-544	Administrators	C:\Windows\SYSTEM32\ntdll.dll	\Windows\System32\basesrv.dll
24	1508	10.0.2.15:138	*:*	C:\Windows\system32\sppsvc.exe	S-1-1-0	Everyone	C:\Windows\system32\CSRSRV.dll	\Windows\System32\KernelBase.dll
25	948	0.0.0.0:0	*:*	C:\Windows\System32\svchost.exe -k secsv	S-1-5-11	Authenticated Users	C:\Windows\system32\basesrv.DLL	\Windows\System32\lpk.dll
26	1856	:::0	*:*	"C:\Program Files\Windows Media Player\	S-1-16-16384	System Mandatory Level	C:\Windows\system32\winsrv.DLL	\Windows\System32\advapi32.dll
27	480	0.0.0.0:13	*:*	C:\Windows\system32\SearchIndexer.exe	S-1-5-18	Local System	C:\Windows\system32\USER32.dll	\Windows\System32\sechost.dll
28	296	0.0.0.0:13	*:*	"taskhost.exe"	S-1-5-32-544	Administrators	C:\Windows\system32\GDI32.dll	\Windows\System32\usp10.dll
29	1988	:::13	*:*	"C:\Windows\system32\Dwm.exe"	S-1-1-0	Everyone	C:\Windows\SYSTEM32\kernel32.dll	\Windows\System32\msvcrt.dll

Obrázok 17 Náhľad spojených dát v exceli

3.4 Vizualizácia pomocou grafov

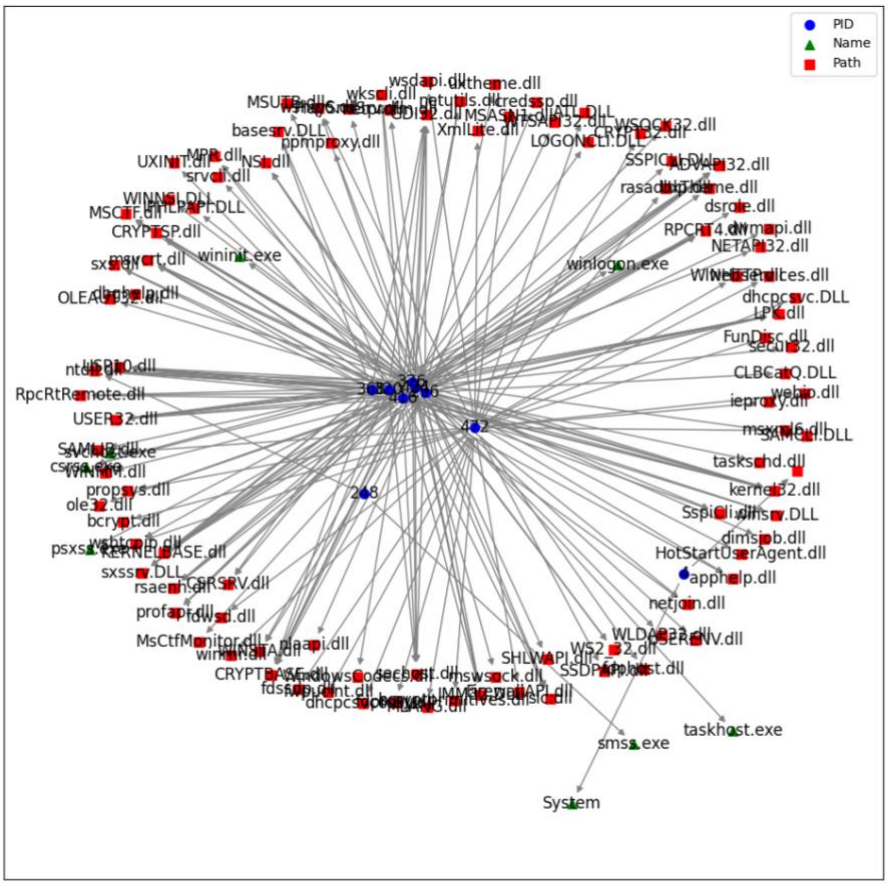
V tejto kapitole sa bližšie zameriame na vytvorené grafy jednotlivo a poukážeme na vzťahy medzi dátami v operačnej pamäti. Presnejšie sa pozrieme na šesť grafov, na ktorých sú vizualizované vzťahy samotných procesov, na základe ID procesu, a rôzne ďalšie napojenia na konkrétne procesy.

Obrázok 18 zobrazuje graf, z ktorého je možné vyčítať procesy a ich nadradené procesy, inak nazývané aj rodičovské procesy (parent process ID - PPID) a názvy procesov. Orientovaná šípka z ID procesu smeruje k ID nadradenému procesu, čo značí, ktorý proces, je ktorému nadradený. Názvy procesov sú priradené k procesom taktiež orientovanou šípkou, ktorá smeruje z PID, respektíve PPID k jeho názvu. Možno zachytiť, že jeden proces môže byť nadradený viacerým procesom, taktiež nadradený proces môže mať iný proces ako svoj nadradený. Takýmto spôsobom vieme zachytiť nejaký vyčnievajúci proces, ktorý môže evokovať podozrenie.



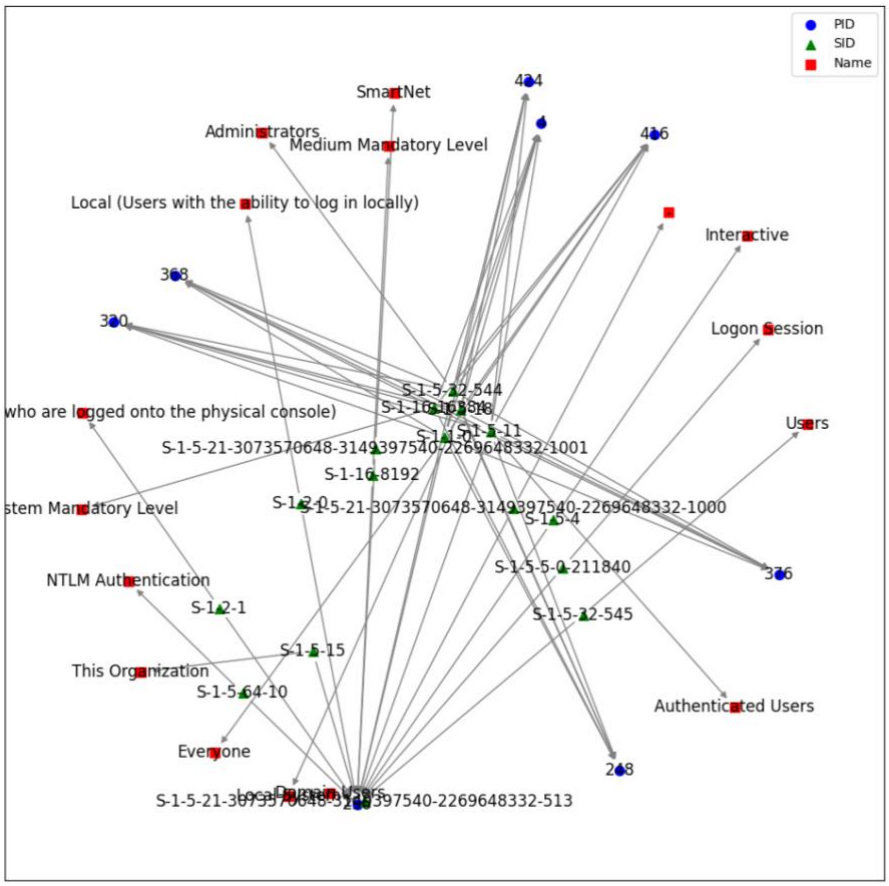
Obrázok 18 Graf s informáciami o procesoch

Obrázok 19 je graf, ktorý zobrazuje vzťahy procesov a dll knižníc, z ktorých sme získali iba konkrétny názov dll súboru z absolútnej cesty, ktorá bola poskytnutá pri výpise. V tomto prípade vzťahov môžeme vidieť veľké množstvo dll knižníc, s ktorými jednotlivé procesy komunikovali, s ktorými nejakým spôsobom nadviazali vzťah. Z dôvodu čitateľnosti sme vybrali iba niekoľko dát, avšak i z malého počtu vybraných procesov je možné vidieť, koľko knižníc bolo využitých. Pri týchto dll knižniciach je dobré sledovať počet vzťahov rôznych knižníc jedného procesu, prípadne množstvo vzťahov jednej dll knižnice s viacerými procesmi.



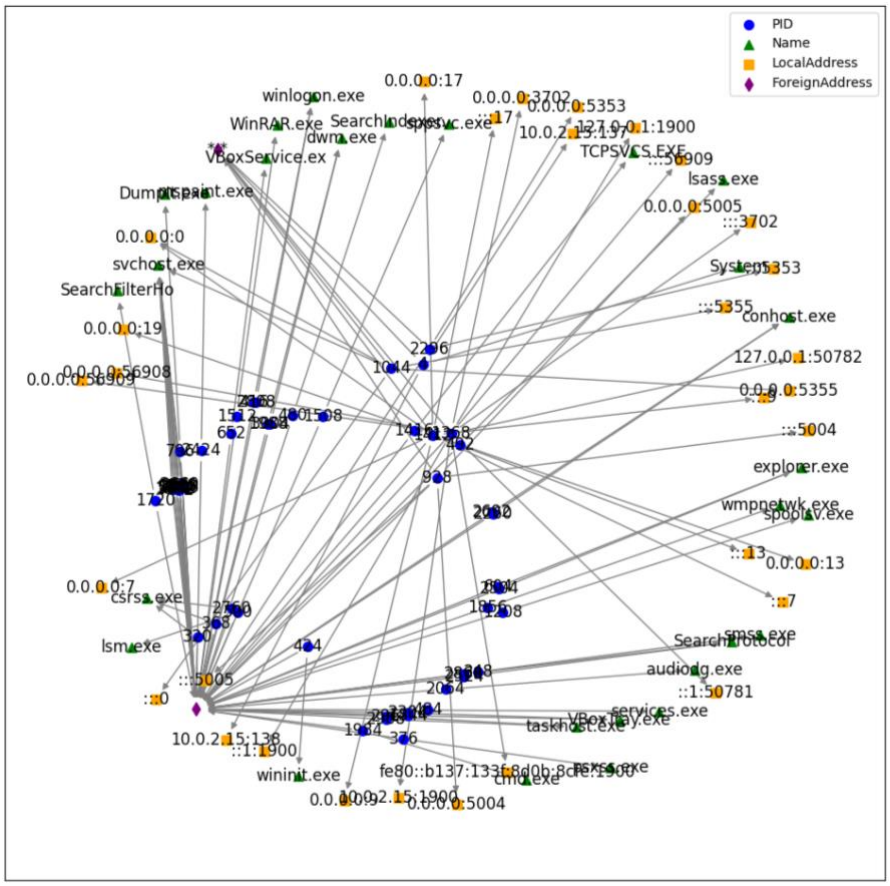
Obrázok 19 Graf vzťahov procesov a dll knižníc

Nasledujúcim grafom sledujeme vzťahy jednotlivých používateľov procesov. Vieme pristúpiť k jednotlivému používateľovi na základe jeho identifikátora a zároveň máme možnosť získať aj jeho názov. Obrázok 20 vizualizuje, ktorí používatelia pristupovali ku ktorým súborom. Je dobré mať vyobrazených používateľov, ktorí pristupovali k procesom, aby mohli analytici skôr zaznamenať, ktorý používateľ je podozrivý, prípadne útočník.



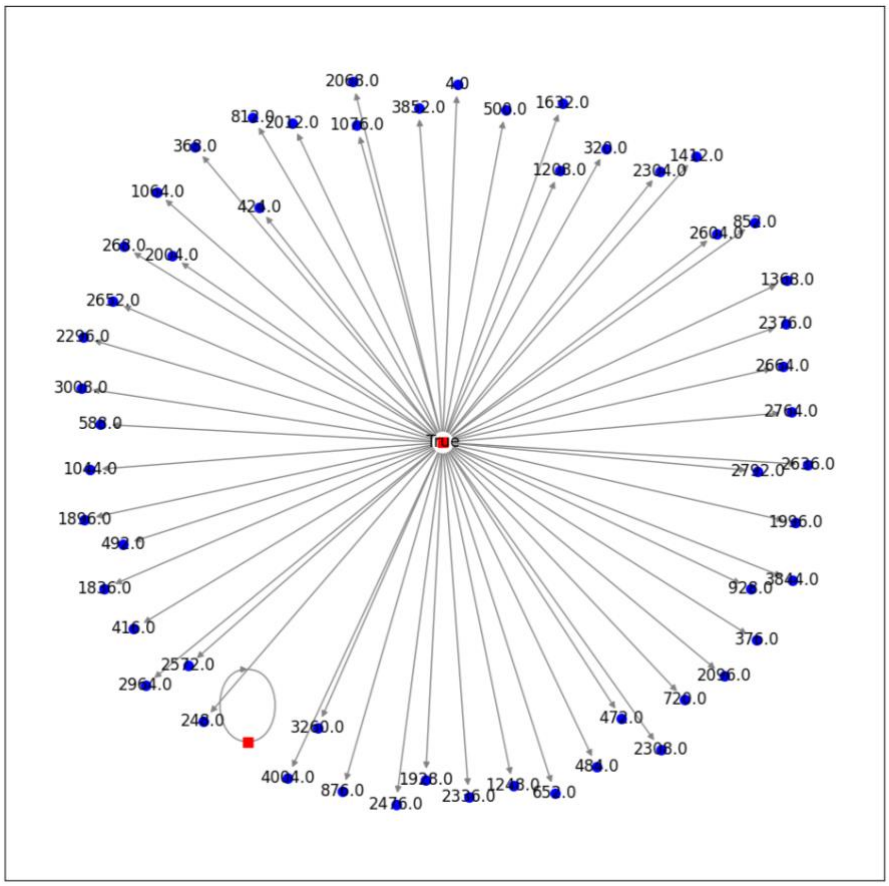
Obrázok 20 Graf vzťahov procesov a používateľov

Obrázok 21 Graf vzťahov procesov so sieťovými aktivitami nám grafovou vizualizáciou jasne znázorňuje sieťové aktivity jednotlivých procesov v systéme. Pre procesy, z ktorých vzťahová väzba smeruje iba na lokálnu adresu ukazuje, na ktorých lokálnych adresách proces počúva. Ak orientovaná šípka smeruje iba k cudzej adrese, znázorňuje to, na aké adresy sa proces aktívne pripája. Ak šípky smerujú aj k lokálnej adrese a aj cudzej, symbolizuje to pripájanie a zároveň počúvanie. Takouto vizualizáciou vieme sledovať množstvo pripojených zariadení a množstvo pripájaní sa procesov na cudzie adresy.



Obrázok 21 Graf vzťahov procesov so sieťovými aktivitami

Posledným z našich grafov, je Obrázok 22, vďaka ktorému môžu analytici zmapovať procesy, ktoré sú skryté v pamäti. Sledujeme to tým, že ak je proces, a teda PID zaznamenané v pslist, avšak v psscan nie, znamená to, že proces je skrytý, avšak nachádza sa v pamäti. Zameriavame sa na hodnoty True a False. V našom prípade sú všetky procesy zaznamenané aj v pslist a aj v psscan a teda sa v dátach nevyskytuje ani raz možnosť False.



Obrázok 22 Graf zobrazujúci skryté procesy

3.5 Vyhodnotenie

V našej práci sme využili technológiu AccessData FTK Imager na zaistenie operačnej pamäte. Na analýzu obrazu operačnej pamäte sme zvolili technológiu Volatility verzie 2 a zároveň aj Volatility verzie 3. Použitie oboch verzií Volatility zapríčinilo, že niektoré pluginy staršej verie neboli súčasťou verzie 3. Ako sme opísali v 3. kapitole tejto práce, využili sme aspoň osem pluginov Volatility 3 a najmenej päť pluginov Volatility 2. Zápis do textového súboru a následné parsovanie a vytváranie CSV súborov prebehlo s dátami takmer každého spomenutého pluginu, okrem pluginu mftparser Volatility 2 z dôvodu náročnosti výpisu, a to nesúmernosti dát a nejednotných oddeľovačov. Spoločnou hodnotou v najväčšom množstve dát bol artefakt označujúci PID, a teda ID procesu, na základe ktorého sme spájali dáta z pluginov. Spojili sme do jedného CSV viacero výstupov pluginov, presnejšie pluginy ako pslist, getsids, thrdscan, cmdline, dlllist, ldrmodules, netscan, psxview a vadinfo.

Následne sme prešli k vizualizácií dát, kde sme vykreslili grafy vzťahov niektorých nami určených dát v spojitosti k procesom, keďže naše spájanie dát záležalo na PID, ktoré obsahuje výstup každého pluginu.

Záver

Cieľom tejto práce bolo porovnanie aktuálnych prístupov k forenznej analýze operačnej pamäte, analýza a spracovanie forenzných artefaktov obsiahnutých v operačnej pamäti a návrh, implementácia a overenie nástroja na automatizáciu forenznej analýzy operačnej pamäte. Na základe porovnania sme vytvorili a implementovali nástroj overený na štyroch rôznych obrazoch operačnej pamäte pre automatizáciu forenznej analýzy, ktorý vedie k redukcii počtu spúšťaní príkazov pre výpis forenzných artefaktov a zjednoduší analýzu a spracovanie forenzných artefaktov operačnej pamäte.

V prvej kapitole sme si definovali pojmy týkajúce sa tejto problematiky, primárne išlo o forenznú analýzu, zaisťovanie operačnej pamäte, kde sme sa pozreli aj na rôzne technológie pre prístup k operačnej pamäti, z ktorých sme zvolili AccessData FTK Imager ako najvhodnejší a využili sme ho pri zaistení našej operačnej pamäte a taktiež sme sa zamerali na samotnú analýzu operačnej pamäte, kde sme si vysvetlili aj forenzné artefakty operačnej pamäte.

Druhá kapitola patrila nástrojom na analýzu operačnej pamäte. Najprv sme si špecifikovali jednotlivé nástroje Volatility 2, Volatility 3 a Rekall, kde následne sme porovnali ich možnosti, klady a zápory, z čoho sme sa zamerali na najlepší nástroj na analýzu operačnej pamäte, a to Volatility 3.

V tretej kapitole tejto práce vysvetľujeme nami vytvorený návrh na riešenie problému práce, a to výberom technológií na implementáciu, kde sme použili kombinácie pluginov Volatility 2 a Volatility 3 pre získanie forenzných artefaktov, Python a knižnicu pandas pre parsovanie a zapisovanie do vytvorených CSV súborov ako výstupov. Na záver sme vizualizovali vzťahy pomocou grafovej štruktúry v Pythone využitím knižníc matplotlib a networkx.

Výsledkom našej práce je teda automatizácia prístupu a zjednodušenie analýzy forenzných artefaktov operačnej pamäte vytvorením nástroja na parsovanie a spájanie dát s CSV výstupom jednoducho načítateľným v exceli a vizualizácia vzťahov artefaktov pamäte pomocou grafovej štruktúry.

Zoznam použitej literatúry

1. Forenzná analýza (Forensic Analysis). In: ManagementMania.com [online]. 2015 [cit. 2024-05-13]. Dostupné z: https://managementmania.com/sk/forenzna-analyza-forensic-analysis
2. Johansen, G. (2017). Digital forensics and incident response. Packt Publishing Ltd.
3. Frawley, Richard T., Memory Forensics: Effective Digital Forensics Investigations Basics In *ADF News* [online]. 2023 [cit. 2024-05-13]. Dostupné z: https://www.adfsolutions.com/adf-blog/memory-forensics-101-the-basics-you-need-to-know-for-effective-digital-forensics-investigations
4. What Is Fileless Malware? [online]. [cit. 2024-05-11]. Dostupné z: https://www.trellix.com/en-us/security-awareness/ransomware/what-is-filelessmalware.html
5. Cyberattacks 2021 In *Statistics from the Last Year* [online]. 2022 [cit. 2024-05-13]. Dostupné z: https://spanning.com/blog/cyberattacks-2021-phishingransomware-data-breach-statistics/
6. Nyholm, H., Monteith, K., Lyles, S., Gallegos, M., DeSantis, M., Donaldson, J., & Taylor, C. (2022). The evolution of volatile memory forensics. Journal of Cybersecurity and Privacy, 2(3), 556-572
7. Brezinski D., Guidelines for Evidence Collection and Archiving [online]. 2002 [cit. 2024-03-07]. Dostupné z: https://datatracker.ietf.org/doc/html/rfc3227
8. Ligh, M. H., Case, A., Levy, J., & Walters, A. (2014). The art of memory forensics: detecting malware and threats in windows, linux, and Mac memory. John Wiley & Sons.
9. KnTTools [online]. Dostupné z: http://www.gmgsystemsinc.com/knttools/
10. F-Response [online]. Dostupné z: https://www.f-response.com/
11. Mandiant Memoryze [online]. Dostupné z: https://www.mandiant.com/
12. HBGary FastDump [online]. Dostupné z: https://windowsir.blogspot.com/2009/02/hbgary-fastdump-and-responder.html
13. MoonSols [online]. Dostupné z: http://www.moonsols.com/wp-content/plugins/download-monitor/download.php?id=7

14. AccessData FTK Imager [online]. Dostupné z: https://www.exterro.com/digital-forensics-software/ftk-imager
15. WinEn [online]. Dostupné z: http://forensiczone.blogspot.com/2008/06/winenexe-ram-imaging-tool-included-in.html
16. Belkasoft [online]. Dostupné z: https://belkasoft.com/ram-capturer
17. ATC-NY [online]. Dostupné z: https://www.prnewswire.com/news-releases/atc-ny-announces-new-computer-forensic-tool---windows-memory-reader-183336621.html
18. Winpmem [online]. Dostupné z: https://winpmem.velocidex.com/
19. Latzo, T., Palutke, R., & Freiling, F. (2019). A universal taxonomy and survey of forensic memory acquisition techniques. Digital Investigation, 28, 56-69
20. Case, A., & Richard III, G. G. (2017). Memory forensics: The path forward. Digital investigation, 20, 23-33
21. Lucideus. Windows Volatile Memory Acquisition In *Forensics 2018 | Lucideus Forensics* [online]. 2018 [cit. 2024-05-05]. Dostupné z: https://medium.com/@lucideus/windows-volatile-memory-acquisition-forensics-2018-lucideus-forensics-3f297d0e5bfd
22. Kamal, K. M. A., Alfadel, M., & Munia, M. S. (2016, December). Memory forensics tools: Comparing processing time and left artifacts on volatile memory. In 2016 International Workshop on Computational Intelligence (IWCI) (pp. 84-90). IEEE
23. Hassan, N. A. (2019). Digital forensics basics: A practical guide using Windows OS. Apress.
24. What is digital forensics and incident response (DFIR)? [online]. 2022 [cit. 2024-01-25]. Dostupné z: https://www.ibm.com/topics/dfir
25. Volatility 2 [online] 2020. Dostupné z: https://github.com/volatilityfoundation/volatility
26. Volatility 3 [online] 2024. Dostupné z: https://github.com/volatilityfoundation/volatility3
27. Rekall [online] 2020. Dostupné z: https://github.com/google/rekall
28. MemLabs [online] 2021. Dostupné z: https://github.com/stuxnet999/MemLabs

29. The Case of the Stolen Szechuan Sauce. [online] 2020. Dostupné z: https://dfirmadness.com/the-stolen-szechuan-sauce/

30. Jupyter Notebook [online] 2024. Dostupné z: https://jupyter.org/

Prílohy

Príloha A: Zdrojové kódy na implementáciu nástroja na automatizáciu forenznej analýzy operačnej pamäte