Univerzita Pavla Jozefa Šafárika v Košiciach

Prírodovedecká fakulta

ANALÝZA VZŤAHOV MEDZI FORENZNÝMI ARTEFAKTAMI S VYUŽITÍM TEÓRIE GRAFOV

BAKALÁRSKA PRÁCA

Študijný odbor:	**Analýza dát a umelá inteligencia**
Školiace pracovisko:	**Ústav informatiky**
Vedúci práce:	**doc. RNDr. JUDr. Pavol Sokol, PhD.**
Konzultant:	**RNDr. Eva Marková**

Košice 2024

Sophia Petra Krišáková

Poďakovanie

Touto cestou by som chcela poďakovať svojmu vedúcemu práce doc. JUDr. RNDr. Pavlovi Sokolovi, PhD. za cenné rady, odborné vedenie a motiváciu. Tiež by som chcela vyjadriť vďaku svojej konzultanke RNDr. Eve Markovej za odbornú pomoc pri spracovaní práce.

36145173622977796

Univerzita P. J. Šafárika v Košiciach
Prírodovedecká fakulta

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Sophia Petra Krišáková
Študijný program: analýza dát a umelá inteligencia (medziodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: Informatika, Matematika
Typ záverečnej práce: Bakalárska práca
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Analýza vzťahov medzi forenznými artefaktami s využitím teórie grafov

Názov EN: Analysis of relationships between forensic artifacts using graph theory

Cieľ: (1) Porovnať aktuálne prístupy k analýze forenzných artefaktov pomocou grafových algoritmov.
(2) Navrhnúť vhodnú reprezentáciu forenzných artefaktov s využitím teórie grafov.
(3) Navrhnúť a implementovať grafový algoritmus na analýzu vzťahov medzi forenznými artefaktami.

Literatúra:
(1) Raschka, Sebastian, and Vahid Mirjalili. Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2. Packt Publishing Ltd, (2019).
(2) Iqbal, Salman, and Soltan Abed Alharbi. "Advancing automation in digital forensic investigations using machine learning forensics." Digital Forensic Science. IntechOpen, (2019).
(3) Palmer, Imani, Boris Gelfand, and Roy Campbell. "Exploring digital evidence with graph theory." (2017).
(4) Costantini, Stefania, Giovanni De Gasperis, and Raffaele Olivieri. "Digital forensics and investigations meet artificial intelligence." Annals of Mathematics and Artificial Intelligence 86.1-3 (2019): 193-229

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

Dátum schválenia: 14.05.2024

Abstrakt

Počet kybernetických útokov neustále narastá a ich sofistikovanosť stúpa vďaka novým technikám a stratégiám útočníkov. Organizácie musia neustále zdokonaľovať metódy detekcie a reakcie na tieto útoky, aby ochránili svoje siete a informačné systémy. Čas medzi výskytom bezpečnostného incidentu a jeho identifikáciou trvá priemerne 100 - 200 dní, s reakčným časom organizácií 50 - 70 dní. Cieľom našej práce je skrátiť tento čas, aby organizácie mohli rýchlejšie reagovať na bezpečnostné incidenty. V práci využívame teóriu grafov pri forenznej analýze v operačnom systéme Windows. Hlavným cieľom práce je identifikácia digitálnych stôp a vzťahov medzi nimi. Pre tento účel pracujeme s dátovými sadami z rôznych súťaží typu Capture the flag (CTF). Popisujeme fázy spracovania digitálnych stôp, ich transformáciu do grafov a následne identifikujeme anomálie a cykly v grafoch s cieľom poskytnúť čitateľom hlbší pohľad na túto problematiku.

Kľúčové slová: *grafy, grafové algoritmy, forenzná analýza, artefakt, kybernetická bezpečnosť*

Abstract

The number of cyber-attacks is constantly growing and their sophistication is increasing due to new techniques and strategies of attackers. Organisations need to continuously improve their methods of detecting and responding to these attacks to protect their networks and information systems. The time between the occurrence of a security incident and its identification takes an average of 100 - 200 days, with organisations having a response time of between 50 - 70 days. The goal of our work is to reduce this time so that organizations can respond to security incidents more quickly. In this work, we use graph theory for forensic analysis in Windows operating system. The main objective of the work is to identify digital traces and the relationships between them. For this purpose, we work with datasets from various Capture the flag (CTF) competitions. We describe the processing stages of the digital traces, their transformation into graphs, and then identify anomalies and cycles in the graphs in order to provide readers with a deeper insight.

Keywords: *graphs, graph algorithms, forensic analysis, artifact, cybersecurity*

Obsah

Úvod **10**

1 Základné pojmy **12**

1.1 Forenzná veda 12

1.2 Digitálne stopy 13

1.3 Forenzný artefakt 13

1.3.1 Artefakty súborového systému 14

1.3.2 Záznamy v operačnom systéme 15

1.3.3 Register operačného systému 17

1.4 Teória grafov a základné pojmy 18

1.4.1 Nezávislosť a bipartitnosť grafu 19

1.4.2 Súvislosť grafu 19

1.4.3 Excentricita, centrum, periféria 19

2 Prehľad súčasného stavu **20**

2.1 Analýza bezpečnostných dát 20

2.2 Forenzná analýza v sieťovej komunikácii s využitím teórie grafov 24

3 Dátové sady a predspracovanie dát **26**

3.1 Získavanie a popis dát 26

3.2 Predspracovanie dát 27

4 Analýza vzťahov medzi digitálnymi stopami **31**

4.1 Vytváranie grafov - supertimeline 31

4.1.1 Prípad - Szechuan Sauce 31

4.1.2 Prípad - NIST Hacking Case 32

4.1.3 Prípad - NIST Data Leakage Case 34

4.1.4 Prípady - Magnet CTF 34
4.2 Vytváranie grafov - EVT 35
4.2.1 Prípad - Szechuan Sauce 36
4.2.2 Prípad NIST Data Leakage Case 36
4.2.3 Prípady Magnet CTF 37
4.3 Vytváranie grafov - FILE 37

5 Analýza vzťahov v časových osiach pomocou vlastností grafov 39
5.1 Bipartitnosť 39
5.2 Súvislosť 40
5.3 Cykly v grafe 41
5.4 Excentricita 42

6 Hľadanie vzťahov v časových osiach 45

Záver 49

Prílohy 55
A Popis atribútov tabuľky EVT 56
B Popis atribútov tabuľky FILE 59
C Zdrojový kód 62

Zoznam obrázkov

1.1 Windows Time Rules [10] 15
1.2 Ukážka grafu 18

4.3 Graf z datasetu prípadu Szechuan Sauce DC - supertimeline s atribútmi MACB a host 32
4.4 Graf z datasetu prípadu Szechuan Sauce DC - supertimeline s atribútmi MACB a user 33
4.5 Graf z datasetu prípadu NIST Hacking Case supertimeline s atribútmi MACB a host 33
4.6 Graf z datasetu prípadu Magnet CTF 2019 supertimeline s atribútmi MACB a host 34
4.7 Graf z datasetu prípadu Szechuan Sauce EVT s atribútmi inode a user_sid 35
4.8 Graf z datasetu prípadu NIST Data Leakage Case EVT s atribútmi computer_name a user_sid 36
4.9 Graf z datasetu prípadu Magnet CTF 2022 EVT s atribútmi inode a user_sid 37
4.10 Graf z datasetu prípadu Szechuan Sauce - FILE s atribútmi MACB a filename 38

5.11 Graf z datasetu prípadu Szechuan Sauce EVT s atribútmi inode a event_id 40
5.12 Graf z datasetu prípadu Szechuan Sauce FILE s atribútmi MACB a inode 41
5.13 Cykly v grafe na obrázku 5.12 41
5.14 Cykly v grafe na obrázku 4.10 42
5.15 Graf z datasetu prípadu NIST Data Leakage Case EVT s atribútmi computer_id a event_id 43

5.16 Cykly v grafe na obrázku 4.9 43
5.17 Minimálna excentricita v grafe na obrázku 4.9 43
5.18 Excentricity v grafe na obrázku 5.12 44

6.19 Graf z datasetu prípadu Szechuan Sauce DC - supertimeline s atribútmi MACB a type 46
6.20 Graf z datasetu prípadu Szechuan Sauce DC - supertimeline s atribútmi MACB a source 47
6.21 Graf z datasetu prípadu NIST Hacking case s atribútmi user a type 48

Zoznam tabuliek

2.1 Porovnanie podobných prác 23
2.2 Porovnanie podobných prác - sieťová forenzná analýza 25

3.1 Popis atribútov tabuľky podľa [39] 27
3.2 Popis atribútov tabuľky EVT 29
3.3 Popis atribútov tabuľky FILE 30

Úvod

V digitálnom svete je bezpečnosť dát a sietí kľúčovým záujmom pre organizácie všetkých veľkostí a odvetví. S nárastom kybernetických útokov a ich stále sa meniacou povahou sa organizácie musia neustále prispôsobovať, aby ochránili svoje aktíva a zabezpečili bezpečnosť svojich informácií. Kybernetickí útočníci sa neustále posúvajú vpred a prichádzajú s novými spôsobmi, ako preniknúť do systémov a získať neoprávnený prístup k citlivým údajom. Tieto útoky sa stávajú sofistikovanejšími, preto je pre organizácie výzvou identifikovať útoky čo najrýchlejšie a zareagovať na ne adekvátne a ideálne.

Jedným z hlavných problémov je čas medzi výskytom bezpečnostného incidentu, jeho identifikáciou a následnou reakciou. Tento čas môže byť netriviálny, často sa meria v stovkách dní, čo dáva útočníkom dostatok času na spôsobenie škôd a uniknutie bez toho, aby boli odhalení. Okrem toho, aj po identifikácii incidentu organizácia potrebuje určitý čas na jeho vyriešenie a obnovenie bežného fungovania.

Naším cieľom je zredukovať tento časový interval a umožniť organizáciám rýchlejšiu a efektívnejšiu reakciu na kybernetické útoky. V práci sa zameriavame na využitie teórie grafov v procese forenznej analýzy v operačnom systéme Windows. Analýza grafov nám umožňuje identifikovať vzťahy medzi rôznymi digitálnymi stopami a ich atribútmi, čím lepšie porozumieme povahe útoku a jeho šíreniu v sieti.

Pracujeme s rôznymi dátovými sadami, ktoré boli vytvorené na účely rôznych súťaží typu Capture the Flag (CTF). Tieto dáta obsahujú rôzne typy útokov a digitálnych stôp, ktoré nám umožňujú simulovať reálne kybernetické incidenty a testovať naše metódy analýzy.

Výsledkom našej práce je identifikácia anomálií, cyklov, centra a periférie v grafoch digitálnych stôp. Získané poznatky môžu poskytnúť organizáciám komplexnejší a hlbší pohľad na ich bezpečnostné postupy a pomôcť im lepšie sa chrániť pred kybernetickými hrozbami.

V kapitolách sme postupne rozobrali základné pojmy, podobné práce, našu im-

plementáciu a výsledky. Kapitola 1 vymedzuje v prvej podkapitole základné pojmy forenznej vedy a súborový systém a v druhej teóriu grafov, kde vysvetľujeme základné pojmy potrebné na prácu s grafmi, nezávislosť, bipartitnosť, súvislosť grafu a excentricitu. V ďalšej kapitole sa venujeme podobným prácam v oblasti kybernetickej bezpečnosti, strojového učenia a grafom. Kapitola 3 popisuje prácu s dátami, ich predspracovanie a vysvetľuje, čo dáta predstavujú. Kapitola 4 opisuje proces tvorenia grafu, teda výber atribútov a ich spájanie. V kapitole 5 sa venujeme interpretácii grafov pomocou ich vlastností. V ostatnej kapitole 6 hľadáme a analyzujeme vzťahy medzi digitálnymi stopami pomocou grafov.

Kapitola 1

Základné pojmy

1.1 Forenzná veda

Forenzná veda je zodpovedanie právnych otázok aplikáciou rôznych druhov vedných disciplín. Zaoberá sa získavaním stôp, ktoré môžu slúžiť ako dôkazy v súdnom konaní (v trestnom, či občianskom). Forenzní analytici, ktorí pracujú v tejto oblasti, zhromažďujú, uchovávajú a analyzujú vedecké stopy počas vyšetrovania. Niektorí forenzní analytici pracujú na mieste činu, získavajú stopy, zatiaľ čo iní skúmajú v laboratóriách získané forenzné artefakty [1].

Digitálna forenzná analýza (ďalej len forenzná analýza) je odvetvie forenznej vedy, ktoré sa zameriava na identifikáciu, získavanie, spracovanie, analýzu a podávanie správ o údajoch uložených v elektronickej podobe. Elektronické dôkazy sú súčasťou takmer každej trestnej činnosti a digitálna forenzná podpora je pre vyšetrovanie orgánov činných v trestnom konaní veľmi dôležitá. Hlavným cieľom digitálnej forenznej analýzy je získať údaje z elektronických dôkazov, spracovať ich na použiteľné informácie a predložiť výsledky orgánom činným v trestnom konaní. Všetky procesy využívajú spoľahlivé forenzné techniky, aby sa zabezpečila prípustnosť výsledkov na súde [2].

V rámci forenznej analýzy sa vyšetrovatelia snažia získať digitálne stopy z digitálnych a kybernetických/fyzických zariadení vrátane sieťových zariadení, počítačov, inteligentných a mobilných senzorov a zariadení, ako aj dronov a robotov [3].

1.2 Digitálne stopy

Informačné a komunikačné technológie sa stali nástrojom zločinu, no vďaka pokroku vo forenznej analýze digitálnych stôp sa zároveň používajú na boj proti trestnej činnosti.

Digitálne stopy sú informácie uložené v binárnej podobe, ktoré sú relevantné pre súdne konanie [4]. Zahŕňajú počítačové dôkazy, digitálne zvukové záznamy, digitálne video, mobilné telefóny, digitálne faxy atď. [5]. Čoraz viac organizácií sa stretáva s nutnosťou zhromažďovať digitálne stopy vo svojich sieťach v reakcii na incidenty, ako sú počítačové prieniky, podvody, krádeže duševného vlastníctva, sexuálne obťažovanie a dokonca aj násilné trestné činy. [6]

Orgány činné v trestnom konaní (polícia, prokuratúra) investujú nemalé prostriedky do infraštruktúry na zber a analýzu digitálnych stôp, aby bojovali proti elektronickej kriminalite. Napriek tomu táto snaha prináša výzvy, ako je neustále vzdelávanie príslušníkov v oblasti získavania digitálnych dôkazov a udržiavanie kroku s rýchlym vývojom technológií, ako sú operačné systémy počítačov [4].

1.3 Forenzný artefakt

Forenzné artefakty sú predmety, ktoré majú forenznú hodnotu. To znamená všetky predmety, ktoré obsahujú údaje alebo stopy o niečom, čo sa stalo počas bezpečnostného incidentu. Napríklad v operačnom systéme Windows to sú záznamy (logy), register operačného systému, artefakty súborového systému (napr. žurnál), kôš, webové prehliadače, a pod. Napríklad kôš systému Windows obsahuje niekoľko skvelých artefaktov na forenznú analýzu. Máme súbory "$I" a súbory "$R", ktoré obsahujú metadáta a obsah odstránených súborov.

Webové prehliadače uchovávajú široké spektrum informácií, vrátane súborov cookie, histórie prehliadania, údajov o webových stránkach v medzipamäti a stiahnutých súborov. Tieto informácie sú cenným zdrojom počas forenzného vyšetrovania [7].

Podobne máme k dispozícii zo súborového systému ďalšie z mnohých forenzných artefaktov a môžeme s nimi pracovať v rámci forenznej analýzy.

1.3.1 Artefakty súborového systému

Operačný systém Windows využíva NewTechnologyFileSystem(NTFS) súborový systém na efektívne organizovanie, ukladanie a hľadanie súborov v pevnom disku. V súborovom systéme NTFS je každý objekt považovaný za súbor, pričom dedí všetky vlastnosti súboru. To zahŕňa aj metadáta súborového systému, ktoré definujú jeho štruktúru. Systém NTFS organizuje tieto štruktúry na zväzkoch disku v štyroch logických blokoch (NFTS Boot Sector, MFT, File System Data, MFT Copy).

Hlavnou funkciou systému NTFS je **MFT (Master File Table)**, ktorá je implementovaná ako pole záznamov. Každý súbor a adresár má aspoň jeden záznam v MFT, pričom MFT sama má tiež svoj záznam. Každý záznam v MFT sa nazýva záznam o súbore a má predvolenú pevnú veľkosť 1024 B. Prvých 42 bajtov záznamu obsahuje hlavičku MFT a zvyšok sa využíva na uloženie atribútov. Atribút je malá dátová štruktúra s konkrétnym účelom, ako napríklad $STANDARD_INFORMATION, $FILE_NAME alebo $DATA. V každom zázname súboru sa nachádzajú dva štandardné atribúty, $STANDARD_INFORMATION a $FILE_NAME. Okrem toho môže mať súbor ďalšie atribúty, ktoré sú špecifické pre jeho typ, napríklad $DATA pre dátové súbory a $INDEX_ROOT pre adresárové súbory. Je dôležité poznamenať, že súbory môžu mať aj ďalšie atribúty, ktoré sú systémom nepoužívané [8].

Štyri časové pečiatky v $STANDARD_INFORMATION a $FILE_NAME sa nazývajú MACB a poskytujú časovú hodnotu pre štyri špecifické udalosti [9]:

- M (modified) - čas poslednej úpravy obsahu súboru,
- A (accessed) - čas posledného prístupu k obsahu súboru,
- C (changed) - čas zmeny v MFT tabuľke (posledná zmena metadát súboru),
- B (birth) - čas vytvorenia súboru.

Na obrázku 1.1 sú zobrazené niektoré kombinácie a pravidlá týkajúce sa priradzovania časových pečiatok k jednotlivým záznamom v MFT tabuľke. Napríklad, ak modifikujeme súbor, tak v atribúte $STANDARD_INFORMATION sa zapíše časová pečiatka M.C. (Modified (M) - time of data modification, Metadata (C) - time of metadata modification) a v atribúte $FILE_NAME sa časová pečiatka súboru nezmení.

Obr. 1.1: Windows Time Rules [10]

1.3.2 Záznamy v operačnom systéme

Proces zaznamenávania udalostí v prostredí Windows je implementovaný ako systémová služba. V operačnom systéme Windows je služba proces, ktorý beží na pozadí a je riadený správcom riadenia služieb (SCM) a vykonáva potrebné činnosti. Služba protokolovania udalostí je zodpovedná za riadenie všetkých prístupov k protokolom udalostí systému Windows. Aplikácie a ovládače nesmú priamo čítať a zapisovať do protokolov udalostí. Musia požiadať SCM, aby služba protokolovania udalostí pre nich načítala a zapísala údaje o udalostiach a vykonala v ich mene základné operácie údržby protokolov (napríklad zálohovanie, vymazanie atď.). Existencia jediného prístupového bodu k protokolom udalostí zabezpečuje, že všetky operácie s protokolmi sa budú vždy vykonávať správne, prostredníctvom jediného jednotného API, ktoré vyžadujú všetky procesy na prístup k informáciám v protokoloch, a používatelia s nedostatočnými oprávneniami nebudú mať prístup k údajom v protokoloch [11].

V operačnom systéme sa odohráva veľké množstvo udalostí, z ktorých väčšinu nie je možné zaznamenať. Na tento účel existuje efektívny mechanizmus nazývaný zaznamenávanie udalostí (Event Logging) [12]. Tento štandardizovaný a centralizovaný systém slúži na zaznamenávanie dôležitých informácií, ktoré pochádzajú z nainštalovaného softvéru, hardvéru alebo činností používateľa.

Udalosť (event) je akákoľvek činnosť, ktorá sa stala v systéme alebo aplikácii [13]. Tieto udalosti zhromažďuje a ukladá služba zaznamenávania udalostí (Event Logging Service), ktorá umožňuje uchovávanie udalostí z rôznych zdrojov. Záznamy udalostí poskytujú chronologické informácie, ktoré pomáhajú identifikovať problémy s používaním a bezpečnosťou operačného systému, monitorujú aktivity používateľov a využitie systémových prostriedkov. Konkrétne údaje, ktoré sa zaznamenajú, však závisia od nastavení operačného systému a konfigurácie aplikácií [14].

Počas vývoja operačného systému Windows sa zároveň zdokonaľoval aj systém zaznamenávania udalostí. Prvýkrát bol predstavený v systéme Windows 3.1. Každá nová verzia Windows priniesla menšie zmeny, ale názvy súborov a cesty k záznamom udalostí zostali nemenné až do verzie Windows 2003. Počiatočné verzie používali binárny formát .evt, ktorý nebol vhodný na hľadanie reťazcov alebo prehľadávanie informácií bez špeciálneho softvéru. Tieto protokoly mali tiež obmedzenú veľkosť, čo znamenalo, že nové udalosti mohli prepísať staršie dáta. Pred systémom Windows Vista boli záznamy udalostí uložené v adresári:

`%win_dir%\System32\config`

Od operačných systémov Vista a Server 2008 došlo k výrazným zmenám v štruktúre, typoch a umiestnení záznamov udalostí v súborovom systéme. Nová verzia bola navrhnutá aj na riešenie niektorých problémov s výkonom. Bol zavedený nový formát protokolov udalostí .evtx. Najzásadnejšie zmeny nastali v štruktúre samotných záznamov udalostí. Zmenilo sa aj ich umiestnenie na:

`%win_dir%\System32\winevt\Logs`

Štruktúra záznamu udalosti [15]:

- ID záznamu udalosti,
- čas a dátum záznamu udalosti,
- kategória záznamu udalosti (napr. Application, Security, System a pod.),
- zdroj záznamu udalosti (napr. Outlook, Power, Storage, Audio, Dhcp-client a pod.),
- kategória záznamu udalosti (napr. Warning, Critical a pod.),
- používateľ,
- detailný popis udalosti atď.

1.3.3 Register operačného systému

Register operačného systému Windows je mimoriadne cenným zdrojom forenzných artefaktov pre každého analytika. Ide o centrálnu hierarchickú databázu, ktorá sa používa na ukladanie informácií potrebných na konfiguráciu systému pre jedného alebo viacerých používateľov, aplikácie a hardvérové zariadenia [16]. Register obsahuje údaje, na ktoré operačný systém Windows pravidelne odkazuje počas svojej prevádzky, vrátane [17]:

- konfigurácie systému,
- informácií o hardvérových zariadeniach,
- používateľských účtov,
- osobných nastavení a nastavení webového prehliadača,
- aktivity pri prehliadaní webových stránok,
- otvorených súborov,
- informácií o spustených programoch,
- a ďalších.

Register operačného systému Windows je binárna dátová štruktúra, ktorá nahradila konfiguračné a inicializačné súbory (.ini) používané v operačnom systéme DOS a Windows 3.x. Tieto inicializačné súbory boli roztrieštené, neštandardizované a ich prístup bol pomerne pomalý [18]. Vývoj od inicializačných súborov k registrom zahŕňal používanie konfiguračných súborov, ako napríklad System.dat a User.dat v operačnom systéme Windows 9x a Windows NT, a Classed.dat v operačnom systéme Windows ME.

Register operačného systému Windows pozostáva z piatich koreňových kľúčov (najvyššej úrovne). Z nich sú dva hlavné a tri sú odvodené, čo znamená, že sú prepojené s kľúčmi v rámci hlavných kľúčov. Týchto päť koreňových kľúčov zahŕňa [15]:

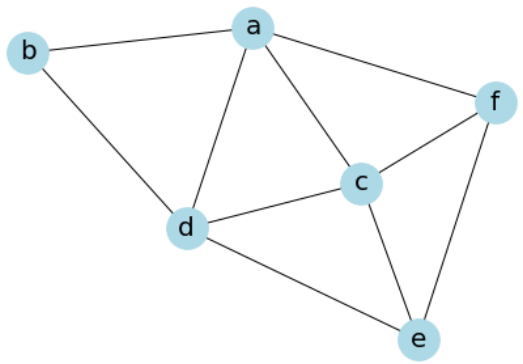
- HKEY_CLASSES_ROOT (HKCR),
- HKEY_LOCAL_MACHINE (HLKM),

- HKEY_USERS,
- HKEY_CURRENT_USER (HKCU),
- HKEY_CURRENT_CONFIG.

1.4 Teória grafov a základné pojmy

Teória grafov je časť diskrétnej matematiky, ktorá sa zaoberá štúdiom grafov. Graf pozostáva z množiny vrcholov a množiny hrán, z ktorých každá spája dva vrcholy. Obvykle je objekt reprezentovaný vrcholom a vzťah medzi dvoma objektmi je reprezentovaný hranou. Graf sa teda môže použiť na reprezentáciu akýchkoľvek informácií, ktoré možno modelovať ako objekty a vzťahy medzi týmito objektmi.

Graf G je dvojica (V, E), ktorá pozostáva z konečnej množiny V vrcholov a konečnej množiny E hrán; každá hrana je neusporiadaná dvojica vrcholov. Dva vrcholy spojené s hranou e sa nazývajú koncové vrcholy hrany e. Hranu medzi dvoma vrcholmi u a v často označujeme (u, v). Množinu vrcholov grafu G označujeme $V(G)$ a množinu hrán grafu G označujeme $E(G)$. Nech $e = (u, v)$ je hrana grafu G, potom hovoríme, že dva vrcholy u a v sú v grafe G susedné a hrana e je incidentná s vrcholmi u a v. Vrchol u sa tiež nazýva sused v v grafe G a naopak. Graf na obr. 1.2 má šesť vrcholov a, b, c, d, e, f a desať hrán. Vrcholy a a b sú koncové vrcholy hrany (a, b). Vrcholy a a b sú teda susedné. Vrcholy b, c a f sú susedmi vrcholu a.



Obr. 1.2: Ukážka grafu

Stupeň vrcholu v v grafe G, označovaný $deg(v)$ alebo $d(v)$, je počet hrán incidentných s v v G. Maximálny stupeň grafu G, označený ako $\Delta(G)$, je maximálna hodnota spomedzi stupňov všetkých vrcholov grafu G. Analogicky označujeme aj minimálny stupeň grafu G [19].

Sledom v grafe G označujeme alternujúcu postupnosť vrcholov a hrán. Sled, v ktorom sú všetky hrany navzájom rôzne, nazývame ťahom. Ťah, v ktorom sú rôzne aj všetky vrcholy, pomenúvame **cesta**. Inak povedané, cesta v grafe je postupnosť vrcholov, pre ktorú platí, že v grafe existuje hrana z vrcholu A do vrcholu B. Žiadne dva vrcholy (a teda ani hrany) sa neopakujú. Ťah, v ktorom sú všetky vrcholy okrem prvého a posledného rôzne, nazývame **cyklom**.

1.4.1 Nezávislosť a bipartitnosť grafu

Nech $G = (V, E)$ je graf. Podmnožina vrcholov $V' \subseteq V$ sa nazýva nezávislá množina v grafe G, ak pre každú dvojicu vrcholov $u, v \in V$ neexistuje v grafe G hrana spájajúca dva vrcholy u a v. Graf G sa nazýva bipartitný graf, ak množinu vrcholov V grafu G možno rozdeliť na dve nesúvislé neprázdne množiny V_1 a V_2, ktoré sú nezávislé. Dve množiny V_1 a V_2 sa často nazývajú partitné množiny grafu G. Graf nazývame **bipartitný**, práve vtedy, keď jeho vrcholovú množinu vieme rozdeliť na dve množiny V_1, V_2 tak, že každá hrana grafu má konce v dvoch rôznych množinách. Ak v grafe G existujú cykly, tak sú iba párnej dĺžky.

1.4.2 Súvislosť grafu

Graf nazývame **súvislý**, ak medzi ľubovoľnými dvomi jeho vrcholmi existuje cesta, ktorá ich spája. Maximálne súvislé podgrafy grafu nazývame komponenty. Hranu grafu nazývame **most**, ak sa po jej odobratí zvýši počet **komponentov**. **Vrchol grafu** nazývame **artikulácia**, ak sa po jeho odobratí zvýši počet komponentov grafu.

1.4.3 Excentricita, centrum, periféria

Excentricita $ecc(v)$ bodu v je vzdialenosť bodu v od najvzdialenejšieho bodu v grafe G. Polomer grafu $rad(G)$ je minimálna excentricita bodu v grafe G a priemer $diam(G)$ je maximálna excentricita bodu v grafe G. Vrchol c v G sa nazýva centrálny, ak $ecc(v) = rad(G)$. Centrum $C(G)$ je množina všetkých centrálnych vrcholov v grafe G. **Excentrický vrchol** vrcholu v je vrchol, ktorý je od neho najvzdialenejší [20]. Vrchol v sa nazýva **periférny**, ak $ecc(v) = diam(G)$. **Periféria** $Per(G)$ grafu G je množina všetkých perifférnych vrcholov v grafe G.

Kapitola 2

Prehľad súčasného stavu

Pod pojmom analýza dát a strojové učenie si často predstavíme prvky umelej inteligencie. Nemusí to byť vždy o analýze údajov týmto spôsobom. Dáta potrebujeme aj vizualizovať, predspracovať, získať základné štatistiky a pod. Práve pri vizualizácii nám dokážu pomôcť grafy. V oblasti analýzy forenzných artefaktov pomocou grafov už bolo realizovaných niekoľko prác. Mnohé z nich sa zameriavali na sieťovú komunikáciu, čo ale nie je forma dát, akú používame v tejto práci. Napriek tomu sa nimi môžeme inšpirovať, pretože ponúkajú pohľad na reprezentáciu forenzných artefaktov pomocou grafov a prepojenia medzi nimi. Vybrali sme niekoľko z týchto prác, aby sme získali základný prehľad o súčasnom stave v danej oblasti.

2.1 Analýza bezpečnostných dát

V posledných rokoch sa väčšina údajov, ako sú knihy, videá, obrázky, lekárske a genetické informácie, presúva do digitálnych formátov. Hlavným zdrojom tejto digitálnej transformácie sú laptopy, tablety, smartfóny a nositeľné zariadenia, ktoré sa stávajú súčasťou nášho každodenného života. S týmto prechodom sa stávame ľahkým cieľom pre rôzne typy kybernetických zločinov. Digitálne forenzné vyšetrovanie poskytuje spôsob, ako obnoviť stratené alebo úmyselne zmazané súbory z podozrivých zariadení. Súčasné ľudské a vládne zdroje nie sú dostatočné na vyšetrovanie kyberzločinov. Existujúce postupy digitálneho vyšetrovania vyžadujú veľkú interakciu s ľuďmi, čo spomaľuje proces vyšetrovania vzhľadom na rýchlosť, akou sa kyberzločiny páchajú. Strojové učenie, odvetvie umelej inteligencie, používa explicitné programovanie na napodobňovanie ľudského správania. Kombinácia strojového

učenia s automatizáciou v procese digitálneho vyšetrovania môže výrazne pomôcť digitálnym vyšetrovateľom [21].

Autori Constantini, Gasperis, Olivieri v článku [22] sa vo svojom výskume zaoberali možnosťou využitia umelej inteligencie a nástrojov výpočtovej logiky, konkrétne prístupu programovania množiny odpovedí, na automatizáciu analýzy dôkazov. Ukázali, ako sa zložité vyšetrovania, ktoré sú pre ľudských expertov ťažko riešiteľné, dajú vyjadriť ako optimalizačné problémy a ako tieto problémy možno vyriešiť automatizáciou analýzy dôkazov. Cieľom fázy "Analýza dôkazov" v rámci digitálnej forenznej analýzy a digitálneho vyšetrovania je poskytnúť objektívne údaje a spracovať ich tak, aby sa pomohlo pri vytváraní hypotéz v súdnom procese. Existuje mnoho prístupov a veľké množstvo výskumov na identifikáciu rôznych foriem vplyvu (neutrálneho alebo podvodného) medzi členmi siete pomocou algoritmov založených na teórii grafov.

Ako uvádza Ch. Easttom v článku [23], teória grafov je dôležitou časťou diskrétnej matematiky, ktorá skúma vzťahy medzi objektmi, nazývanými vrcholy. Táto teória sa často využíva na analýzu rôznych vzťahov, a jej aplikácia v trestnom vyšetrovaní nie je výnimkou. Easttom opisuje aplikáciu teórie grafov v súvislosti s vyšetrovaním trestných činov. Prostredníctvom matematického modelovania udalostí umožňuje lepšie porozumenie dátam a vzťahom medzi zúčastnenými entitami, ako sú podozriví, obete a informačné systémy.

I. Palmer, R. Campbell a B. Gelfand vysvetľujú vo svojom článku [24], že grafy a grafová teória dokážu vysvetľovať vzťahy a môžeme ich použiť vo forenznej analýze. V dôsledku rozšírenia technológií v spoločnosti sa možnosti digitálneho forenzného skúmania zmenšujú. Súčasná situácia prináša množstvo výziev pre proces vyšetrovania. Napriek existencii mnohých nástrojov na riešenie týchto výziev, väčšina z nich je zameraná na identifikáciu konkrétnych dôkazov namiesto podpory samotného vyšetrovacieho procesu. V danom článku demonštrujú aplikáciu grafov a ich implementáciu. Tvrdia, že teória grafov môže slúžiť ako základ pre analýzu dát generovaných nástrojmi digitálnej forenznej analýzy, teda grafické zobrazenie umožňuje vyšetrovateľom nielen vizualizovať, ale aj vykonávať analýzu údajov. Efektívne dokážu lokalizovať dôležité informácie. Vo svojej práci preskúmali potenciálne výhody použitia grafického zobrazenia vo forenznej analýze a získali vysokoúrovňový pohľad na systém bez potreby rozsiahlych znalostí operačného systému a jeho aplikácií.

V článku Centrality and Scalability Analysis on Distributed Graph of Large-Scale E-mail Dataset for Digital Forensics [25] sa autori zamerali na e-maily, pretože dokážu byť dôležitým zdrojom dôkazov, ale ich prehľadávanie môže byť časovo náročné. Zaoberajú sa distribuovanou analýzou grafov pre digitálne forenzné údaje a skúmajú presnosť algoritmov centrality. Experimenty vykonávali na súbore e-mailov, aby porovnali časy behu medzi rôznymi algoritmami a tradičnými prístupmi. Ich výsledky naznačujú, že distribuovaná analýza grafov môže zlepšiť efektivitu a presnosť analýzy digitálnych stôp.

Aj keď autori článku J. Binwal, R. Devi a B. Singh sa vo svojom článku [26] venovali odtlačkom prstom, ich reprezentácii v grafoch a vytvoreniu dvoch algoritmov: 1. nájdenie grafickej reprezentácie odtlačku prsta a 2. určenie, či sú grafy akýchkoľvek dvoch vzorov odtlačkov prstov izomorfné alebo nie. Vstupné dáta pre nás nie sú rovnaké, aj keď sa týkajú forenznej analýzy. Napriek tomu môžeme využiť poznatky získané z tohto článku.

Uvádzame aj ďalší prehľad článku [27], kde sú grafy využívané na forenznú analýzu, ktorá obsahuje aj antiforenzné techniky. Grafy útokov sa používajú na zistenie potenciálnych ciest útokov z konfigurácie systému a známych zraniteľností. Pomáhajú eliminovať známe zraniteľné body, čím bránia útokom, a poskytujú forenzným expertom podrobný prehľad o možných spôsoboch útoku. Po útoku sa forenzná analýza využíva na lepšie pochopenie scenára útoku, pričom prepojenie dôkazov s útokmi hrá kľúčovú úlohu. S narastajúcim používaním antiforenzných nástrojov sa stáva forenzná analýza náročnejšou. Navrhujeme využiť graf útoku na efektívnu forenznú analýzu, ktorá zahŕňa aj antiforenzné možnosti. To umožní vysvetliť chýbajúce dôkazy prostredníctvom komplexnejších ciest útoku, ktoré skryjú potenciálne stopy útoku. Autori ukazujú túto schopnosť na príkladovej štúdii útoku na databázu.

Našim prínosom bude kombinácia týchto prístupov a aplikácia konceptov na zlepšenie procesu digitálnej forenznej analýzy a vyšetrovania. Naša práca sa sústreďuje na automatizáciu analýzy digitálnych stôp využitím teórie grafov na lepšie porozumenie vzťahom medzi digitálnymi stopami a entitami v digitálnom prostredí. V tabuľke 2.1 uvádzame stručný prehľad vyššie uvedených článkov, metódy a dáta v nich použité.

Popis	Metódy	Dáta
[21] - automatizácia a urýchlenie analýzy dôkazov v rámci forenzného vyšetrovania	umelá inteligencia, strojové učenie	NA
[22] - automatizácia analýzy dôkazov v rámci forenzného vyšetrovania	umelá inteligencia, strojové učenie, teória grafov	akademický problém "escaped zebra", "monkey and banana", ...
[23] - aplikácia teórie grafov na matematické modelovanie forenzného vyšetrovania	vysvetlenie vzťahov medzi entitami vo forenznom vyšetrovaní	vyšetrovanie trestných činov
[24] - komplexná analýza digitálneho forenzného procesu	grafická vizualizácia, vysvetlenie vzťahov	dáta z pamäťovej foreznej analýzy
[25] - analýza grafov pre digitálne forenzné údaje	centralita	e-maily
[26] - analýza odtlačkov prstov s využitím grafov	izomorfizmus grafov	odtlačky prstov
[27] - využitie grafov na forenznú analýzu a elimináciu antiforenzných techník	graf útokov	útok na databázu

Tabuľka 2.1: Porovnanie podobných prác

2.2 Forenzná analýza v sieťovej komunikácii s využitím teórie grafov

Napriek tomu, že sa v našej práci venujeme dátam zo súborového systému NTFS operačného systému Windows, zhrnieme aj niekoľko prác z oblasti digitálnej forenznej analýzy v sieťovej komunikácii. Tieto dáta vyzerajú inak, ale rozhodne vieme nájsť prepojenie medzi dátami súborového systému a dátami v sieťovej komunikácii. V oboch prípadoch ide o dáta, ktoré využívame na digitálnu forenznú analýzu, preto sa v tejto podkapitole budeme venovať článkom zameraným na forenznú analýzu v sieťovej komunikácii.

Stále aktuálna výzva zisťovania a zmierňovania porúch v počítačových sieťach sa stala dôležitejšou ako kedykoľvek predtým, najmä v súvislosti s obrovským počtom inteligentných zariadení, ktoré sa každý deň pripájajú k najrôznejším sieťam. Bez ohľadu na to, či je hlavnou príčinou danej anomálie narušenie bezpečnosti, zlyhanie súčiastky, faktor prostredia alebo dokonca akákoľvek kombinácia týchto dôvodov, anomálie je potrebné včas a správne odhaliť a zmierniť. V článku [28] sa autori venujú prehľadu a vyhodnoteniu najnovších štúdií týkajúcich sa problematiky detekcie anomálií v počítačových sieťach. Uvádzajú podrobný opis problému detekcie anomálií a znázorňujú rôzne kategorizácie jeho riešení. Ilustrujú tiež niektoré najnovšie riešenia na úrovni siete a zobrazujú súčasné trendy v riešení anomálií spôsobených malvérom v sieťach smartfónov. Okrem toho hodnotia prezentované riešenia a upozorňujú na ich nedostatky.

V článku [29] autori predstavujú nový prístup k forenznej analýze sietí založený na grafoch. Ich centrálnou myšlienkou je model grafu digitálnych stôp, ktorý umožňuje prezentáciu dôkazov a automatizované úvahy. Navrhujú hierarchický rámec zdôvodňovania s cieľom odvodiť funkčné stavy sieťových entít z lokálnych pozorovaní a identifikovať dôležité entity zo štruktúry grafu. Ďalej prezentujú rámec pre interaktívne testovanie hypotéz, ktorý pomáha identifikovať útočné aktivity zo sekundárnych dôkazov. Autorský prototyp implementuje tieto techniky, a experimentálne výsledky ukazujú ich schopnosť extrahovať útoky a scenáre s menšou závislosťou od pevne zakódovaných expertných znalostí.

S rastúcim objemom a komplexitou internetovej prevádzky sa zvyšuje aj počet počítačových trestných činov. V reakcii na tento trend vznikla oblasť počítačovej a sieťovej forenznej analýzy. Hlavným cieľom sieťovej forenznej analýzy je iden-

tifikácia škodlivých aktivít prostredníctvom zhromažďovania a analýzy dôkazov, napríklad pri hackerských útokoch. Napriek tomu, kvôli veľkému množstvu internetovej prevádzky nie sú všetky dáta relevantné pre vyšetrovanie. Na základe analýzy existujúcich metód sieťovej forenznej analýzy autori článku [30] navrhli novú metódu, ktorá využíva kombináciu sieťovej zraniteľnosti a grafu sieťových dôkazov. Ich metóda efektívne rekonštruuje scenáre útokov a identifikuje viacstupňové útoky prostredníctvom zdôvodňovania na základe dôkazov. Experimentálne výsledky potvrdzujú úplnosť a dôveryhodnosť autormi navrhovanej metódy.

Pochopenie šírenia malvéru v počítačových sieťach je kľúčové. S rastúcim počtom prepojených zariadení sa zvyšuje riziko infekcie viacerých zariadení. Potrebujeme nové metódy obrany. Využitie teórie spektrálnych grafov pomáha pochopiť šírenie malvéru v sieťových systémoch. Článok [31] analyzuje toto šírenie s reálnymi dátami a novinkou je využitie rôznych Laplaciánových matíc na sledovanie zmien v sieťových modeloch. Zistenia autorov môžu viesť k lepšiemu pochopeniu situácie a rýchlejšiemu získaniu informácií o infekciách.

Popis	**Metódy**	**Dáta**
[28] - detekcia anomálií vo forenznej analýze	metódy na hľadanie anomálií	dáta z forenznej analýzy
[29] - forenzná analýza sietí	model grafu digitálnych stôp	dáta z forenznej analýzy sietí
[30] - identifikácia škodlivých aktivít prostredníctvom zhromažďovania a analýzy dôkazov	návrh metódy - kombinácia sieťovej zraniteľnosti a grafu sieťových dôkazov	NA
[31] - pochopiť šírenie malvéru v sieťových systémoch	teória spektrálnych grafov	reálne dáta sieťových modelov

Tabuľka 2.2: Porovnanie podobných prác - sieťová forenzná analýza

Kapitola 3

Dátové sady a predspracovanie dát

V tejto kapitole sme sa venovali získaniu, predspracovaniu dát a ich vysvetleniu, aby sme ich v neskoršom kroku vedeli spájať do grafov, filtrovať a analyzovať. Popísali sme vytvorenie supertimeline a následnú transofromáciu na ďalšie dva datasety.

3.1 Získavanie a popis dát

Vybrali sme si sedem fiktívnych prípadov zo súťaží CTF zameraných na forenznú analýzu, reakciu na incidenty a vyhľadávanie hrozieb. Z týchto prípadov sme používali obrazy diskov.

Prvým príkladom je prípad ukradnutej sečuánskej omáčky z portálu DFIR Madness s názvom **Case001 - The case of the Stolen Szechuan sauce** [32], kde v tomto prípade hlavným cieľom bolo zistiť, ako sa recept spoločnosti CITADEL dostal na temnú webovú stránku. Spoločnosť požiadala o forenznú analýzu a identifikáciu nežiaducich aplikácií, ktoré boli nainštalované v systéme a zistenie miesta a času inštalácie. Prípad ponúka aj informácie, či bol vytvorený, upravený, zmazaný nejaký obsah, alebo došlo k úniku údajov. Pracovali sme s artefaktami z riadiaceho servera domény DC (ďalej ako "DC server") spoločnosti.

Ďalšie tri prípady **Magnet CTF 2019** [33], **Magnet CTF 2020** [34] a **Magnet CTF 2022** [35] pochádzali zo súťaže CTF (Capture the Flag) Magnet Forensics. Nešlo o klasickú forenznú analýzu, ale skôr o odpovede na otázky typu: "kedy sme získali obraz disku" alebo "kedy bol softvér nainštalovaný".

Posledné dva prípady **NIST Data Leakage Case** [36] a **NIST Hacking Case** [37] slúžia na osvojenie si rôznych foriem úniku dát a zdokonalenie techník

ich vyšetrovania. Zamerali sme sa na vyšetrovanie prípadu úniku dát, kde je kľúčové odhaliť dôkazy o nezákonných aktivitách a získať všetky informácie vytvorené podozrivým.

3.2 Predspracovanie dát

V rámci predspracovania dát sme postupovali podľa rovnakých krokov vo všetkých siedmich prípadoch, konkrétne sme špecifikovali proces predspracovania iba pre jeden prípad. Pracovali sme s dátami v súlade s postupom popísaným v článku "Detection of relevant digital evidence in the forensic timelines" [38].

Atribút	**Popis**
Date	dátum, kedy nastala udalosť
Time	čas, kedy nastala udalosť
Timezone	časová zóna
MACB	časové pečiatky (Modification, Access, Creation, Birth)
Source	skratka názvu zdroja (napr. REG - záznamy z registra)
Sourcetype	popis zdroja
Type	typ časovej značky (napr. posledný zápis)
User	meno používateľa (ak existuje), ktoré je spojené s udalosťou
Host	názov hostiteľa (ak existuje), ktorý je priradený k udalosti
Short	obsahuje pole s krátkym popisom, v ktorom je uložený text
Desc	pole, ktoré obsahuje väčšinu analyzovaných informácií
Version	číslo verzie časovej pečiatky
Filename	názov súboru, ktorý je spojený s udalosťou
Inode	číslo inodu analyzovaného súboru
Notes	miesto na ukladanie dodatočných informácií
Format	vstupný modul, ktorý bol použitý na analyzovanie
Extra	pole s parsovanými informáciami, ktoré sú tu spojené a uložené

Tabuľka 3.1: Popis atribútov tabuľky podľa [39]

Na vytvorenie časovej osi sme použili nástroj Log2timeline [40] a jeho doplnky. Výslednú časovú os sme ďalej upravili nástrojom psort.py a jazykom Python s knižnicou pandas. Náš dataset obsahoval záznamy z 11 rôznych zdrojov údajov, pričom najvýznamnejšie zastúpené boli záznamy z FILE, EVT a REG, ktoré tvorili 87%

všetkých záznamov. Extrahované atribúty sme rozdelili do siedmich kategórií.

Dataset sme zúžili na čas bezpečnostného incidentu a manuálne identifikovali relevantné digitálne stopy, vrátane inodov súborov: 84630, 84880, 84987, 86966, 86967, 86968, 86970, 86971, 86975, 87059, 87060, 87064, 87111, 87112, 8713 a súborov s názvami: 'coreupdater.exe', 'FILESH 1', 'Secret', 'BETH_S 1.TXT', 'Beth-_Secret.lnk', 'SECRET 1.TXT', 'SECRET_beth.lnk', 'Szechuan', 'SZECHU-1.TXT', 'Secret.lnk', 'NoJerry.lnk', 'No-Jerry.txt', 'f01b4d95cf55d32a.automatic-Destinationsms', 'SECRET _beth.txt', 'Beth_Secret.txt', 'Secret.zip', 'coreupdater.exe.2424urv.partial'.

Ďalej sme vykonali analýzu inodov a názvov súborov, pričom sme vylúčili niektoré inody a súbory. Na záver sme použili agregačné funkcie a vytvorili kombinácie atribútov pre analýzu údajov.

Manuálnu identifikáciu inodov súborov a názvov súborov sme vykonali iba v prípade receptu ukradnutej sečuánskej omáčky, pre ostatné datasety identifikované inody a ani názvy súborov neboli zatiaľ identifikované.

Každý zo siedmich datasetov je supertimeline - časová os, ktorá obsahuje 17 atribútov, riadky v nej sú záznamy (udalosti). V tabuľke 3.1 je ich zoznam a popis. Okrem základných 7 datasetov sme vytvorili 6 ďalších csv súborov, ktoré obsahujú záznamy z FILE a 7 csv súborov obsahujúcich záznamy z EVT. Datasety z FILE majú 44 atribútov a z EVT 40 atribútov. Tieto súbory boli vytvorené extrahovaním údajov z pôvodnej supertimeline-y. Napríklad, ak bol v pôvodnom datasete stĺpec MACB, tak v novom (EVT alebo FILE) sa vytvorili štyri nové stĺpce, pôvodný sa vymazal. Nové stĺpce sú 'M', 'A', 'C', 'B'. Ak sa v pôvodných dátach nachádzal záznam '.ACB', tak v príslušných stĺpcoch 'A', 'C', 'B' sme zapísali 1 a do stĺpca 'M' sme zapísali 0. Týmto spôsobom nám z časti vznikli binárne dáta resp. stĺpce. Nie všetky atribúty sa dali takto konvertovať, preto stĺpce ako 'date' alebo 'time' ostali nezmenené.

Z datasetu supertimeline sme vytvorili ďalšie datasety, evt.csv a file.csv. Vytvorili sme ich pre každý prípad okrem datasetu NIST Hacking Case pre evt záznamy. Tieto dva artefakty sme si vybrali na základe toho, že práve záznamy z EVT a FILE tvorilo 87% zo všetkých záznamov. Sedem kategoriálnych premenných zo supertimeline sme transformovali do binárnej podoby a niektoré atribúty z pôvodnej supertimeline sme vynechali [38]. V záznamoch z EVT takto vzniklo 40 atribútov popísaných v prílohe v tabuľke A.1 a názorne v tabuľke 3.2 a pri záznamoch z FILE vzniklo 44 atribútov popísaných v prílohe v tabuľke B.1 a názorne v tabuľke 3.3.

Atribút	**Popis atribútu**
inode	pôvodný stĺpec inode, identifikácia Inode v rámci NTFS súborového systému
datetime	časová pečiatka, spojené dátumové a časové stĺpce, časové pásmo UTC
computer_name	extrahované z stĺpca desc alebo extra
source_name	extrahované z stĺpca desc alebo extra - zdroj logu (napr. Service Control Manager)
filename_type	
filename_security	extrahované z názvu súboru, logovací stĺpec v Security.evtx
...	...
recovered	extrahované z extra stĺpca, True, ak bol záznam obnovený (obnovené dáta = zmazané dáta)
user_sid	extrahované z extra stĺpca, ID používateľa, ktorý vytvoril proces
execution_process_id	extrahované z extra stĺpca, ID procesu, ktorý sa vykonáva
channel	extrahované z extra stĺpca, napr. System
event_id	špecifický typ udalosti
audit_account_logon	rozdelené na základe skupín evt - do 9 skupín, určených pomocou Event ID
keywords	
keywords_audit_failure	AuditFailure = 0x10000000000000L, pripojené ku všetkým neúspešným bezpečnostným udalostiam auditu. Toto kľúčové slovo používajte iba pre udalosti v bezpečnostnom zázname.
timestamp	modifikovaná časová pečiatka (dátum a čas)
hour	hodina (extrakcia z časovej pečiatky)
minute	minúta (extrakcia z časovej pečiatky)

Tabuľka 3.2: Popis atribútov tabuľky EVT

Atribút	**Popis atribútu**
filename	názov súboru
inode	pôvodný stĺpec inode, označenie Inode v rámci súborového systému NTFS
extra	stĺpec s informáciami - z tohto stĺpca boli extrahované rôzne atribúty
datetime	časová pečiatka, spojenie stĺpcov dátumu a času, časové pásmo UTC
M	modifikované
A	prístup
C	zmenené, modifikované MFT
B	vytvorené, čas vytvorenia súboru
file_stat	stĺpec na základe hodnôt zo stĺpca sourcetype
...	...
NTFS_USN_change	stĺpec na základe hodnôt zo stĺpca sourcetype
file_path	súbor s celou cestou
name	názov stĺpca extrahovaný z desc
typef	typ súboru ('None', 'súbor', 'adresár', 'odkaz')
directory	1, ak je to priečinok, 0, ak nie
mft	stĺpec vytvorený na základe hodnôt z formátovacieho stĺpca
file_size	veľkosť súboru
hour	hodina (extrakcia z časovej pečiatky)
minute	minúta (extrakcia z časovej pečiatky)

Tabuľka 3.3: Popis atribútov tabuľky FILE

Kapitola 4

Analýza vzťahov medzi digitálnymi stopami

V tejto časti práce sme pracovali s dátami z csv súborov supertimeline, súborového systému (FILE) a logov (EVT) pomocou grafov. Samostatne v podkapitolách sme sa venovali vytváraniu grafov z rôznych zdrojov - supertimeline, EVT a FILE. Pracovali sme v jupyter notebookoch a používali Python knižnicu networkx, ktorá dokáže jednoducho vytvoriť z pandas DataFrame-u graf ako prepojenie atribútov, ktoré mu určíme. Pri práci s touto knižnicou sme si všimli, že pri vytváraní grafu pomocou príkazu 'nx.from_pandas_edgelist()' dokáže spracovať maximálne dva atribúty, čiže dva stĺpce z DataFrame. Preto sme vždy vybrali dvojicu atribútov, ktorá môže spolu potenciálne súvisieť a hľadali vzťahy medzi týmito atribútmi. V prílohe C sme uviedli zdrojové kódy, ktoré sme použili pri práci s grafmi.

4.1 Vytváranie grafov - supertimeline

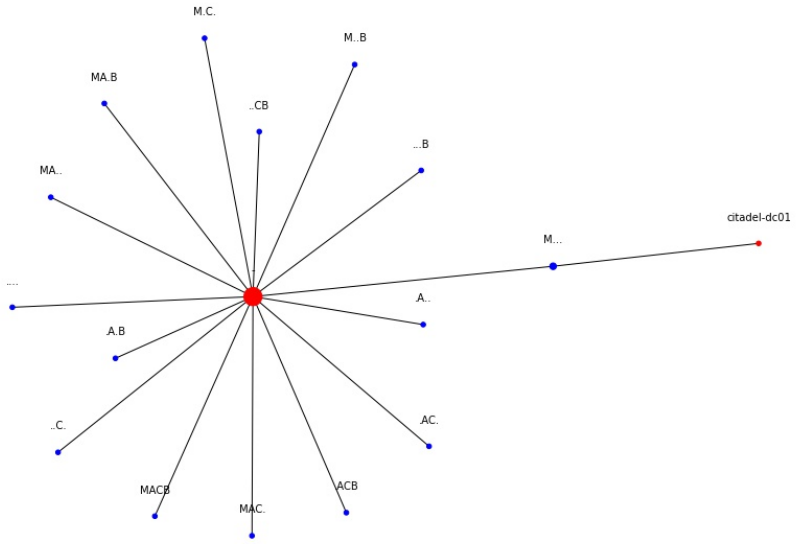
Prvým príkladom hľadania vzťahov v digitálnych stopách je použitie supertimelines. Časové osi (supertimeline-y) obsahujú 17 atribútov, bližšie popísaných v kapitole 3.2 v tabuľke 3.1.

4.1.1 Prípad - Szechuan Sauce

Stĺpce, ktoré sme si vybrali v prípade ukradnutej sečuánskej omáčky a datasetu DC sú: 'MACB', 'source', 'sourcetype', 'type', 'user', 'host', 'inode', 'format'. Nevyužili sme ich všetky, ale aspoň pre zúženie datasetu nám to pomohlo. Dataset ukradnutej sečuánskej omáčky z DC má 1 263 787 záznamov. Predpokladali sme, že

časová pečiatka 'MACB' a 'host' budú spolu súvisieť. V ďalšom kroku sme sa snažili nájsť prepojenia medzi jednotlivými atribútmi, kombinovali sme ich a vykresľovali grafy pomocou knižnice networkx. Vytvorili sme graf na obrázku 4.3, ktorý vznikol spojením týchto dvoch atribútov. Atribút 'host' má v tomto datasete dve unikátne hodnoty: '-'(neznámy) a 'citadel-dc01'.

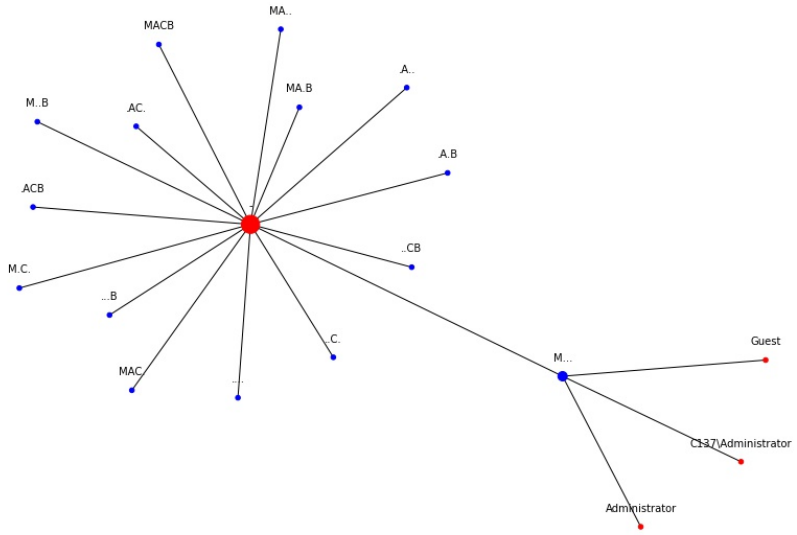
Po vykreslení tohto grafu môžeme vidieť, že každá časová pečiatka je spojená s '-', ale len jedna ('M...') sa spája s 'citadel-dc01'. Ešte nevieme, či je toto správanie normálne, alebo sa za touto anomáliou niečo skrýva, preto sme sa to rozhodli preskúmať v ďalších kapitolách. Najskôr sme si ale vytvorili ďalšie grafy, ktoré sú tiež tvorené časovou pečiatkou 'MACB' a atribútmi 'user' (obr. 4.4) a 'type' (obr. 6.19). Vytvorili sme aj graf so 'source' (obr. 6.20), ale ten spočiatku veľa nenaznačil.



Obr. 4.3: Graf z datasetu prípadu Szechuan Sauce DC - supertimeline s atribútmi MACB a host

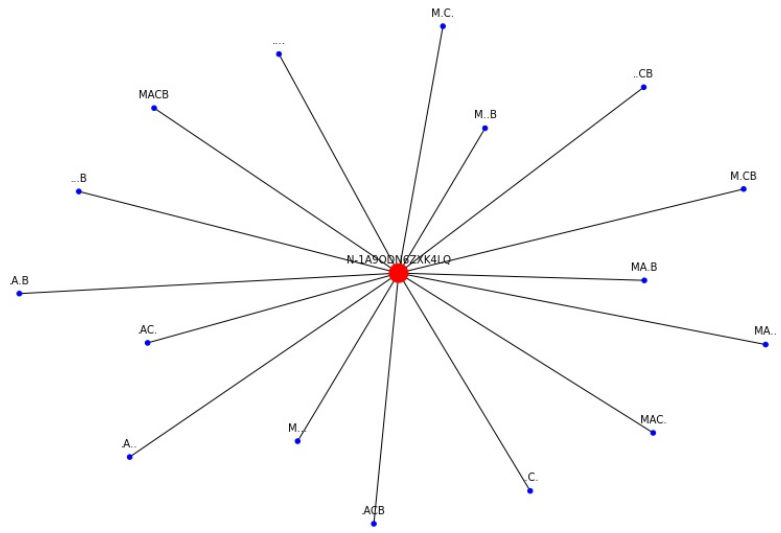
4.1.2 Prípad - NIST Hacking Case

Podobne ako v prípade ukradnutej sečuánskej omáčky, sme aj v ostatných prípadoch začali rovnakým spôsobom. Dáta sme načítali, názvy atribútov sa nezmenili, ani ich počet. Jediná zmena nastala v počte záznamov. V niektorých prípadoch ich bolo oveľa viac, v iných zas menej. V tejto podkapitole sme sa zamerali na prípad NIST Hacking Case. Dataset obsahoval 216 910 záznamov. Rovnako ako v predchádzajúcom prípade sme vytvorili grafy. Začali sme grafom, ktorý vznikol spojením



Obr. 4.4: Graf z datasetu prípadu Szechuan Sauce DC - supertimeline s atribútmi MACB a user

atribútov MACB a host na obr. 4.5. Ďalej sme postupovali analogicky ako v prípade ukradnutej sečuánskej omáčky, vytvárali sme podobné grafy, čiže s rovnakými atribútmi, len v inom datasete.



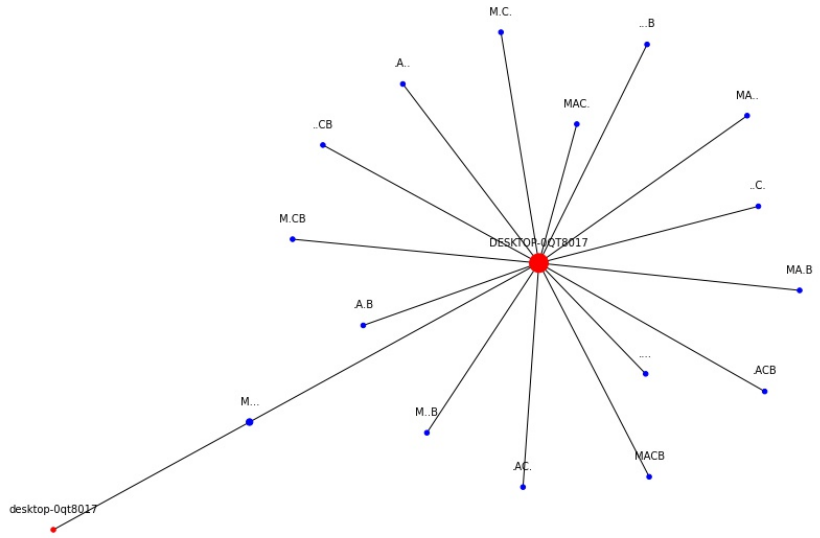
Obr. 4.5: Graf z datasetu prípadu NIST Hacking Case supertimeline s atribútmi MACB a host

4.1.3 Prípad - NIST Data Leakage Case

Pokračovali sme s datasetom NIST Data Leakage Case, kde sa v datasete supertimeline sa nachádzalo 3 203 758. To nám zatiaľ neprekážala, pretože atribúty, ktoré sme spájali v rámci supertimeline, mali málo unikátnych hodnôt, čiže boli prehľadné. Grafy sme vygenerovali a ďalej analyzovali v kapitole 6.

4.1.4 Prípady - Magnet CTF

V prípadoch Magnet CTF 2019, Magnet CTF 2019, Magnet CTF 2019 sme vytvárali rovnaké grafy ako v prechádzajúcich prípadoch, teda spájali sme atribúty MACB, host, user, source a type. Uviedli sme iba niektoré z nich (napr. obr. 4.6), pretože podrobnejšie sme ich analyzovali v kapitole 6. Grafy týchto atribútov a ďalších dvoch prípadov - Magnet CTF 2020 a Magnet CTF 2022 vyzerajú podobne, to znamená, že majú jeden vrchol atribútu host, ktorý je spojený so vrcholmi časových pečiatok MACB a jeden vrchol atribútu host je spojený s iba jedným vrcholom časovej MACB a to "M...". Dataset supertimeline prípadu Magnet CTF 2019 má 2 687 232 záznamov, dataset supertimeline prípadu Magnet CTF 2020 má 9 335 096 záznamov a z prípadu Magnet CTF 2022 sme mali k dispozícii 13 837 814.



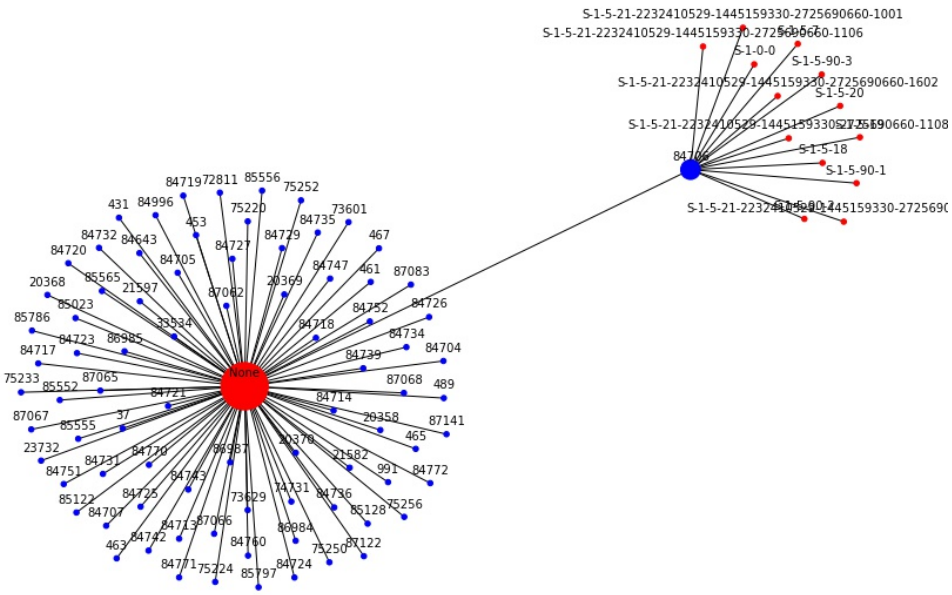
Obr. 4.6: Graf z datasetu prípadu Magnet CTF 2019 supertimeline s atribútmi MACB a host

4.2 Vytváranie grafov - EVT

Otázkou ostáva, na základe čoho sme vyberali atribúty spomedzi mnohých stĺpcov v DataFrame. Uvažujme napríklad súbor DC_evt (Prípad ukradnutej sečuánskej omáčky), ktorý má 40 atribútov, uvádzame iba niektoré: 'inode', 'id', 'datetime', 'computer_name', 'source_name', 'filename_type', ... , 'user_sid', 'execution_process_id', 'channel', 'event_id', 'audit_account_logon', ... , 'audit_system', 'task', 'keywords', 'keywords_audit_failure', ... , 'keywords_wdi_diagnostic', 'timestamp', 'epochtime', 'hour' a 'minute'.

Pre každý stĺpec sme určili počet unikátnych hodnôt, ktoré daný atribút obsahuje. Bolo zrejmé, že stĺpec 'id' bude mať toľko unikátnych hodnôt, aký je počet riadkov v datasete.

Čím je hodnota unikátnych hodnôt nižšia, zobrazenie v grafe je jednoduchšie. To však neznamená, že nebolo možné použiť aj iné atribúty, ale pre názornosť to stačí, pretože tieto atribúty vedia poskytnúť veľa informácií relevantných pre prípad.



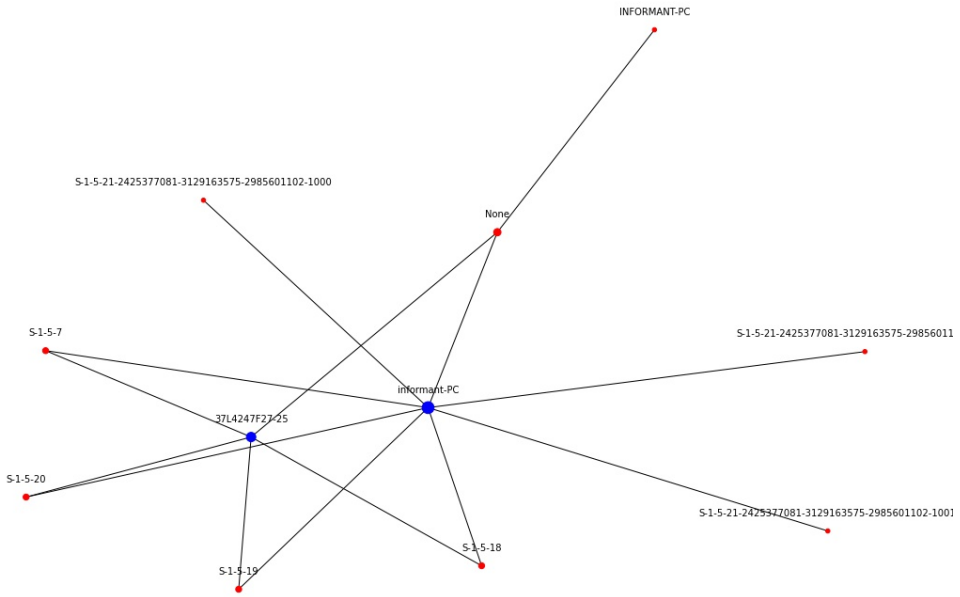
Obr. 4.7: Graf z datasetu prípadu Szechuan Sauce EVT s atribútmi inode a user_sid

4.2.1 Prípad - Szechuan Sauce

Na obrázku 4.7 vidíme grafické prepojenia medzi inode a user_sid. Počet unikátnych hodnôt atribútu inode bol 81 a počet unikátnych hodnôt user_sid bol 14. V grafe sú vrcholy predstavujúce inode zobrazené modrou farbou a vrcholy zobrazujúce user_sid červenou. Ak sa v datasete vyskytne aspoň jeden riadok, v ktorom je zároveň jeden špecifický inode a jedno user_sid, tak medzi týmito dvoma vrcholmi sa vytvorí hrana. To znamená, že medzi týmito vrcholmi existuje nejaký vzťah, teda existuje záznam spájajúci daný inode a user_sid.

4.2.2 Prípad NIST Data Leakage Case

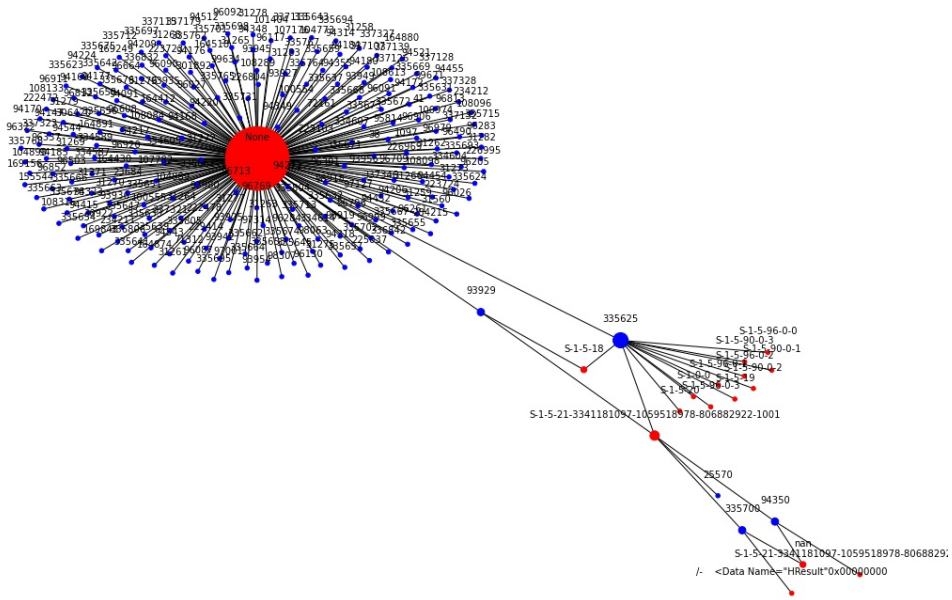
Podobne ako v prípade Szechuan Sauce sme vytvorili graf, ktorý obsahoval atribúty inode a user_sid. Tento graf vyzeral podobne ako v predchádzajúcom prípade. Ďalej sme vytvorili grafy, ktoré ako atribúty obsahovali inode a event_id, computer_name a event_id a ďalší graf computer_name a user_sid, ktorý sme ako príklad uviedli na obrázku 4.8.



Obr. 4.8: Graf z datasetu prípadu NIST Data Leakage Case EVT s atribútmi computer_name a user_sid

4.2.3 Prípady Magnet CTF

Rovnakým spôsobom sme postupovali aj v prípadoch Magnet CTF 2019, Magnet CTF 2020 a Magnet CTF 2022. Grafy, ktoré boli tvorené rovnakými atribútmi ako v predchádzajúcich prípadoch vyzerali zvyčajne podobne, preto sme ich neuviedli. Venovali sme sa im až v ďalších kapitolách, kde sme sa bližšie zaoberali ich vlastnosťami. Predstavili sme iba graf na obrázku 4.9, ktorý je odlišný od všetkých ostatných grafov v iných prípadoch, ktoré boli tvorené práve týmito atribútmi.



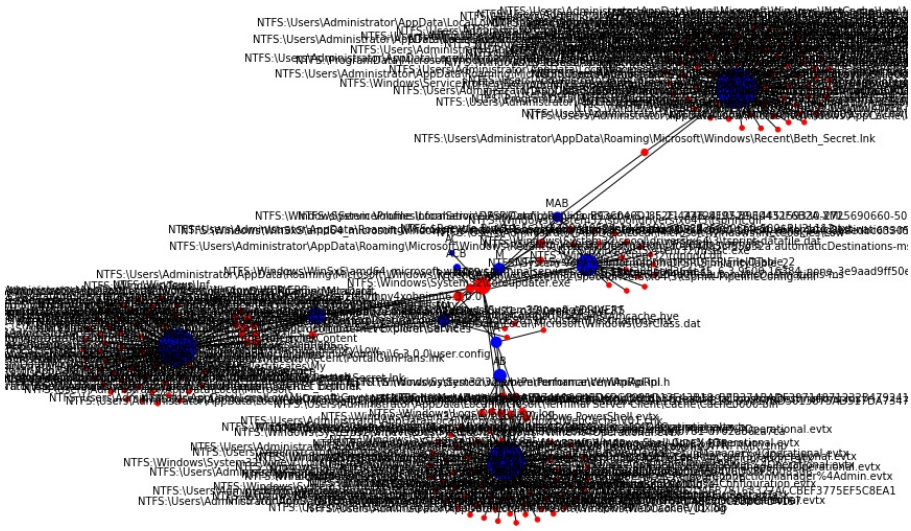
Obr. 4.9: Graf z datasetu prípadu Magnet CTF 2022 EVT s atribútmi inode a user_sid

4.3 Vytváranie grafov - FILE

Na základe grafov na obrázkoch 4.3 a 4.4 sme sa pozreli na tabuľkové dáta zo supertimeline a zistili, že všetky záznamy, ktoré vyzerali byť nezvyčajné sa udiali 19.09.2020. Preto sme pri práci s datasetom file.csv v prípade ukradnutej sečuánskej omáčky vyfiltrovali dáta iba z tohto dňa. Urobili sme tak preto, lebo počet riadkov v tomto datasete bol veľmi vysoký, a ak sme chceli vytvoriť nejaký graf, tak sa stal nečitateľným, resp. nepodarilo sa nám ho ani vytvoriť. Po filtrácii (dôkladnejšie vysvetlenej v kapitole 6) podľa dňa sme vytvorili graf spojením atribútov MACB

a inode (obr. 5.12) a MACB a filename (obr. 4.10). V oboch grafoch sú časové pečiatky označené ako modré vrcholy a inody resp. filename sú modré vrcholy. Zistili sme, že graf na obrázku 4.10 je veľmi nečitateľný. Tým sme poukázali na to, že nejde vždy prioritne o vizualizáciu, ale o vlastnosti grafu, ktoré sme popísali neskôr.

Podobne sme sa pokúšali o vytvorenie grafov aj z ďalších datasetov zo zdroja FILE, bolo to ale náročnejšie, pretože tieto datasety mali veľký počet záznamov a nemali sme na vytvorenie žiadaných grafov dostatočné výpočtové prostriedky. To neznamená, že sme nevytvorili žiaden graf, ale podrobnejšie sme sa tejto analýze nevenovali.



Obr. 4.10: Graf z datasetu prípadu Szechuan Sauce - FILE s atribútmi MACB a filename

Kapitola 5

Analýza vzťahov v časových osiach pomocou vlastností grafov

Analýzu vzťahov medzi digitálnymi stopami sme nerobili iba spôsobom, že sme sa pozreli na graf a vedeli sme z neho vyčítať všetky podrobnosti. Grafy neboli vždy prehľadné a niekedy nebolo možné identifikovať ani názov vrchola (napr. graf na obr. 4.10). Preto je dôležité sa zamerať na vlastnosti grafov, ktoré sme skúmali po ich vygenerovaní. V nasledujúcich podkapitolách sme sa zamerali na bipartitnosť, súvislosť, hľadali sme cykly v grafe a skúmali excentricity v grafoch.

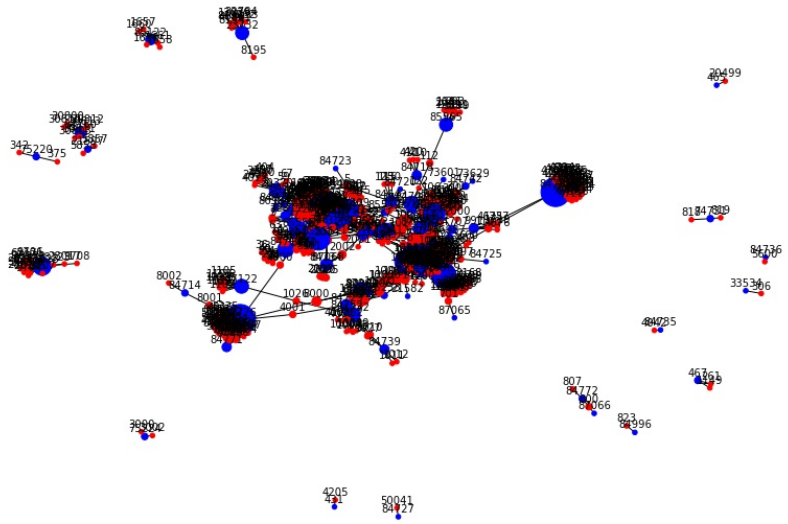
5.1 Bipartitnosť

Jednou z vlastností grafov, ktorú je možné využiť pre účely analýzy vzťahov digitálnych stôp, je bipartitnosť. Tú sme definovali v podkapitole 1.4.1. Bipartitnosť je vlastnosť grafu, kedy môžeme rozdeliť vrcholy grafu do dvoch partií tak, že žiadne dva vrcholy v rovnakej skupine nie sú priamo spojené hranou.

Ak nájdeme graf, ktorý nie je bipartitný, musí obsahovať cyklus nepárnej dĺžky. Našim cieľom nebolo zisťovať, či je graf bipartitný alebo nie, snažili sme sa nájsť cyklus. Samozrejme, že sme ho našli aj bez toho, aby sme vedeli, či graf je bipartitný alebo nie, ale bol to skôr prvý krok k výsledku, pretože algoritmy na nájdenie cyklov sú časovo náročnejšie ako na zistenie, či graf je bipartitný alebo nie. Ak nájdeme graf, ktorý nie je bipartitný, máme 100% istotu, že v grafe sa nachádza cyklus. Nájdenie cyklu v grafe znamená (fiktívne v tomto prípade), že nejaký inode je spojený s user_sid a ten s ďalším inode, ktorý je spojený s ďalším user_sid, ktorý je spojený s prvým inode a tvoria cyklus. Teda záznamy medzi týmito vrcholmi sú nejako

prepojené v datasete a môžeme si položiť otázku prečo, bližšie ich preskúmať.

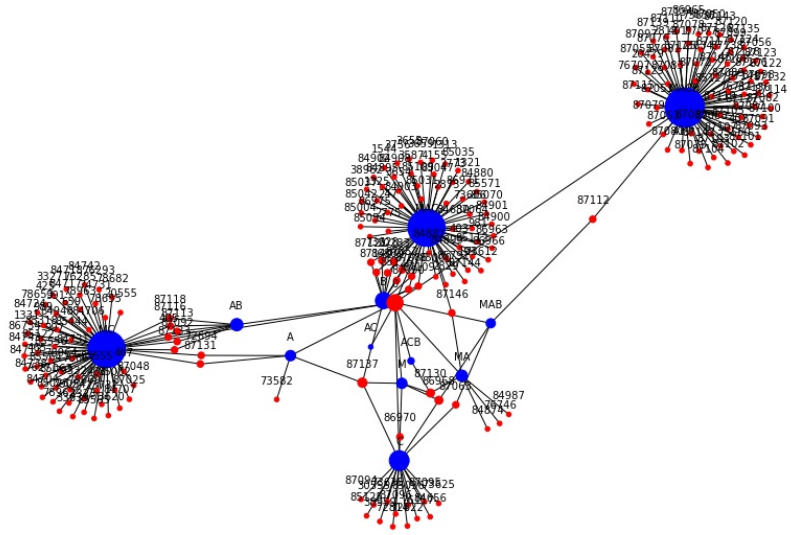
V grafe na obrázku 4.4 je bipartitnosť jednoznačná a viditeľná, ale neskôr ukážeme grafy, kde to nebude také zrejmé. Pomocou Pythonu a knižnice networkx vieme zistiť, či je graf bipartitný alebo nie. V tomto prípade teda vidíme, že graf na obrázku 4.4 je bipartitný.



Obr. 5.11: Graf z datasetu prípadu Szechuan Sauce EVT s atribútmi inode a event_id

5.2 Súvislosť

Ďalšou zaujímavou vlastnosťou grafov je súvislosť, bližšie popísaná v podkapitole 1.4.2. To, že nájdeme súvislý graf nám hovorí, že medzi atribútmi v grafe existuje prepojenie. Na obrázku 5.11 je graf reprezenujúci dáta z DC evt, presnejšie, prepojenia medzi atribútmi inode a event_id. Tento graf je aj príklad toho, že nie každý graf musí byť nutne zobrazený pekne, teda nemusí byť všetko viditeľné. Čo je ale zrejmé, graf nie je súvislý. Inak povedané, v dátach existuje prepojenie medzi niektorými inode a event_id, ale tie už ďalej nie sú spojené s ďalšími. Môžeme ich v ďalšom kroku analýzy preskúmať, prípadne tieto malé grafové zhluky vylúčiť z grafu, kým nám neostane jeden graf. S týmto samostatným, väčším grafom už potom pracujeme klasicky ako v ostatných prípadoch. Alebo naopak, pozrieť sa na malé komponenty a hľadať v nich anomálie.



Obr. 5.12: Graf z datasetu prípadu Szechuan Sauce FILE s atribútmi MACB a inode

5.3 Cykly v grafe

Po vytvorení grafu na obrázku 5.12 sme ho začali analyzovať. Už nebolo také viditeľné na čo konkrétne sa máme zamerať, preto sme museli aplikovať vlastnosti grafu. Vyskúšali sme vyhľadať bázické cykly v grafe pomocou knižnice networkx a príkazu "$nx.cycle_basis(G)$". Cykly sú vypísané na obrázku 5.13. V červených obdĺžnikoch sú zvýraznené inody, ktoré sa nachádzajú v cykloch v grafe na obrázku 5.12 a zároveň boli ručnou analýzou označené ako relevantné pre prípad.

```
Cyklus: ['C', 87063, 'MAB', 0]
Cyklus: ['MACB', 87112, 'MAB', 0]
Cyklus: ['M', 86968, 'MAB', 0]
Cyklus: ['C', 86968, 'MAB', 0]
Cyklus: ['MAC', 87146, 'MA', 0]
Cyklus: ['M', 87130, 'MA', 0]
Cyklus: ['ACB', 87130, 'MA', 0]
Cyklus: ['MC', 72694, 'A', 0]
Cyklus: ['MC', 87131, 'A', 0]
Cyklus: ['M', 87137, 'A', 0]
Cyklus: ['B', 87137, 'A', 0]
Cyklus: ['C', 87137, 'A', 0]
Cyklus: ['C', 86970, 'M', 0]
Cyklus: ['MC', 405, 'AB', 0]
Cyklus: ['MC', 87133, 'AB', 0]
Cyklus: ['MC', 87113, 'AB', 0]
Cyklus: ['MC', 87092, 'AB', 0]
Cyklus: ['MC', 87118, 'AB', 0]
Cyklus: ['MC', 87116, 'AB', 0]
Cyklus: ['MAC', 73628, 'B', 0]
Cyklus: ['MAC', 87121, 'B', 0]
Cyklus: ['MAC', 87052, 'B', 0]
Cyklus: ['MAC', 87140, 'B', 0]
Cyklus: ['MAC', 87109, 'B', 0]
Cyklus: ['MAC', 87090, 'B', 0]
Cyklus: ['MAC', 87086, 'B', 0]
Cyklus: ['MAC', 86800, 'B', 0]
Cyklus: ['MAC', 85126, 'B', 0]
Cyklus: ['MAC', 20283, 'B', 0]
```

Obr. 5.13: Cykly v grafe na obrázku 5.12

Graf na obrázku 4.10 je oveľa menej čitateľný ako predchádzajúci graf. Preto sme nemali veľa možností na analýzu voľným okom a hneď sme začali prehľadávať graf pomocou jeho vlastností, teda cyklami. Rovnakým spôsobom ako v predchádzajúcom grafe sme našli bázické cykly vypísané na obrázku. 5.14. V červených obdĺžnikoch sú zvýraznené názvy súborov, ktoré sa nachádzajú v cykloch v grafe na obrázku 4.10 a zároveň boli ručnou analýzou označené ako relevantné pre prípad.

Podobným spôsobom sme uvažovali aj v prípade grafu na obrázku 5.15, kde sme našli bázické cykly v danom grafe, ktorých je 65. V každom sa stále vykytujú dve rovnaké hodnoty z atribútu computer_name a jedna z event_id. Mení sa len ďalšia, štvrtá hodnota atribútu event_id.

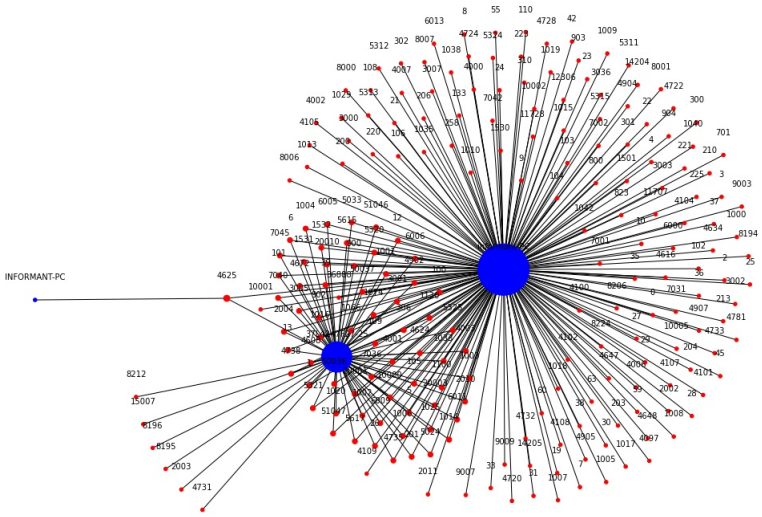
Za pozornosť stál aj graf na obrázku 4.9, ktorý bol, ako sme spomenuli už v kapitole 4.2.3, odlišný od grafov v ostatných prípadoch, ktoré boli tvorené tými istými atribútmi z dát z rovnakého zdroja (v tomto prípade EVT). Našli sme tri bázické cykly, ktoré sú vypísané na obrázku 5.16.

```
Cyklus: ['NTFS:\\$MFT', 'C', 'NTFS:\\Windows\\System32\\coreupdater.exe', 'B']
Cyklus: ['NTFS:\\Users\\Administrator\\AppData\\Roaming\\Microsoft\\Windows\\Recent\\AutomaticDestinations\\9b9cdc69c1c24e2b.au
tomaticDestinations-ms', 'MAB', 'NTFS:\\System Volume Information\\DFSR\\Config\\Replica_E936046D-8F2F-444B-8F97-B9E0432F6B2A.X
ML', 'C']
Cyklus: ['NTFS:\\$MFT', 'MAB', 'NTFS:\\System Volume Information\\DFSR\\Config\\Replica_E936046D-8F2F-444B-8F97-B9E0432F6B2A.XM
L', 'C']
Cyklus: ['NTFS:\\$MFT', 'MACB', 'NTFS:\\Users\\Administrator\\AppData\\Roaming\\Microsoft\\Windows\\Recent\\Beth_Secret.lnk',
'MAB']
Cyklus: ['M', 'NTFS:\\Users\\Administrator\\AppData\\Roaming\\Microsoft\\Windows\\Recent\\AutomaticDestinations\\f01b4d95cf55d3
2a.automaticDestinations-ms', 'C', 'NTFS:\\Windows\\System32\\coreupdater.exe']
Cyklus: ['M', 'NTFS:\\Users\\Administrator\\AppData\\Roaming\\Microsoft\\Windows\\Recent\\AutomaticDestinations\\9b9cdc69c1c24e
2b.automaticDestinations-ms', 'C', 'NTFS:\\Windows\\System32\\coreupdater.exe']
Cyklus: ['NTFS:\\$MFT', 'A', 'NTFS:\\Windows\\System32\\coreupdater.exe', 'B']
Cyklus: ['NTFS:\\$MFT', 'MC', 'NTFS:\\Windows\\System32\\wbem\\Performance\\WmiApRpl.ini', 'A']
Cyklus: ['NTFS:\\Windows\\System32\\wbem\\Performance\\WmiApRpl.h', 'MC', 'NTFS:\\Windows\\System32\\wbem\\Performance\\WmiApRp
l.ini', 'A']
Cyklus: ['NTFS:\\Users\\Administrator\\AppData\\LocalLow\\Microsoft\\CryptnetUrlCache\\MetaData\\6BADA8974A10C4BD62CC921D13E43B
18_BEB37ABADF39714871232B4792417E04', 'AB', 'NTFS:\\Windows\\Inf\\WmiApRpl\\WmiApRpl.h', 'MC']
Cyklus: ['NTFS:\\$MFT', 'AB', 'NTFS:\\Windows\\Inf\\WmiApRpl\\WmiApRpl.h', 'MC']
Cyklus: ['NTFS:\\Users\\Administrator\\AppData\\LocalLow\\Microsoft\\CryptnetUrlCache\\MetaData\\6BADA8974A10C4BD62CC921D13E43B
18_D9817BD5013875AD517DA73475345203', 'AB', 'NTFS:\\Windows\\Inf\\WmiApRpl\\WmiApRpl.h', 'MC']
Cyklus: ['NTFS:\\Users\\Administrator\\AppData\\Roaming\\Microsoft\\Windows\\Recent\\CustomDestinations\\28c8b86deab549a1.custo
mDestinations-ms', 'AB', 'NTFS:\\Windows\\Inf\\WmiApRpl\\WmiApRpl.h', 'MC']
Cyklus: ['NTFS:\\Users\\Administrator\\Documents\\Default.rdp', 'AB', 'NTFS:\\Windows\\Inf\\WmiApRpl\\WmiApRpl.h', 'MC']
Cyklus: ['NTFS:\\Users\\Administrator\\AppData\\Local\\Microsoft\\Terminal Server Client\\Cache\\Cache0000.bin', 'AB', 'NTF
S:\\Windows\\Inf\\WmiApRpl\\WmiApRpl.h', 'MC']
Cyklus: ['NTFS:\\$MFT', 'M', 'NTFS:\\Windows\\System32\\coreupdater.exe', 'B']
Cyklus: ['NTFS:\\$MFT', 'ACB', 'NTFS:\\Users\\Administrator\\AppData\\Roaming\\Microsoft\\Windows\\Recent\\AutomaticDestination
s\\1bc392b8e104a00e.automaticDestinations-ms', 'M']
Cyklus: ['NTFS:\\Users\\Administrator\\AppData\\Roaming\\Microsoft\\Windows\\Recent\\CustomDestinations\\590aee7bdd69b59b.custo
mDestinations-ms', 'MA', 'NTFS:\\Users\\Administrator\\AppData\\Roaming\\Microsoft\\Windows\\Recent\\AutomaticDestinations\\1bc
392b8e104a00e.automaticDestinations-ms', 'M', 'NTFS:\\Windows\\System32\\coreupdater.exe', 'B', 'NTFS:\\Users\\Administrator\\A
ppData\\Local\\Temp\\1', 'MAC']
Cyklus: ['NTFS:\\$MFT', 'MA', 'NTFS:\\Users\\Administrator\\AppData\\Roaming\\Microsoft\\Windows\\Recent\\AutomaticDestinations
\\1bc392b8e104a00e.automaticDestinations-ms', 'M']
```

Obr. 5.14: Cykly v grafe na obrázku 4.10

5.4 Excentricita

Zaujímavou vlastnosťou, ktorú sme skúmali v grafoch, ktoré sme vytvárali, či už z datasetov supertimeline, EVT alebo FILE, bola excentricita, bližšie popísaná v kapitole 1.4.3. Zisťovali sme minimálnu a maximálnu excentricitu v grafe.



Obr. 5.15: Graf z datasetu prípadu NIST Data Leakage Case EVT s atribútmi computer_id a event_id

```
Cyklus: [94350, 'S-1-5-21-3341181097-1059518978-806882922-1001', 335700, nan]
Cyklus: [93929, 'S-1-5-18', 335625, 'S-1-5-21-3341181097-1059518978-806882922-1001']
Cyklus: [93929, 'None', 335625, 'S-1-5-21-3341181097-1059518978-806882922-1001']
```

Obr. 5.16: Cykly v grafe na obrázku 4.9

Minimálnu excentricitu z dôvodu, aby sme zistili, ktoré vrcholy sú cetrálne, teda tvoria centrum grafu. Na druhej strane maximálnu excentricitu z dôvodu, aby sme zistili, ktoré vrcholy tvoria perifériu grafu, teda sú odľahlé v grafe. Uvedieme príklad využitia excentricity v našich dátach na prípade Magnet CTF 2022 na grafe na obrázku 4.9. V tomto grafe sme našli 3 vrcholy s najnižšou excentricitou 3 uvedené na obrázku 5.17 a 230 vrcholov s najvyššou excentricitou 5.

```
Atribut user_sid - Excentricita vrcholu S-1-5-21-3341181097-1059518978-806882922-1001: 3
Atribut inode - Excentricita vrcholu 93929: 3
Atribut inode - Excentricita vrcholu 335625: 3
```

Obr. 5.17: Minimálna excentricita v grafe na obrázku 4.9

Pomocou excentricity sme preskúmali aj graf na obrázku 5.12, aby sme zistili, čo je centrum grafu a čo periféria. Na obrázku 5.18 je vypísaný jeden vrchol s minimálnou excentricitou, ktorá príslúcha atribútu inode a má hodnotu 0, potom 12 vrcholov (časovej pečiatky MACB) s excentricitou 3 a všetky ostatné vrcholy (aj

tie, ktoré nie sú vypísané) majú excentricitu 4. Môžeme tvrdiť, že centrum grafu je 0 a periféria sú všetky vrcholy s excentricitou 4.

```
Vrchol: 0 Excentricita: 2
Vrchol: C Excentricita: 3
Vrchol: AB Excentricita: 3
Vrchol: MACB Excentricita: 3
Vrchol: MC Excentricita: 3
Vrchol: MAC Excentricita: 3
Vrchol: B Excentricita: 3
Vrchol: MA Excentricita: 3
Vrchol: MAB Excentricita: 3
Vrchol: M Excentricita: 3
Vrchol: A Excentricita: 3
Vrchol: AC Excentricita: 3
Vrchol: ACB Excentricita: 3
Vrchol: 84656 Excentricita: 4
Vrchol: 38 Excentricita: 4
Vrchol: 84666 Excentricita: 4
Vrchol: 78965 Excentricita: 4
Vrchol: 84684 Excentricita: 4
```

Obr. 5.18: Excentricity v grafe na obrázku 5.12

Kapitola 6

Hľadanie vzťahov v časových osiach

Pri hľadaní vzťahov v časových osiach (supertimeline) sme sa zamerali na atribúty, z ktorých sme vytvárali grafy, hľadali v nich prepojenia, skúmali grafy z pohľadu ich vlastností, ktoré sme už spomenuli v podkapitolách 5.1 - bipartitnosť, 5.2 - súvislosť, 5.3 - cykly a 1.4.3 - excentricita.

Po spojení 'MACB' a 'user' môžeme vidieť taký istý vzor ako pri 'MACB' a 'host', že časová pečiatka 'M...' sa spája aj s ďalšími používateľmi, konkrétne $'Administrator'$, $'Guest'$, $'C137/Administrator'$.

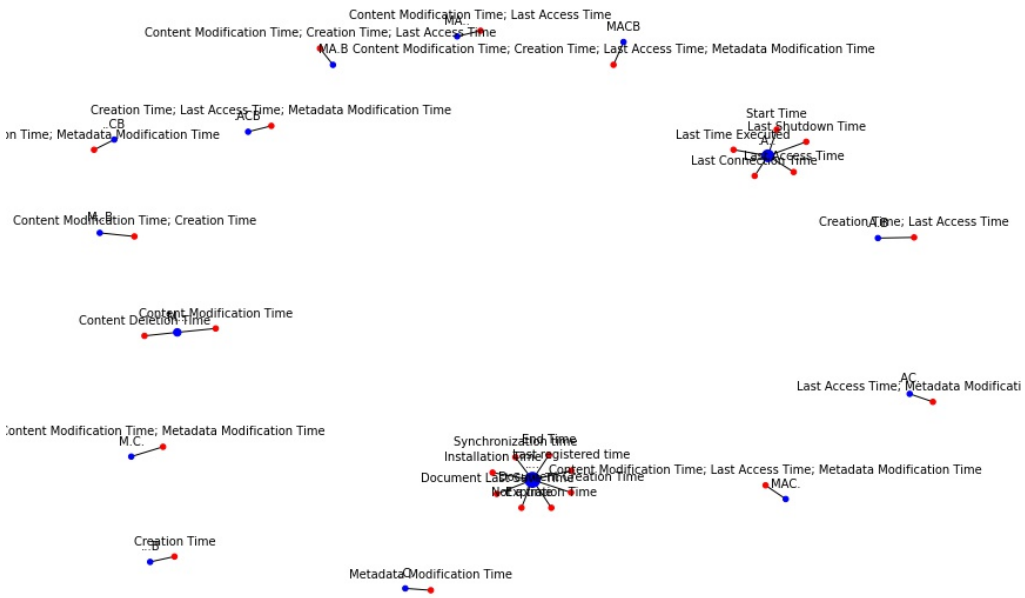
Graf vytvorený z atribútov 'MACB' a 'type' je odlišný od predchádzajúcich dvoch. Nie je súvislý a má viac menších komponentov (hviezd), ktoré sú väčšinou tvorené dvomi vrcholmi a jednou hranou, ale môžeme tu vidieť aj dve hviezdy, jednu s 5 listami, jednu s 8 listami a jednu s 2 listami. Práve jedna z týchto hviezd je pre nás podstatná, tá, ktorá má znova časovú pečiatku ('M...').

Rozhodli sme sa vyfiltrovať dataset. Najskôr sme ho zúžili na záznamy, ktoré obsahujú časovú pečiatku ('M...'). Tým sme vynechali veľké množstvo záznamov, ale stále nám ich ostalo 397 060, čo je približne tretina z pôvodných.

Tento dataset sme ďalej filtrovali na záznamy, ktoré obsahujú v atribúte 'host' hodnotu 'citadel-dc01'. Ostalo nám 6 záznamov, ktoré už bolo pomerne jednoduché preskúmať. Tu sme sa vrátili ku grafu 6.20, kde bola časová pečiatka 'M...' spojená s 'FILE', 'OLECF', 'AMCACHE', 'LNK', 'RECBIN' a 'PE'. Práve v týchto šesť záznamoch sa vyskytoval 'OLECF'. Z týchto šiestich záznamov je jeden taký, ktorý obsahuje nami identifitovaný relevantný názov súboru 'f01b4d95cf55d32a.automatic-Destinationsms'. Tiež tento záznam má aj inode, ktorý sme identifikovali '86970'

a v ostatných 5 záznamoch sa nachádza ďalší inode ’86968’.

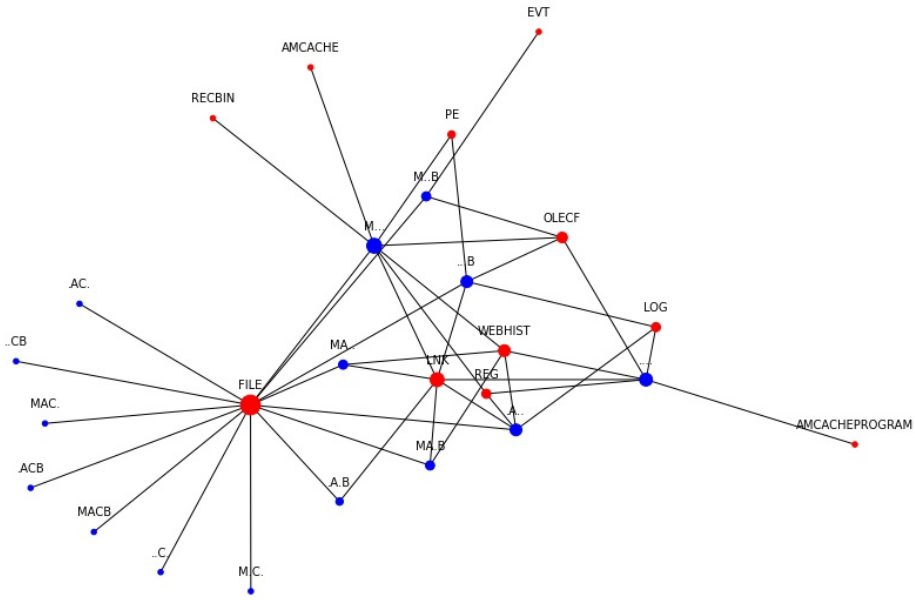
Filtrovali sme aj pomocou atribútu ’user’, teda podľa používateľov $'Guest'$, $'Administrator'$, $'C137/Administrator'$. Filtráciou podľa týchto používateľov sme dostali vždy jeden záznam, kde máme síce ’source’ - ’REG’ a ’type’ - ’Content Modification Time’, čo síce korešponduje s vytvorenými grafmi, ale v názve súboru nie je súbor, ktorý by sme označili ako relevantný pre prípad. Samozrejme, ak by sme nevedeli, ktoré súbory máme hľadať, bola by potrebná podrobnejšia analýza aj tých súborov, ktoré sa nachádzajú v záznamoch: ’SAM’, ’NTUSER.DAT’.



Obr. 6.19: Graf z datasetu prípadu Szechuan Sauce DC - supertimeline s atribútmi MACB a type

Veľmi zaujímavou bola filtrácia podľa ’type’ - ’Content Deletion Time’, pretože podľa ’Content Modification Time’ sme získali 397 059 záznamov, čo je veľmi veľa na ručné analyzovanie. Tak vznikol iba jeden záznam, ktorý má ’source’ - ’RECBIN’ a ’desc’ - ’C:/FileShare/Secret/SECRET_beth.txt’, kde sa nachádza názov súboru, ktorý sme identifikovali ako relevantný pre prípad ’SECRET_beth.txt’

Po získaní nájdených dátumov v supertimeline, ktoré by sme mohli označiť za anomálne, resp. sme mali pocit, že práve v týchto dňoch sa niečo dialo, sme sa posunuli na prehľadávanie datasetov zo zdrojov EVT a FILE. V grafe na obrázku 4.7 môžeme vidieť, že veľmi veľa inodov je pospájaných s jedným user_sid, ale toto user_sid je None. Čo je ale zaujímavé, jeden inode je viditeľne vzdialený od ostat-



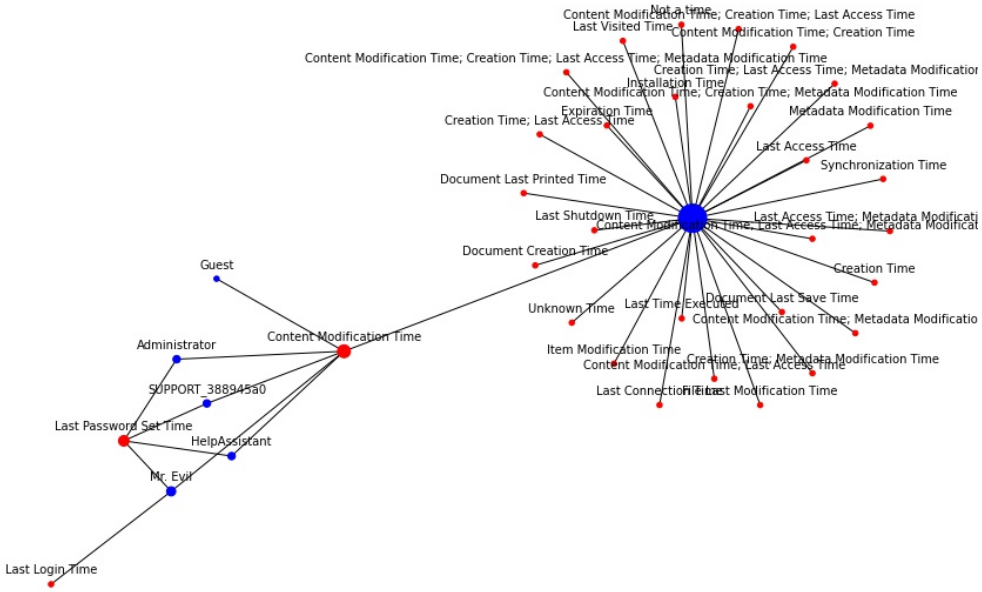
Obr. 6.20: Graf z datasetu prípadu Szechuan Sauce DC - supertimeline s atribútmi MACB a source

ných. To, že je vzdialený, neznamená, že obsahuje viac záznamov. Ide iba o grafickú reprezentáciu, knižnica networkx to tak vykresľuje, kvôli prehľadnosti. Inode 84706 je spojený s mnohými ďalšími user_sid.

Namieste sú konkrétne parciálne otázky, čo inode reprezentuje, či je daný atribút relevantný pre analýzu vzťahov, alebo prečo je spojený s takým množstvom user_sid. Práve tu sa nám otvárajú dvere pre podrobnejšiu analýzu tohto grafu a záznamov z datasetu. Vieme si vyfiltrovať záznamy, ktoré obsahujú len inode 84706 a ďalej, či súvisí nejako s ostatnými inode alebo súbormi relevantnými pre prípad. Zaznamenávame anomáliu.

V prípade NIST Hacking case sme postupovali podobným spôsobom v analýze supertimeline ako v prípade ukradnutej sečuánskej omáčky. Po vytvorení grafu na obrázku 4.5 sa nám nenaskytlo veľa otázok, preto sme museli analyzovať iné grafy ako tento, pretože tento graf je hviezda. V strede má centrálny vrchol atribútu host a ten je spojený iba s hodnotami atribútu časovej pečiatky MACB. Preto sme vytvárali ďalšie grafy, ako napríklad na obrázku 6.21, kde môžeme vidieť niekoľkých užívateľov spojených iba s niektorými časovými pečiatkami a preto sme sa rozhodli dataset vyfiltrovať podľa týchto užívateľov. Pri každom užívateľovi sme dostali na výstupe nie príliš veľké množstvo záznamov. Čo sme si ale všimli, bolo, že takmer

všetky záznamy vyfiltrovaných užívateľov boli v dátume 19.08.2004 a jeden záznam 27.08.2004. Podobne ako v predchádzajúcom prípade môžeme v ďalšom vyhľadávaní v datasete EVT zúžiť dataset na tieto dva dni, resp. na toto rozmedzie.



Obr. 6.21: Graf z datasetu prípadu NIST Hacking case s atribútmi user a type

Záver

V práci sme sa venovali rozboru digitálnej forenznej analýzy, teórii grafov a ich spojenia. Prezentovali sme metódy na vytváranie grafov a hľadanie vzťahov medzi digitálnymi stopami. K dispozícii sme mali sedem datasetov zo súťaží typu "Capture the Flag".

Našim cieľom bolo porovnať aktuálne prístupy k analýze forenzných artefaktov pomocou grafových algoritmov. Tomuto cieľu sme sa venovali v kapitole 2, kde sme popísali podobné práce v oblasti digitálnej forenznej analýzy. Súčasne sme sa venovali aj prácam zameraným na aplikáciu umelej inteligencie vo forenznej analýze a implementácii grafov a grafových algoritmov. Napokon sme skúmali využitie grafov v sieťovej forenznej analýze.

Druhým cieľom tejto bakalárskej práce bolo navrhnúť vhodnú reprezentáciu forenzných artefaktov s využitím teórie grafov. Tomuto cieľu sme sa venovali v kapitole 4. V kapitole sme popisovali predspracovanie dát a úpravu do vhodnej podoby, aby sme ich mohli reprezentovať grafmi. Problematické bolo určiť hrany, ale nakoniec sme ich určovali spôsobom, že ak sa dva vrcholy (dve hodnoty atribútu) nachádzajú v tom istom riadku (zázname) v dátach, tak sme ich spojili. Na vytváranie grafov sme používali knižnicu implementovanú v pythone, networkx. Po vytvorení grafov sme ich analyzovali a hľadali vzťahy medzi jednotlivými atribútmi v grafe, čo bolo našim ďalším cieľom.

Poslednému cieľu sme sa venovali v kapitole 5. Bolo ním navrhnúť a implementovať grafový algoritmus na analýzu vzťahov medzi forenznými artefaktami. V rámci kapitoly sme hľadali anomálne hodnoty a vzťahy medzi atribútmi. Vychádzali sme z predpokladu, že záznamy (digitálne stopy), ktoré sú odľahlé od ostatných záznamov (outlieri) môžu predstavovať nezvyčajné správanie v rámci operačného systému. To môže byť významné pre skúmaný prípad a zaujímavé pre forenzného vyšetrovateľa. Vzťahy medzi atribútmi sme hľadali a vysvetľovali pomocou rôznych vlastností grafov, akými sú napríklad bipartitnosť, súvislosť, hľadanie cyklov v grafe,

alebo excentricita. Problematické bolo, ak sme mali príliš veľa cyklov - ich prehľadávanie sa stalo časovo náročné, alebo ak sme mali iba dve hodnoty excentricít, ťažko bolo určiť na čo sa máme zamerať. Výhodou je to, že využívanie týchto vlastností niekoľkonásobne znižuje čas prehľadávania dát.

Nami analyzované metódy sme odskúšali na dátach vytvorených z "Capture the Flag" súťaží. Takto sme ukázali, že dáta z bezpečnostných incidentov vieme analyzovať aj pomocou grafov. To nám pomohlo nájsť vzťahy medzi atribútmi a interpretovať ich. Našim cieľom nebolo úplne odbremeniť bezpečnostného analytika od jeho práce pri riešení bezpečnostného incidentu, ale pomôcť mu zamerať sa na údaje, ktoré vedli k zisteniu, čo sa dialo počas bezpečnostného incidentu. V prípade ukradnutej sečuánskej omáčky sme mali ohodnotené dáta, vedeli sme, ktoré inody a názvy súborov sú relevantné pre prípad, preto sme si vedeli zverifikovať, či nami implementované metódy spĺňajú požadovaný účel. V ostatných prípadoch sa zatiaľ môžeme iba domnievať, ale môžeme podať bezpečnostnému analytikovi poznatky, ktoré sme nadobudli pri výskume v tejto práci, a to by mu mohlo uľahčiť jeho prácu pri určovaní relevantných záznamov pre prípad.

Vzhľadom na rozsah skúmanej problematiky, táto práca nedokáže poňať celú problematiku, čo umožňuje ďalší výskum a rozšírenie o nové dáta a prípady. Môžeme navrhnúť nový spôsob reprezentácie grafu v Pythone pomocou slovníka, aby sme mohli spájať vrcholy s viacerými atribútmi. Ďalším krokom by bolo preskúmať reprezentáciu cez maticu susednosti a vypočítať jej vlastné číslo, čo by mohlo byť veľmi informatívne. Ak budeme mať dostatočné výpočtové prostriedky, chceme rozšíriť kapitolu 4.3, kde sme nevytvorili všetky grafy kvôli ich nedostatku.

Zoznam použitej literatúry

[1] Pavol Sokol, Ladislav Bačo, Tomáš Bajtoš, Digitálna forenzná analýza I [online], [cit. 2024-02-25]. Dostupné z: https://unibook.upjs.sk/img/cms/2020/pf/digitalna-forenzna-analyza-1.pdf. ISBN 978-80-8152-916-0

[2] Digital forensics [online], [cit. 2024-02-25]. Dostupné na: https://www.interpol.int/How-we-work/Innovation/Digital-forensics

[3] Jean-Paul A. Yaacoub, Hassan N. Noura, Ola Salman, Ali Chehab, 2021. Digital Forensics vs. Anti-Digital Forensics: Techniques, Limitations and Recommendations, In: *American University of Beirut, Electrical and Computer Engineering Department.*

[4] Digital Evidence and Forensics [online], [cit. 2024-05-09]. Dostupné na: https://nij.ojp.gov/digital-evidence-and-forensics

[5] C. M. Whitcomb, 2002. An historical perspective of digital evidence: A forensic scientist's view, In: *International Journal of Digital Evidence 1.1, 2002: 7-15*

[6] E. Casey, 2011. Digital evidence and computer crime: Forensic science, computers, and the internet. Academic press

[7] Windows Forensics Analysis — Windows Artifacts (Part I) [online]. [cit. 2024-03-14]. Dostupné na: https://nasbench.medium.com/windows-forensics-analysis-windows-artifacts-part-i-c7ad81ada16c

[8] Ewa Huebnera, Derek Bema, Cheong Kai Wee, 2006. In: *Digital investigation 3*

[9] Michael Galhuber, Robert Luh, Time for Truth: Forensic Analysis of NTFS Timestamps [online], [cit. 2024-05-17]. Dostupné na: https://eprints.cs.univie.ac.at/7091/1/3465481.3470016.pdf

[10] NTFS Timestamp changes on Windows 10 [online], [cit. 2024-05-09]. Dostupné na: https://www.senturean.com/posts/19_04_22_win10_ntfs_time_rules/

[11] P. K. Sahoo, R. K. Chottray, S. Pattnaiak, 2012. Research Issues on Windows Event Log, In: *International Journal of Computer Applications*.

[12] Event Logging [online], [cit. 2024-05-17]. Dostupné na: https://learn.microsoft.com/en-us/windows/win32/eventlog/event-logging

[13] Security and resilience — Vocabulary [online], [cit. 2024-05-17]. Dostupné na: https://www.iso.org/standard/77008.html

[14] A. Shaaban, K. Sapronov, 2016. Practical Windows Forensics: Leverage the power of digital forensics for Windows systems

[15] Event Schema Elements [online], [cit. 2024-05-17]. Dostupné na: https://learn.microsoft.com/en-us/windows/win32/wes/eventschema-elements

[16] Windows Server 2003 Resource Kit Registry Reference [online], [cit. 2024-05-17]. Dostupné na: https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc778196(v=ws.10)

[17] Harlan Carvey, 2015. Windows Registry Forensics: Advanced Digital Forensic Analysis of the Windows Registry

[18] The System Registry [online], [cit. 2024-05-17]. Dostupné na: https://learn.microsoft.com/en-us/previous-versions/windows/internet-explorer/ie-developer/scripting-articles/ms970651(v=msdn.10)?redirectedfrom=MSDN

[19] Md. Saidur Rahman, 2017. Basic Graph Theory In: *Undergraduate Topics in Computer Science*. Cham, Switzerland: Springer International Publishing AG, s. 1–30. ISBN 978-3-319-49475-3.

[20] Peter Dankelmann, Wayne Goddard, Christine S. Swart. The Average Eccentricity of a Graph and its Subgraphs

[21] Iqbal, Salman, Soltan Abed Alharbi, 2019. Advancing automation in digital forensic investigations using machine learning forensics, In: *Digital Forensic Science. IntechOpen*.

[22] Stefania Costantini, Giovanni De Gasperis, Raffaele Olivieri, 2019. Digital forensics and investigations meet artificial intelligence, In: *Annals of Mathematics and Artificial Intelligence.*

[23] Chuck Easttom, 2017. Utilizing Graph Theory to Model Forensic Examination, In: *International Journal of Innovative Research in Information Security (IJIRIS).* Vol. 04, no. 02. ISSN: 2349-7009(P)

[24] Imani Palmer, Roy Campbell, Boris Gelfand, 2017. Exploring Digital Evidence with Graph Theory, In: *ADFSL Conference on Digital Forensics, Security and Law.*

[25] Selim Ozcan, Merve Astekin, Narasimha K. Shashidhar, Bing Zhou, 2020. Centrality and Scalability Analysis on Distributed Graph of Large-Scale E-mail Dataset for Digital Forensics, In: *IEEE International Conference on Big Data (Big Data).*

[26] Jitendra Binwal, Renu Devi, Bhupendra Singh, 2023. Mathematical modelling and simulation of fingerprint analysis using graph isomorphism, domination, and graph pebbling, In: *Advances and Applications in Discrete Mathematics.*

[27] Changwei Liu, Anoop Singhal, Duminda Wijesekera, 2012. Using Attack Graphs in Forensic Examinations, In: *2012 Seventh International Conference on Availability, Reliability and Security.*

[28] Sherenaz Al-Haj Baddar, Alessio Merlo, Mauro Migliardi, 2014. Anomaly Detection in Computer Networks: a State-of-the-Art Review.

[29] Wei Wang, Thomas E. Daniels 2005. Building Evidence Graphs for Network Forensics Analysis, In: *21st Annual Computer Security Applications Conference (ACSAC'05).*

[30] Jingsha He, Chengyue Chang, Peng He, Muhammad Salman Pathan 2016. Network Forensics Method Based on Evidence Graph and Vulnerability Reasoning, In: *MDPI, future internet.*

[31] Connor McGee, Jing Guo, Zhaohong Wang 2021. The Application of the Graph Laplacian in Network Forensics, In: *2021 IEEE 12th Annual Infor-*

mation Technology, Electronics and Mobile Communication Conference (IEMCON).

[32] Case 001 – The Stolen Szechuan Sauce [online], [cit. 2024-02-18]. Dostupné na: `https://dfirmadness.com/the-stolen-szechuan-sauce/`

[33] Magnet CTF 2019 Windows Desktop, 2019 [online], [cit. 2024-02-18]. Dostupné na: `https://digitalcorpora.s3.amazonaws.com/corpora/scenarios/magnet/2019%20CTF%20-%20Windows-Desktop.zip.`

[34] Magnet CTF 2020 Windows, 2020, [online], [cit. 2024-02-18]. Dostupné na: `https://digitalcorpora.s3.amazonaws.com/corpora/scenarios/magnet/2020%20CTF%20-%20Windows.zip.`

[35] Magnet CTF 2022 Windows, 2022, [online], [cit. 2024-02-18]. Dostupné na: `https://digitalcorpora.s3.amazonaws.com/corpora/scenarios/magnet/2022%20CTF%20-%20Windows.zip.`

[36] Data Leakage Case, [online], [cit. 2024-02-18]. Dostupné na: `https://cfreds-archive.nist.gov/data_leakage_case/data-leakage-case.html.`

[37] Hacking Case, [online], [cit. 2024-02-18]. Dostupné na: `https://cfreds-archive.nist.gov/Hacking_Case.html`

[38] Eva Marková, Pavol Sokol, Kristína Kováčová, 2022, Detection of relevant digital evidence in the forensic timelines, In:*International Conference on Electronics, Computers and Artificial Intelligence (ECAI) (pp. 1-7). IEEE*

[39] Bhandari, S., Jusas, V., 2020, The Phases Based Approach for Regeneration of Timeline in Digital Forensics, In: *2020 International Conference on INnovations in Intelligent SysTems and Applications (INISTA).*

[40] Plaso [online]. [cit. 2024-04-13]. Dostupné z: `https://plaso.readthedocs.io/en/latest`

Prílohy

Príloha A

Popis atribútov tabuľky EVT

Atribút	**Popis atribútu**
inode	pôvodný stĺpec inode, identifikácia Inode v rámci NTFS súborového systému
datetime	časová pečiatka, spojené dátumové a časové stĺpce, časové pásmo UTC
computer_name	extrahované z stĺpca desc alebo extra
source_name	extrahované z stĺpca desc alebo extra - zdroj logu (napr. Service Control Manager)
filename_type	
filename_security	extrahované z názvu súboru, logovací stĺpec v Security.evtx
filename_application	extrahované z názvu súboru, logovací stĺpec v Application.evtx
filename_system	extrahované z názvu súboru, logovací stĺpec v System.evtx
filename_rdp	extrahované z názvu súboru, log pod Microsoft-Windows-RemoteDesktopServices-RdpCoreTS%4Operational.evtx
filename_powershell	extrahované z názvu súboru, log v Microsoft-Windows-PowerShell%4Operational.evtx
filename_other	extrahované z názvu súboru, log sa neobjavil v žiadnych vyššie uvedených skupinách v názve

Atribút	**Popis atribútu**
recovered	extrahované z extra stĺpca, True, ak bol záznam obnovený (obnovené dáta = zmazané dáta)
user_sid	extrahované z extra stĺpca, ID používateľa, ktorý vytvoril proces
execution_process_id	extrahované z extra stĺpca, ID procesu, ktorý sa vykonáva
channel	extrahované z extra stĺpca, napr. System
event_id	špecifický typ udalosti
audit_account_logon	rozdelené na základe skupín evt - do 9 skupín, určených pomocou Event ID
audit_account_management	rozdelené na základe skupín evt - do 9 skupín, určených pomocou Event ID
audit_detailed_tracking	rozdelené na základe skupín evt - do 9 skupín, určených pomocou Event ID
audit_logon_logoff	rozdelené na základe skupín evt - do 9 skupín, určených pomocou Event ID
audit_ds_access	rozdelené na základe skupín evt - do 9 skupín, určených pomocou Event ID
audit_object_access	rozdelené na základe skupín evt - do 9 skupín, určených pomocou Event ID
audit_policy_change	rozdelené na základe skupín evt - do 9 skupín, určených pomocou Event ID
audit_privilege_usage	rozdelené na základe skupín evt - do 9 skupín, určených pomocou Event ID
audit_system	rozdelené na základe skupín evt - do 9 skupín, určených pomocou Event ID
task	extrahované z extra stĺpca. Získava identifikátor úlohy pre časť aplikácie alebo komponentu, ktorý zverejňuje udalosť. Úloha je hodnota s dĺžkou 16 bitov, pričom 16 najvyšších hodnôt je rezervovaných. Tento typ umožňuje použitie ľubovoľnej hodnoty medzi 0x0000 a 0xffef.
keywords	

Atribút	**Popis atribútu**
keywords_audit_failure	AuditFailure = 0x10000000000000L, pripojené ku všetkým neúspešným bezpečnostným udalostiam auditu. Toto kľúčové slovo používajte iba pre udalosti v bezpečnostnom zázname.
keywords_audit_success	AuditSuccess = 0x20000000000000L, Pripojené ku všetkým úspešným bezpečnostným udalostiam auditu.
keywords_correlation_hint	CorrelationHint = 0x10000000000000, Pripojené ku všetkým úspešným bezpečnostným udalostiam auditu.
keywords_event_log_classic	EventLogClassic = 0x80000000000000L, Pripojené k udalostiam, ktoré sú vyvolané pomocou funkcie RaiseEvent.
keywords_sqm	Sqm = 0x08000000000000L, Pripojené ku všetkým udalostiam služby kvality služby (SQM).
keywords_wdi_context	WdiContext = 0x02000000000000L, Pripojené ku všetkým udalostiam infraštruktúry diagnostiky systému Windows (WDI).
keywords_wdi_diagnostic	WdiDiagnostic = 0x04000000000000L, Pripojené ku všetkým diagnostickým udalostiam infraštruktúry diagnostiky systému Windows (WDI).
timestamp	modifikovaná časová pečiatka (dátum a čas)
epochtime	epochtime (od 1.1.1970)
hour	hodina (extrakcia z časovej pečiatky)
minute	minúta (extrakcia z časovej pečiatky)

Popis atribútov tabuľky EVT

Príloha B

Popis atribútov tabuľky FILE

Atribút	**Popis atribútu**
filename	názov súboru
inode	pôvodný stĺpec inode, označenie Inode v rámci súborového systému NTFS
extra	stĺpec s informáciami - z tohto stĺpca boli extrahované rôzne atribúty
id	stĺpec s vygenerovaným unikátnym identifikátorom (na získanie záznamu)
datetime	časová pečiatka, spojenie stĺpcov dátumu a času, časové pásmo UTC
M	modifikované
a	prístup
C	zmenené, modifikované MFT
B	vytvorené, čas vytvorenia súboru
file_stat	stĺpec na základe hodnôt zo stĺpca sourcetype
NTFS_file_stat	stĺpec vytvorený na základe hodnôt zo stĺpca sourcetype
file_entry_shell_item	stĺpec na základe hodnôt zo stĺpca sourcetype
NTFS_USN_change	stĺpec na základe hodnôt zo stĺpca sourcetype
file_path	súbor s celou cestou
name	názov stĺpca extrahovaný z desc
typef	typ súboru ('None', 'súbor', 'adresár', 'odkaz')
filef	1, ak je to súbor, 0, ak nie

Atribút	**Popis atribútu**
directory	1, ak je to priečinok, 0, ak nie
link	1, ak je to súbor odkazu, 0, ak nie
dir_type	typ priečinka - umiestnenie súboru ('Windows', 'AppData', 'Iný', 'Používatelia')
dir_appdata	popis cesty - extrahovaný stĺpec - na základe hodnôt z desc, 1, ak je AppData v ceste
dir_win	popis cesty - extrahovaný stĺpec - na základe hodnôt z desc, 1, ak je Windows v ceste
dir_user	popis cesty - extrahovaný stĺpec - na základe hodnôt z desc, 1, ak je Používateľ v ceste
dir_other	popis cesty - extrahovaný stĺpec - na základe hodnôt z desc, 1, ak nie je žiadne z troch vyššie uvedených
file_type	typ súboru podľa prípony
file_executable	typ súboru podľa prípony - stĺpec extrahovaný na základe hodnôt typu súboru z názvu (spustiteľné súbory)
file_graphic	typ súboru podľa prípony - stĺpec extrahovaný na základe hodnôt typu súboru z názvu (grafické súbory)
file_documents	typ súboru podľa prípony - stĺpec extrahovaný na základe hodnôt typu súboru z názvu (dokumenty)
file_ps	typ súboru podľa prípony - stĺpec extrahovaný na základe hodnôt typu súboru z názvu (súbory powershell)
file_other	typ súboru podľa prípony - stĺpec extrahovaný na základe hodnôt typu súboru z názvu (ostatné)
mft	stĺpec vytvorený na základe hodnôt z formátovacieho stĺpca
lnk_shell_items	stĺpec vytvorený na základe hodnôt z formátovacieho stĺpca
olecf_olecf_automatic...	stĺpec vytvorený na základe hodnôt z formátovacieho stĺpca

Atribút	**Popis atribútu**
winreg_bagmru/shell_items	stĺpec vytvorený na základe hodnôt z formátovacieho stĺpca
usnjrnl	stĺpec vytvorený na základe hodnôt z formátovacieho stĺpca
is_allocated	stĺpec z extrahovanej hodnoty zo stĺpca extra (1, ak sa informácia nachádza)
is_allocated0	1, ak bola hodnota v stĺpci is_allocated jedna (súbor sa nachádza v súborovom systéme)
is_allocated1	1, ak bola hodnota v stĺpci is_allocated nula (súbor bol odstránený)
file_size	veľkosť súboru
sha_256	hash extrahovaný zo stĺpca extra
timestamp	modifikovaná časová pečiatka (dátum a čas)
epochtime	epoch čas (od 1.1.1970)
hour	hodina (extrakcia z časovej pečiatky)
minute	minúta (extrakcia z časovej pečiatky)

Popis atribútov tabuľky FILE

Príloha C

Zdrojový kód

```
# Importy a načítanie dát
import pandas as pd
import networkx as nx
import matplotlib.pyplot as plt

df = pd.read_csv('../Data_input/dc_evt.csv',
     delimiter=",",encoding="utf8")

# Počet unikátnych hodnôt v jednotlivých stĺpcoch
for x in df.columns:
    n = str(x)
    unique_values = df[n].unique()
    print(x, unique_values.shape)

# Spájanie niektorých atribútov v datasete FILE
  späť do kategoriálnej podoby
df['MACB'] = ''

for index, row in df.iterrows():
    macb = ''
    if row['M'] == 1:
        macb += 'M'
    if row['A'] == 1:
        macb += 'A'
```

```
    if row['C'] == 1:
        macb += 'C'
    if row['B'] == 1:
        macb += 'B'
    df.at[index, 'MACB'] = macb

df['dir'] = ''

for index, row in df.iterrows():
    dir_t = ''
    if row['dir_appdata'] == 1:
        dir_t += 'dir_appdata'
    if row['dir_win'] == 1:
        dir_t += 'dir_win'
    if row['dir_user'] == 1:
        dir_t += 'dir_user'
    if row['dir_other'] == 1:
        dir_t += 'dir_other'
    df.at[index, 'dir'] = dir_t

df['file'] = ''

for index, row in df.iterrows():
    file = ''
    if row['file_executable'] == 1:
        file += 'file_executable'
    if row['file_graphic'] == 1:
        file += 'file_graphic'
    if row['file_documents'] == 1:
        file += 'file_documents'
    if row['file_ps'] == 1:
        file += 'file_ps'
    if row['file_other'] == 1:
        file += 'file_other'
    df.at[index, 'file'] = file
```

```
# Niektoré vybrané stĺpce, ktoré môžu byť pre nás relevantné
  a majú nižší počet unikátnych hodnôt
df[["inode","computer_name", "source_name","user_sid","channel",
    "event_id","task","keywords"]]

# Funkcia pre zistenie stĺpca z vrcholu a funkcia pre vykreslenie
  grafu a uloženie do súboru
def find_column_for_node(graph, df, node):
    for column in df.columns:
        if node in df[column].values:
            return column
    return None

def drawGraph(G, df, columns_of_graph_G, df2):
    plt.figure(figsize=(12, 8))

    # Nastavenie veľkosti vrcholov podľa stupňa (počet hrán)
    node_size = [G.degree(node) * 20 for node in G.nodes()]
    color_map = []
    column = columns_of_graph_G[0]
    for node in G:
        if find_column_for_node(G, df2, node) == column:
            color_map.append('blue')
        else:
            color_map.append('red')

    # Explicitné určenie pozícií vrcholov
    pos = nx.spring_layout(G)

    # Posunutie popisov vrcholov o niečo vyššie
    pos_labels = {node: (x, y + 0.025) for node, (x, y) in pos.items()}

    # Nakreslenie grafu s pozíciami a popismi vedľa vrcholov
    nx.draw(G, pos=pos, with_labels=False, node_size=node_size,
```

```
    node_color=color_map)

    # Nakreslenie popisov vrcholov
    nx.draw_networkx_labels(G, pos_labels, font_size=10)

    cesta_k_output = f'../Data_output/DC/all_graph_evt{
    columns_of_graph_G[0]}_{columns_of_graph_G[1]}.jpg'
    plt.savefig(cesta_k_output)
    plt.show()

# Vytvorenie grafu napr. inode, user_sid
G = nx.from_pandas_edgelist(df, 'inode', 'user_sid')
columns_of_graph_G = ['inode', 'user_sid']
df2 = df[columns_of_graph_G].copy()
df2

drawGraph(G, df, columns_of_graph_G, df2)

# Bipartitnosť grafu
is_bipartite = nx.is_bipartite(G)

if is_bipartite:
    print("Graf je bipartitný.")
else:
    print("Graf nie je bipartitný.")

# Súvislosť grafu
is_connected = nx.is_connected(G)

if is_connected:
    print("Graf je súvislý.")
else:
    print("Graf nie je súvislý.")
```

```
# Excentricity grafu
eccentricities = nx.eccentricity(G)
print("Excentricity vrcholov:", eccentricities)

# Najnižšia excentricita a vrcholy jej prislúchajúce
minimal = 1000

df2 = df[columns_of_graph_G].copy()
for node_of_interest in G.nodes:
    eccentricity_of_node = nx.eccentricity(G, v=node_of_interest)
    if eccentricity_of_node < minimal:
        minimal = eccentricity_of_node

for node_of_interest in G.nodes:
    eccentricity_of_node = nx.eccentricity(G, v=node_of_interest)
    if eccentricity_of_node == minimal:
        column_name = df2.columns[df2.isin([node_of_interest])
                        .any()].tolist()[0]
        print(f"Atribut {column_name} - Excentricita vrcholu
        {node_of_interest}: {eccentricity_of_node}" )


# Vyhľadanie bázy cyklov v grafe
cycle_basis = nx.cycle_basis(G)
```