UNIVERZITA PAVLA JOZEFA ŠAFÁRIKA V KOŠICIACH

PRÍRODOVEDECKÁ FAKULTA

ANALÝZA ÚDAJOV O BEZPEČNOSTNÝCH HROZBÁCH PRI RIEŠENÍ BEZPEČNOSTNÝCH INCIDENTOV

2022 **Jakub MOHLER**

UNIVERYITA PAVLA JOZEFA ŠAFÁRIKA V KOŠICIACH

PRÍRODOVEDECKÁ FAKULTA

ANALÝZA ÚDAJOV O BEZPEČNOSTNÝCH HROZBÁCH PRI RIEŠENÍ BEZPEČNOSTNÝCH INCIDENTOV

BAKALÁRSKA PRÁCA

Študijný program:	Informatika
Pracovisko (katedra/ústav):	Ústav informatiky
Vedúci bakalárskej práce:	RNDr. JUDr. Pavol Sokol, PhD.
Konzultant bakalárskej práce:	Mgr. Ladislav Bačo

Košice 2022 **Jakub MOHLER**

Univerzita P. J. Šafárika v Košiciach
Prírodovedecká fakulta

ZADANIE ZÁVEREČNEJ PRÁCE

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

Názov: Analýza údajov o bezpečnostných hrozbách pri riešení bezpečnostných incidentov

Názov EN: Analysis of data on security threats in the security incidents handling

Cieľ: (1) Analyzovať možnosti spracovania údajov o bezpečnostných hrozbách, najmä indikátorov kompromitácie prostredníctvom tzv. threat intelligence
(2) Analyzovať nástroje a prístupy k analýze bezpečnostných hrozieb.
(3) Navrhnúť, implementovať a vyhodnotiť nástroj na spracovanie údajov o bezpečnostných hrozbách, najmä indikátorov kompromitácie pri riešení bezpečnostných incidentov prostredníctvom tzv. threat intelligence.

Literatúra: (1) Skopik F, editor. Collaborative cyber threat intelligence: detecting and responding to advanced cyber attacks at the national level. CRC Press; 2017 Oct 16.
(2) Pace C. The Threat Intelligence Handbook: A Practical Guide for Security Teams to Unlocking the Power of Intelligence. Annapolis, CyberEdge Group. 2018.
(3) Roberts SJ, Brown R. Intelligence-Driven Incident Response: Outwitting the Adversary. " O'Reilly Media, Inc."; 2017 Aug 21.

Vedúci: RNDr. JUDr. Pavol Sokol, PhD.
Konzultant: Mgr. Ladislav Bačo
Ústav : ÚINF - Ústav informatiky
Riaditeľ ústavu: doc. RNDr. Ondrej Krídlo, PhD.

Dátum schválenia: 12.05.2021

Poďakovanie

Týmto sa chcem poďakovať vedúcemu svojej práce RNDr. JUDr. Pavlovi Sokolovi, PhD. a konzultantovi Mgr. Ladislavovi Bačovi za odborné vedenie, cenné rady a veľkú pomoc počas tvorby práce.

Abstrakt v štátnom jazyku

Táto práca je zameraná na analýzu údajov o bezpečnostných hrozbách pri riešení bezpečnostných incidentov. Hlavným cieľom práce je navrhnúť a implementovať nástroj, ktorý bude schopný spracovávať údaje o bezpečnostných hrozbách pomocou threat intelligence. Zameriavame sa hlavne na indikátory kompromitácie, keďže vyhodnocovanie indikátorov kompromitácie je súčasťou riešenia bezpečnostných incidentov. Je dôležité vedieť rýchlo analyzovať bezpečnostné incidenty, pre rýchlu odpoveď na tieto incidenty, k čomu môžu pomáhať rôzne nástroje, ktoré môžu analytikom uľahčiť prácu. V rámci nami implementovaného systému využívame nástroje s otvoreným kódom a regulárne výrazy, pomocou ktorých dokážeme vyhľadať indikátory kompromitácie a následne ich obohatiť o doplňujúce informácie. V práci sa taktiež venujeme analýze a porovnaniu už existujúcich riešení.

Kľúčové slová: bezpečnostný incident, threat intelligence, indikátory kompromitácie

Abstrakt v cudzom jazyku

This work is focused on the analysis of cyber security threats data in dealing with cybersecurity incidents. The main goal of this work is to design and implement a tool that will be able to process data on security threats using threat intelligence. We focus mainly on indicators of compromise, whereas evaluation of indicators of compromise is part of dealing with cybersecurity incidents. It is important to be able to quickly analyze cybersecurity incidents for fast incident response, which can be aided by various tools that can make analytics work easier. Within the system implemented by us we use open-source tools and regular expressions, which we use to find indicators of compromise and then enrich them with additional information. In this work we also deal with analysis and comparison of existing solutions.

Keywords: cybersecurity incident, threat intelligence, indicators of compromise

Obsah

Zoznam ilustrácií

Zoznam skratiek a značiek

API Application Programing Interface, rozhranie pre programovanie aplikácií

BTC Bitcoin, bitcoin

IDS Intrusion Detection System, systém detekcie narušenia

IoC Indicator of Compromise, indikátor kompromitácie

IPS Intrusion Prevention System, systém prevencie narušenia

JSON JavaScript Object Notation, JavaScriptový objektový zápis

SIEM Security Information and Event Management, manažment bezpečnostných informácií a udalostí

TTP Tactics, techniques and procedures, taktiky, techniky a procedúry

YARA YARA rules, YARA pravidlá

Úvod

Neustály rozvoj internetu, dostupnosť informácií, online komunikácia a elektronizácia služieb ukazujú veľké možnosti kybernetického priestoru. Avšak spolu s novými možnosťami prichádza stále väčšie množstvo nových bezpečnostných hrozieb, pred ktorými je potrebné sa chrániť. S ochranou aktív pred rôznymi bezpečnostnými hrozbami úzko súvisí informačná a kybernetická bezpečnosť. Kým kybernetická bezpečnosť sa špecializuje na ochranu údajov v kybernetickom priestore, informačná bezpečnosť sa venuje ochrane informácii a údajov bez ohľadu na formu uloženia. Spolu s rozširujúcimi sa bezpečnostnými hrozbami je dôležité sa venovať týmto odvetviam a zdokonaľovať ochranné postupy a prostriedky na základe bezpečnostných incidentov.

Dôležitou súčasťou informačnej a kybernetickej bezpečnosti je riešenie kybernetických bezpečnostných incidentov. Tieto incidenty je možné definovať ako udalosti, ktoré majú potencionálne negatívny dopad na prevádzku a jej aktíva (informačné systémy, počítačovú sieť). Reakcia na bezpečnostné incidenty na druhej strane predstavuje súbor zásad a postupov informačnej bezpečnosti, ktoré sa môžu použiť na identifikáciu, potlačenie a elimináciu kybernetických útokov. Vďaka riešeniu bezpečnostných incidentov a zdieľaniu dát o útokoch, je možné sa v budúcnosti lepšie chrániť pred možnými incidentmi.

Súčasťou riešenia bezpečnostných incidentov je aj vyhodnocovanie indikátorov kompromitácie (IoC). Indikátory kompromitácie poukazujú na dáta, ktoré indikujú narušenie (kompromitáciu) systému. Tým poskytujú bezpečnostným tímom (CSIRT) dôležité znalosti po úniku údajov alebo inom narušení bezpečnosti. Je dôležité ich efektívne vyhľadávať a spracovávať pre detekciu a prevenciu pred ďalšími možnými útokmi alebo dostatočne skorým zastavením útoku. Týmto môže dôjsť k zníženiu miery poškodenia organizácie.

Hlavným cieľom tejto záverečnej práce je navrhnúť vhodný nástroj, ktorý by uľahčoval spracovanie údajov o bezpečnostných hrozbách. Z obrazov disku by mal byť tento nástroj schopný extrahovať sledované indikátory kompromitácie a obohatiť ich o ďalšie informácie pomocou threat intelligence.

V prvom cieli práce sa zameriavame na možnosti spracovania údajov o bezpečnostných hrozbách prostredníctvom threat intelligence. Sledovanými údajmi sú

najmä indikátory kompromitácie. Súčasťou druhého cieľa je analyzovať dostupné nástroje a rôzne prístupy k analýze bezpečnostných hrozieb. Posledným cieľom tejto práce je návrh a implementácia samotného nástroja na spracovanie údajov o bezpečnostných hrozbách. Tento nástroj sa bude zameriavať na spracovanie indikátorov kompromitácie pri riešení bezpečnostných incidentov prostredníctvom threat intelligence.

Práca je rozdelená do štyroch základných kapitol. V prvej kapitole sa venujeme základným definíciám, životnému cyklu a rôznym typom threat intelligence. Tiež v tejto kapitole skúmame rôzne modely threat intelligence a prípady použitia v reálnom svete. Druhá kapitola sa zaoberá indikátormi kompromitácie, ich typmi a komponentami. Taktiež sa zaoberá konceptom pyramídy bolesti a rôznymi formami použitia a uloženia indikátorov kompromitácie. V tretej kapitole porovnávame nástroje. Porovnávame nástroje v oblasti extrakcie indikátorov kompromitácie, skenovania pre indikátory kompromitácie a threat intelligence platformy. Tieto nástroje porovnávame podľa rôznych vybraných atribútov. Štvrtá kapitola je zameraná na nástroj na spracovanie indikátorov kompromitácie. Vysvetľuje sa návrh tohto nástroja, použité technológie a konkrétna implementácia nástroja rozdelená do jednotlivých krokov.

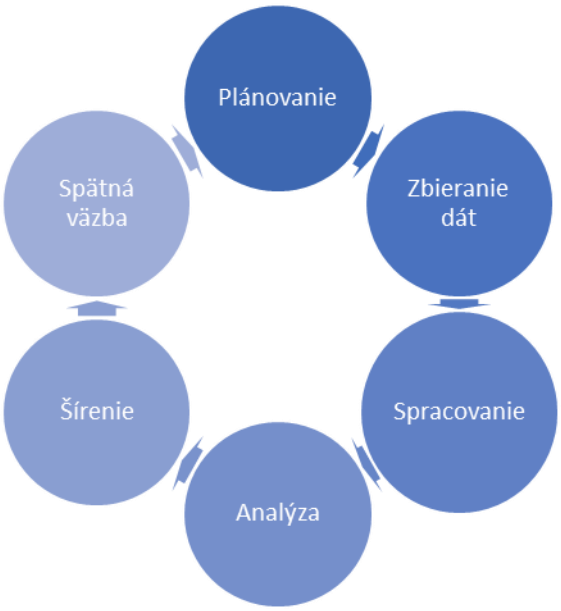
1 Threat Intelligence

Threat intelligence dokáže identifikovať a analyzovať kybernetické bezpečnostné hrozby zamerané na konkrétnu spoločnosť. Threat intelligence je o triedení veľkého množstva údajov. Kontextovo to skúma, aby boli nájdené skutočné problémy a nasadilo sa riešenie špecifické pre nájdený problém [1]. Súčasne môžeme povedať, že ho tvoria poznatky založené na stopách vrátane kontextu, mechanizmov, indikátorov, dôsledkov a rád založených na akciách v súvislosti s existujúcou alebo vznikajúcou hrozbou alebo rizikom pre aktíva. Tieto poznatky môžu byť použité na informovanie o rozhodnutiach týkajúcich sa reakcie subjektu na túto hrozbu alebo riziko [2].

Threat intelligence je dôležitou súčasťou každého ekosystému kybernetickej bezpečnosti. Rôzne programy chránia spoločnosti pred stratou dát tým, že môžu odhaliť bezpečnostné hrozby a zabrániť tak úniku dát. Môžu poskytnúť smer, akým sa uberať pri vykonávaní bezpečnostných opatrení. Identifikáciou a analýzou bezpečnostných hrozieb spozorujú spôsoby, akými útočníci prevádzajú svoje útoky a môžu tak pomôcť zaviesť bezpečnostné opatrenia, ktoré zabránia podobným útokom. Taktiež môžu informovať ostatné spoločnosti o taktikách hekerov a vytvárať tak kolektívnu znalosť.

1.1 Životný cyklus threat intelligence

Threat intelligence nie je len o nespracovaných dátach. Je to proces, ktorý sa skladá zo šesť dielneho cyklu zhromažďovania, spracovania a analyzovania údajov [1]. Ako môžeme vidieť na Obrázku č. 1, ide o sústavne sa opakujúci proces, pretože v priebehu vývoja sa vyskytujú nové otázky a medzery vo vedomostiach, ktoré vedú ku stanoveniu nových požiadaviek na zhromažďovanie, spracovanie a analýzu dát.



Obr. 1 Životný cyklus threat inteligenčne [prevzaté z [1]]

1.1.1 Plánovanie

V prvej fáze je potrebné si stanoviť priority na základe faktorov, ako napríklad ako veľmi sa dotýkajú základných hodnôt danej spoločnosti alebo aký veľký vplyv bude mať výsledné rozhodnutie a ako veľmi je toto rozhodnutie citlivé na čas. Jedným z dôležitých usmerňujúcich faktorov v tejto fáze je pochopenie toho, kto bude používať a mať úžitok z výsledného výsledku [3]. Tento výsledok môže ísť ku tímu analytikov s technickou odbornosťou, ktorí potrebujú rýchlu správu o novom exploite alebo ku výkonnému pracovníkovi, ktorý potrebuje široký prehľad trendov v oblasti bezpečnosti, aby vedel informovať ich investície do bezpečnosti na najbližší čas.

1.1.2 Zber dát

Ďalším krokom threat intelligence je zhromaždenie nespracovaných údajov, ktoré spĺňajú požiadavky stanovené v prvej fáze. Najlepšie je zhromažďovať údaje zo širokej škály zdrojov. Tieto zdroje môžu byť interné, ako sú logy z počítačovej siete a záznamy predchádzajúcich reakcií na incidenty alebo externé z webu, dark webu a iných technických zdrojov. Údaje o bezpečnostných hrozbách si môžeme predstaviť ako zoznamy indikátorov kompromitácie napríklad škodlivé IP adresy, domény a heše súborov [4]. Môžu však obsahovať aj informácie o zraniteľnostiach, ako sú napríklad

osobné údaje zákazníkov, nespracovaný kód z podvodných stránok a text zo spravodajských zdrojov alebo sociálnych sietí [2].

1.1.3 Spracovanie

Po zhromaždení všetkých nespracovaných údajov je potrebné ich usporiadať, roztriediť pomocou označení (tagov) metadát a odfiltrovať nadbytočné informácie. Tiež je dôležité odstrániť údaje, ktoré nesprávne identifikujú určitú udalosť (false positives) [1]. V tejto dobe aj malé organizácie zbierajú milióny záznamov (logov) a indikátorov denne. To je príliš veľké množstvo na to, aby ich ručne prezerali analytici. Z tohto dôvodu musia byť zber a spracovanie údajov automatizované pomocou rôznych systémov. Jedným z týchto systémov je SIEM. V SIEM systéme sa dá robiť relatívne ľahká štruktúra dát pomocou korelačných pravidiel, ktoré je možné nastaviť pre niekoľko rôznych prípadov použitia [2]. Ak sa zhromažďujú neštruktúrované údaje z mnohých rôznych interných a externých zdrojov, je potrebné nasadiť robustnejší systém. Jedným z vývznamných príkladov v tomto smere je spoločnosť, ktorá sa venuje threat intelligence, a to Recorded Future [5]. Ich platforma využíva strojové učenie a spracovanie bežného jazyka na analýzu textu z miliónov neštruktúrovaných dokumentov v siedmich rôznych jazykoch a ich klasifikáciu pomocou jazykovo nezávislých ontológií a udalostí. To analytikom umožňuje vykonávať výkonné a intuitívne vyhľadávania, ktoré presahujú jednoduché kľúčové slová a pravidlá jednoduchej korelácie.

1.1.4 Analýza

Ďalším krokom je pochopenie spracovaných údajov. Cieľom analýzy je vyhľadávať potenciálne bezpečnostné problémy a informovať príslušné tímy vo formáte, ktorý spĺňa požiadavky uvedené vo fáze plánovania a riadenia [4]. Threat intelligence môže mať rôznu podobu v závislosti od počiatočných cieľov a publika. Cieľom však je dostať údaje do formátu, ktorému bude verejnosť rozumieť. Môže to siahať od jednoduchých zoznamov hrozieb až po reporty s recenziou.

1.1.5 Šírenie

Výsledok sa potom distribuuje určeným spotrebiteľom. Ak má byť threat intelligence uplatniteľné, musí sa dostať ku všetkým spoločnostiam a organizáciám, ktoré to potrebujú včas [2]. Je to z dôvodu, aby sa tieto organizácie dokázali brániť pred týmito hrozbami. Taktiež to musí byť sledované, aby medzi jednotlivými životnými cyklami threat intelligence existovala kontinuita a všetky poznatky sa nestratili.

1.1.6 Spätná väzba

Posledným krokom je uzavretie tohto cyklu. To úzko súvisí s počiatočnou fázou plánovania a riadenia. Po prijatí výsledku sa skontroluje a rozhodne, či boli požiadavky prvej fázy splnené. Toto riadi ciele a postupy nasledujúceho cyklu, čím sa opäť stáva nevyhnutnou dokumentácia a kontinuita [1].

1.2 Typy threat intelligence

V predchádzajúcej podkapitole sme sa zamerali na životný cyklus threat intelligence. V tejto podkapitole si ukážeme a vysvetlíme rôzne typy threat intelligence. Threat intelligence možno rozdeliť do štyroch kategórií, ktoré si nižšie bližšie popíšeme [6]:

- **strategické** - poskytuje širšie trendy, typicky určené pre netechnické publikum,
- **taktické** - načrtáva taktiky, techniky a procedúry útočníka pre technickejšie publikum,
- **prevádzkové** - obsahuje technické podrobnosti o konkrétnych útokoch a kampaniach,
- **technické** - dáva informácie o útočníkových zdrojoch, ktoré boli použité pri útoku.

1.2.1 Strategické threat intelligence

Strategické threat intelligence poskytuje široký prehľad bezpečnostných hrozieb pre organizáciu. Je to určené na informovanie pre rozhodnutia na vyššej úrovni, ktoré prijímajú riadiaci pracovníci a iní činitelia s rozhodovacou právomocou v organizácii [1]. Obsah je menej technický a je prezentovaný prostredníctvom správ alebo brífingov. Kvalitné strategické spravodajstvo by malo poskytnúť informácie o oblastiach, ako sú riziká spojené s určitými smermi konania, základné taktiky a ciele útočníkov a geopolitické udalosti a trendy. Bežným zdrojom informácií pre strategické threat intelligence sú dokumenty obsahujúce politiky z mimovládnych organizácií, správy z miestnych a národných médií, odborných publikácií alebo iných odborníkov v danej oblasti, reporty a ďalšie materiály vytvárané bezpečnostnými spoločnosťami. Produkcia strategických threat intelligence začína kladením konkrétnych otázok zameraných na stanovenie požiadaviek. Taktiež je potrebné mať analytikov so skúsenosťami mimo oblasti kybernetickej bezpečnosti, napríklad s dobrým porozumením pre sociálno-politické a obchodné koncepty.

1.2.2 Taktické threat intelligence

Taktické threat intelligence načrtáva taktiky, techniky a procedúry (Tactics, techniques and procedures, TTP) aktérov hrozieb. Malo by organizáciám pomôcť konkrétnym spôsobom pochopiť, ako môže dôjsť k útoku na ich organizáciu a nájsť najlepší spôsob, akým sa proti týmto útokom brániť [1]. Zvyčajne zahŕňa technický kontext a používajú ho pracovníci priamo zapojení do bezpečnostného tímu organizácie, napríklad architekti systému, správcovia a bezpečnostní pracovníci. Reporty dodávané bezpečnostnými agentmi sú najjednoduchším spôsobom, ako získať informácie o taktických hrozbách. V týchto reportoch sa nachádzajú informácie o vektoroch útokov, nástrojoch a infraštruktúre, ktoré útočníci používajú, vrátane podrobností o tom, na ktoré zraniteľné miesta sa zameriavajú. Taktické informácie o hrozbách by sa mali používať na informovanie o zlepšeniach existujúcich bezpečnostných kontrol a procesov a na urýchlenie reakcie na bezpečnostný incident. Pri práci s TTP sa môže využívať MITRE ATT&CK rámec. Tento rámec zachytáva pohľad na správanie útočníkov v reálnom svete. Taktiež poskytuje spôsoby detekcie rôznych techník [8].

1.2.3 Prevádzkové threat intelligence

Prevádzkové threat intelligence je znalosť o kybernetických útokoch, udalostiach a kampaniach [3]. Poskytuje špecializované informácie, ktoré pomáhajú tímom reagujúcim na incidenty pochopiť podstatu, zámer a načasovanie konkrétnych útokov [7]. Prevádzkové threat intelligence obsahuje technické informácie, ako napríklad použitý vektor útoku, aké využívané zraniteľnosti alebo domény boli použité pre riadenie botnetov. Spoločným zdrojom technických informácií sú zdroje (feedy) o bezpečnostných hrozbách, ktoré sa zvyčajne zameriavajú na jeden typ indikátora. Príkladom sú heše malvéru alebo podozrivé domény. Ďalšie zdroje informácií o konkrétnych útokoch môžu pochádzať z uzavretých zdrojov, ako je odpočúvanie komunikácií skupín útočníkov, a to buď infiltráciou do ich skupiny alebo preniknutím do ich komunikačných kanálov. Pri zbieraní týchto typov informácií je niekoľko prekážok [1]:

- **Prístup** - Skupiny útočníkov môžu komunikovať prostredníctvom súkromných a šifrovaných kanálov, alebo vyžadovať nejaký druh overenia totožnosti. Taktiež môže byť problém jazyk, v ktorom komunikujú.
- **Zašumenie** - Môže byť ťažké alebo nemožné ručne zhromaždiť informácie z vysoko objemových zdrojov, ako napríklad chatovacie miestnosti alebo sociálne siete.
- **Obfuskácia** - Aby sa zabránilo odhaleniu, môžu skupiny použiť obfuskáciu. Obfuskácia je zahmlievanie zdrojového kódu tak, aby funkcionalita programu ostala nezmenená, no bol nepochopiteľný pre čitateľa [9].

1.2.4 Technické threat intelligence

Technické threat intelligence sa zameriava na technické stopy, ktoré indikujú hrozbu kybernetickej bezpečnosti. Tento typ je veľmi dôležitý lebo poskytuje predstavu o tom, čo treba pri analýze útokov hľadať [6]. Medzi tieto stopy môžu patriť aj zdroje, ktoré útočník použil pri útoku. Indikátory technického threat intelligence sa zhromažďujú z rôznych kampaní, útokov na iné organizácie, alebo dátových zdrojov (feedov) poskytovaných tretími stranami. Tieto informácie pomáhajú bezpečnostným technikom pridať identifikované indikátory do ochranných systémov ako IPS alebo firewall [7]. IPS

je technológia na zabezpečenie siete a prevencie systému pred narušením, ktorá skúma tok sieťovej prevádzky pre odhalenie a zabránenie zneužitia zraniteľnosti. Ako môžeme vidieť na Obrázku č. 2, technické threat intelligence má krátku životnosť, pretože pre útočníka nie je náročné napríklad pozmeniť malvér tak, aby mal iný heš.

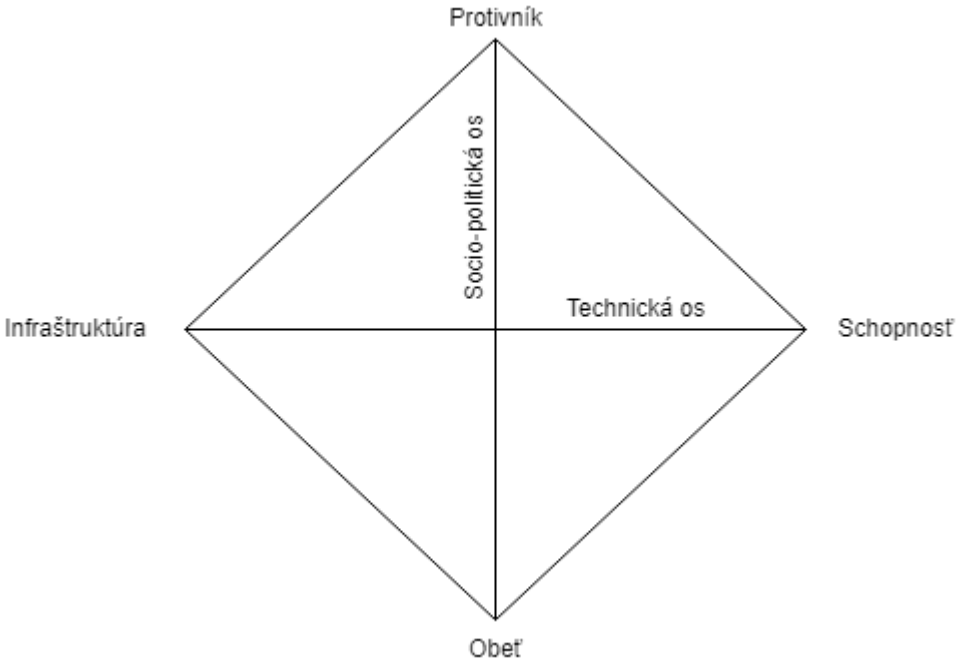
Obr. 2 Typy threat intelligence a ich vplyv (prevzaté z [6])

1.3 Modely používané pri threat intelligence

V oblasti threat intelligence sa využíva niekoľko prístupov na sledovanie a analýzu rôznych charakteristík kybernetických útokov. Tieto prístupy sa vyvinuli do modelov, podľa ktorých sa môžu bezpečnostné tímy a organizácie riadiť. Týchto modelov je viacero, no my si uvedieme tri, a to diamantový model [14], MITRE ATT&CK model [8] a cyber kill chain model [12].

Diamantový model predstavuje koncept analýzy narušenia. Je zložený zo štyroch základných prvkov: protivník, infraštruktúra, schopnosť a obeť [14]. Tento model v najjednoduchšej forme popisuje, že protivník nasadí schopnosť cez nejakú infraštruktúru voči obeti. Tieto aktivity sa volajú udalosti. Tieto prvky sú spojené hranami

tak, ako je vidno na Obrázku č. 3, čo prezentuje ich základné vzťahy. Vodorovná hrana predstavuje technickú os a spája infraštruktúru a schopnosť. To predstavuje technológiu umožňujúcu infraštruktúru a schopnosť prevádzkovať a komunikovať. Zvislá hrana predstavuje socio-politickú os, ktorá spája protivníka a obeť. Tento vzťah funguje na princípe producenta a konzumenta, ktorý je podložený socio-politickými potrebami protivníka. Označuje potrebu protivníka (napr. zisk financií, informácií alebo zvýšenie povedomia v komunite) a schopnosť obete uspokojiť potreby definujúce zámer protivníka napr. formou špionáže. Obeť poskytne produkt (napr. výpočtové zdroje) a protivník spotrebuje tento produkt [11]. Prechádzaním cez vrcholy odhaľujú analytici viac informácií o protivníkových operáciách, schopnostiach, infraštruktúrach a obetiach. Ďalej definuje meta-funkcie na podporu konštrukcií vyššej úrovne, aby sa mohla aplikovať merateľnosť, testovateľnosť a opakovateľnosť pre komplexnejšiu metódu analýzy. Tieto meta-funkcie môžu byť časová pečiatka, fáza, výsledok, metodológia alebo zdroj [11].



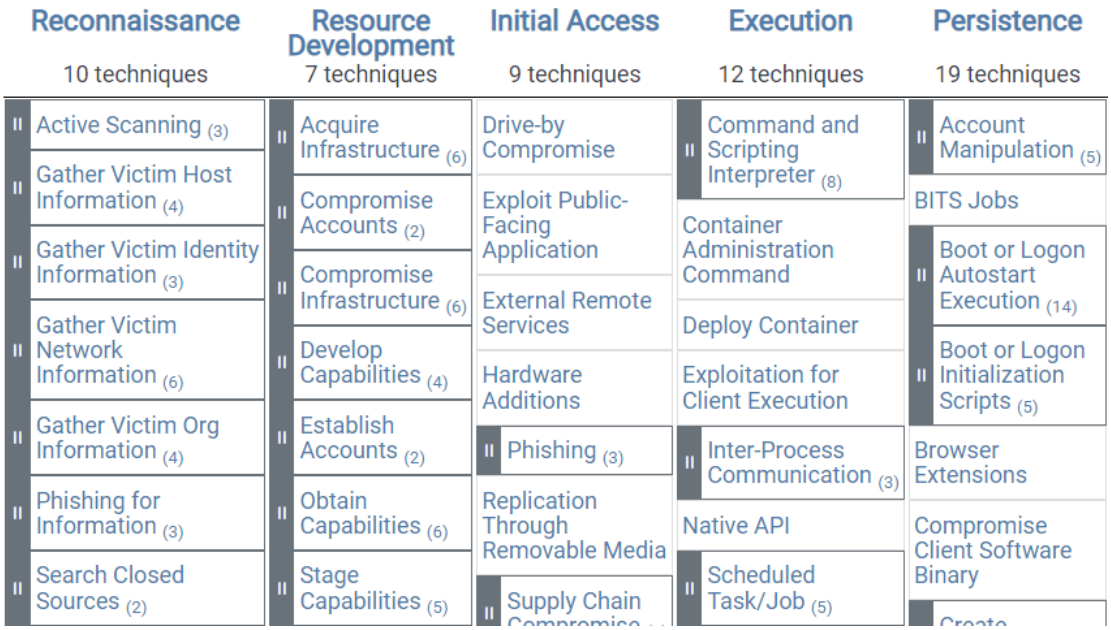
Obr. 3 Diamantový model (prevzaté z [11])

MITRE ATT&CK rámec, ktorého časť môžeme vidieť na Obrázku č. 4, je celosvetovo dostupná vedomostná základňa o taktikách a technikách protivníkov, ktorá je založená na pozorovaniach v reálnom svete. Znalostná základňa tohto rámcu sa

používa ako základ pre vývoj špecifických modelov a metodológií bezpečnostných hrozieb. Skladá sa z nasledujúcich základných komponentov [8]:

- taktika označujúca krátkodobé, taktické ciele protivníka počas útoku,
- techniky popisujúce prostriedky, ktorými protivníci dosahujú taktické ciele a
- zdokumentované použitie techník protivníka a iné metadáta.

MITRE ATT&CK bol vytvorený v roku 2013 ako výsledok experimentu, kde výskumníci napodobňovali správanie protivníka aj obrancu v snahe zlepšiť post-kompromisnú detekciu hrozieb.



Obr. 4 Výrez z MITRE ATT&CK rámca (prevzaté z [8])

Cyber kill chain je model kybernetickej bezpečnosti, ktorý bol prevzatý z armády a súvisí so štruktúrou útoku. Hlavnou myšlienkou je sledovanie fázy útoku, identifikuje zraniteľné miesta a pomáha zastaviť útok v každej fáze reťazca [14]. Pozostáva zo siedmich na seba nadväzujúcich krokov, tak ako ich môžeme vidieť na Obrázku č. 5 [12]:

- **Prieskum** – Útočník zhromažďuje údaje o cieli a taktike útoku. To zahŕňa skenovanie systému obete pre nájdenie zraniteľností.

- **Zbrojenie** – Útočníci vyvíjajú malvér využívaním zraniteľností, ktoré našli v prvom kroku.
- **Doručenie** – Útočník doručuje malvér do systému obete. Medzi najbežnejšie doručovacie metódy patrí podvodné e-maily alebo podvodné stránky. Toto je najdôležitejšia fáza, v ktorej sa dá zastaviť útok.
- **Exploitácia** – Po tom čo je malvér doručený do systému obete, útočník dostáva príležitosť exploitovať systémy organizácie inštalovaním nástrojov alebo spúšťaním skriptov.
- **Inštalácia** – Pomocou malvéru sú nainštalované ”zadné vrátka” alebo trojský kôň so vzdialeným prístupom. Toto je ďalšia fáza, v ktorej môže byť útok zastavený pomocou systémov ako je HIPS (Host-based Intrusion Prevention System).
- **Velenie a riadenie** – Útočník získa kontrolu nad systémom a sieťou organizácie. Získa tým prístup ku účtom s vyššími oprávneniami a môže sa pokúšať získavať prihlasovacie údaje a meniť oprávnenia aby mohol prevziať kontrolu.
- **Akcie na ciele** – Útočník nakoniec plní svoje ciele, čo môže byť zhromažďovanie dát, šifrovanie a extrakcia dôverných informácií z organizácie [13].

Obr. 5 Znázornenie cyber kill chain modelu (prevzaté z [12])

Tieto kroky sú rozdelené do prípravnej fázy a útočnej fázy. V prípravnej fáze je prieskum a zbrojenie a v útočnej fáze sú všetky nasledujúce kroky.

1.4 Prípady použitia threat intelligence

V rámci bezpečnosti organizácii predstavuje pre bezpečnostné tímy threat intelligence základný zdroj informácií. Taktiež je užitočnou súčasťou triedenia, analýzy rizík, riadenia zraniteľností a rozhodnutí so širokým dopadom. V tejto časti si uvedieme prípady, v ktorých sa využíva threat intelligence.

1.4.1 Incident Response

Pri riešení bezpečnostných udalostí musia analytici tráviť veľa času dôkladným manuálnym triedením údajov, aby mohli správne vyhodnotiť problém. Threat intelligence to dokáže zjednodušiť viacerými spôsobmi, ako napríklad automatickou identifikáciou a vylúčením údajov s nesprávnou pozitivitou (false positives), obohatením upozornení o kontext v reálnom čase. Napríklad obohatenie o vlastnú hodnotu rizika alebo porovnaním informácií z interných a externých zdrojov [15].

1.4.2 Security Operations Center

Väčšina tímov bezpečnostného operačného centra (Security Operations Center - SOC) musí riešiť obrovské množstvá varovaní generovaných zariadeniami v rámci počítačových sietí, ktoré monitorujú. Triedenie týchto upozornení trvá príliš dlho a mnohé z nich nikdy nie sú vyšetrené. „Únava z varovania“ vedie analytikov k tomu, aby prijímali varovania menej vážne, ako by mali. Threat intelligence môže pomôcť vyriešiť tieto problémy. Pomáha rýchlejšie a presnejšie zhromažďovať informácie o hrozbách, odfiltrovať falošné poplachy, urýchliť triedenie a zjednodušiť analýzu incidentov [16].

1.4.3 Správa bezpečnostných zraniteľnosti

Aj keď počet bezpečnostných zraniteľností a bezpečnostných hrozieb každým rokom stúpa, výskumy ukazujú, že väčšina ohrození sa zameriava na rovnakú malú časť zraniteľností. Útočníci sú tiež rýchlejší v spôsobe kompromitácie organizácie, resp. exfiltrácie údajov. V súčasnosti trvá v priemere iba pätnásť dní medzi oznámením novej chyby zabezpečenia a nástrojom, ktorý ju vie využiť (exploitom) [1]. To znamená, že organizácie majú k dispozícii dva týždne k oprave svojich systémov. Ak nová

zraniteľnosť nebude zneužitá do dvoch týždňov až troch mesiacov, je nepravdepodobné, že by niekedy bola zneužitá.

1.4.4 Analýza rizík

Modelovanie rizík môže byť pre organizácie užitočným spôsobom na stanovenie investičných priorít. Mnoho modelov rizík trpí nejasnosťami, nekvantifikovaným výstupom, ktorý je narýchlo vytvorený, na základe nekompletných informácií alebo na základe neopodstatnených predpokladov. Threat intelligence poskytuje kontext, ktorý pomáha rizikovým modelom definovať mieru rizika a zvyšovať transparentnosť ich predpokladov, premenných a výsledkov [15].

1.4.5 Prevencia pred podvodmi

Na zaistenie bezpečnosti organizácie, nestačí iba detegovať a reagovať na bezpečnostné hrozby, ktoré sa už nachádzajú v systéme, je potrebné zabrániť podvodnému použitiu vašich údajov alebo značky. Údaje a poznatky zhromaždené z kriminálnych komunít poskytujú prehľad o motiváciách, metódach a taktike útočníkov, najmä ak súvisia s informáciami z webu vrátane technických zdrojov (feedov) a indikátorov kompromitácie [17].

2 Indikátory kompromitácie

V predchádzajúcej kapitole sme sa zamerali na threat intelligence ako celok. V tejto kapitole sa bližšie pozrieme na indikátory kompromitácie, ktoré sú dôležitou časťou threat intelligence a súčasne sú veľmi dôležitým zdrojom informácií pri riešení bezpečnostných incidentov.

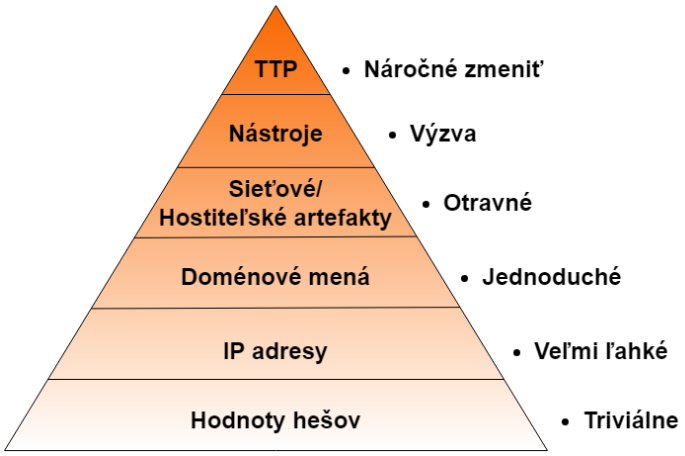
Indikátory kompromitácie (Indicators of Compromise, IoC) sú forenzné artefakty pozorované v rámci počítačovej siete alebo v operačnom systéme, ktoré s vysokou istotou naznačujú narušenie daného zariadenia alebo siete. Hlavnou úlohou IoC je identifikovať nechcenú činnosť v prostredí a identifikovať dôkazy, ktoré podporujú evidenciu anomálie v systéme [18]. Niektoré z týchto artefaktov sa nachádzajú v záznamoch (logoch) a položkách s časovou pečiatkou, iné v aplikáciách a službách. Indikátory kompromitácie sú dôležitou súčasťou pre zdieľanie informácií o bezpečnostných hrozbách medzi organizáciami, ak bola narušená ich bezpečnosť. To pomáha organizáciám získať podrobnú analýzu toho, ako došlo k útokom, čo im predchádzalo a ako sa podobným útokom môžu brániť. Taktiež im to pomôže vylepšiť ich stratégie na riešení a odpovede na bezpečnostné incidenty [19].

Ako príklady indikátorov kompromitácie môžeme uviesť IP adresy a porty, MD5 a SHA heše súborov, URL adresy, podozrivé cesty ku adresárom, zvýšené Read/Write operácie v databáze, podozrivé DNS požiadavky, náznaky DDoS útokov, zmeny hodnôt v registroch, anomálie v aktivitách prihlásených užívateľov, anomálie v logoch, geografická poloha alebo neobvyklá veľkosť HTML odpovede pri webovom serveri [20].

2.1 Pyramída bolesti

Celá pointa za detegovaním indikátora kompromitácie je vytvoriť ťažkosti pre útočníka tým, že sa aplikujú do ochranných systémov. Vzťah medzi IoC a ťažkosťami, s ktorými sa útočník musí vyrovnať keď tento indikátor nemá, je znázornený pyramídou bolesti. Koncept pyramídy bolesti je spôsob, ako vylepšiť aplikovateľnosť IoC.

Pyramída bolesti predstavuje šesť typov indikátorov kompromitácie usporiadaných vzostupne podľa dopadu na útočníka. V každej úrovni sú znázornené rôzne typy IoC, ktoré sa dajú použiť na detekciu aktivít [23]. Sú zoradené podľa toho, koľko bolesti, resp. ťažkostí, spôsobia útočníkovi keď ich nemôže použiť [24].



Obr. 6 Znázornenie pyramídy bolesti (prevzaté z [24]).

Ako môžeme vidieť na Obrázku č. 6, usporiadanie indikátorov v pyramíde je od najmenej bolestivých až po najviac bolestivé pre útočníka.

Hodnoty hešov - Kryptografické heše súborov (MD5, SHA1, SHA256) sú najčastejším IoC používaným v antimalvérových riešeniach [23]. Na jednej strane sú heše najpresnejším typom IoC. No na druhú stranu, stačí ľubovoľná zmena v súbore (zmena jedného bitu, pridanie "null" na koniec) a výsledný heš má kompletne inú hodnotu [24]. Z tohto dôvodu je pre útočníka **triviálne** pozmeniť súbor tak, aby bol heš rozdielny v prípade, ak sú heše jeho súborov odhalené.

IP adresy - Napriek tomu, že IP adresy sú jednými z najčastejších indikátorom útoku, takmer žiadny útočník nepoužíva svoju vlastnú IP adresu. Útočníci používajú Tor, VPN (Virtual Private Network) alebo proxy na pravidelné menenie IP adries [23]. V iných prípadoch sa stretávame s tým, že je zneužité iné zariadenie, ktoré útočník kontroluje a prostredníctvom neho vykonáva prieskumné útoky, distribúciu malvéru a pod. Aj napriek tomu ak je útočníkova IP adresa zablokovaná, je pre neho **veľmi ľahké** si ju zmeniť a aj napriek tomu sa dostať do systému.

Doménové mená - Doménové mená sú o niečo zložitejšie na zmenu, kvôli práci navyše, ktorú musí útočník vykonať pre zmenu [24]. Napriek tomu, že doménové mená musia byť registrované, platené a niekde hostované, existujú služby, pomocou ktorých ich môžu útočníci automaticky meniť cez API [23]. To znamená, že v pyramíde bolesti to je stále označené ako **jednoduché**.

Sieťové/Hostiteľské artefakty - Ak vieme identifikovať a reagovať na indikátory na tejto úrovni, môže to mať negatívny vplyv na útočníka. Napríklad môžeme zistiť, že útočníkov http recon tool používa ako meno používateľa konkrétny reťazec pri prehľadávaní webového obsahu. Ak zablokujeme všetky požiadavky od tohto užívateľa, tak útočník musí zistiť, ako sme detegovali jeho nástroj a opraviť to [24]. To môže byť pre útočníka **otravné**, a preto to je v pyramíde bolesti vyššie.

Nástroje – Útočníci stále modernizujú svoje nástroje, ktoré sú stavané na to, aby skenovali systém a hľadali v ňom chyby, vedeli tam nasadiť škodlivý kód alebo získať ďalšie údaje [23]. Ak však dokážeme detegovať artefakty, ktoré tieto nástroje zanechajú, donúti to útočníkov používať nové nástroje [24]. To sú pre nich veľké **výzvy** z dôvodu financií a času, ktorý musia dať do nájdenia nového nástroja (vyrobenie nového) a učenia sa ako pracovať s daným nástrojom.

TTP – Taktiky, techniky a procedúry sú v podstate útočníkova metodológia. Taktiky sú typy činností, ktoré útočníci používajú na uskutočnenie útoku. Napríklad získanie neoprávneného prístupu ku citlivým dátam. Techniky sú všeobecné metódy, ktoré útočníci využívajú. Napríklad pri kompromitácii webového sídla môže byť použitá technika SQL injection. Každá taktika môže obsahovať niekoľko techník. Procedúry sú špecifický rad krokov, ktoré môžu útočníci využiť pri útoku. Napríklad postup pri SQL injection môže zahŕňať skenovanie zraniteľností, napísanie SQL dopytu so škodlivým kódom a následne zaslanie ho do neochráneného formulára. Správanie útočníka dokáže pomôcť bezpečnostným tímom vyšetriť a reagovať na útok [23]. Ak vieme reagovať na útočníkove TTP dostatočne rýchlo, donútime ich zmeniť kompletný prístup k útoku, a teda naučiť sa nové správanie [24], čo je pre útočníka **veľmi náročné**.

2.2 Typy indikátorov kompromitácie

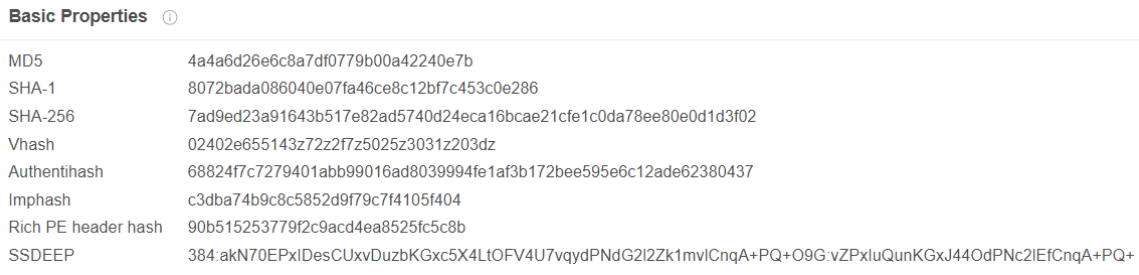
Indikátory kompromitácie sa rozdeľujú na tri základné typy: atomické, vypočítané a behaviorálne. Základnou myšlienkou tohto delenia je rozdeliť ich do kategórií podľa jednoduchosti. Atomické IoC sú samé o sebe indikátory, a teda sú najjednoduchšie. Vypočítané IoC musia prejsť nejakým procesom (napr. vypočítaním hešovacej funkcie), aby sa z nich stali IoC. Najzložitejšie sú behaviorálne, pretože sa môžu skladať z atomických, vypočítaných a správaním útočníka pri útoku.

2.2.1 Atomické indikátory kompromitácie

Atomické indikátory kompromitácie sú také fragmenty údajov, ktoré sa už nedajú ďalej členiť na menšie prvky. Medzi atomické indikátory kompromitácie môže patriť meno hostiteľa, IP adresa, e-mailová adresa, názov procesu, súboru alebo textový reťazec [26]. Teda tento fragment sám o sebe je typom informácie, ktorá bola zaznamenaná pri narušení systému a teda naznačuje aktivitu útočníka [25]. Ako príklad môžeme uviesť, ak pri phishingovej kampani nájdeme e-mailovú adresu z ktorej podvodný mail prišiel. Ďalším príkladom môže byť názov procesu, ktorý bol spustený v nejakom čase, kedy vieme, že mal útočník prístup k systému.

2.2.2 Vypočítané indikátory kompromitácie

Vypočítané indikátory kompromitácie sú fragmenty údajov vypočítaných špecifickým spôsobom za účelom útoku na systém. Sú vyvinuté z materiálu týkajúceho sa incidentu [25]. Do tejto kategórie spadajú heše malvéru alebo štatistické regulárne výrazy [26]. Ako príklad uvedieme malvér Spora [27]. Na Obrázku č. 7 môžeme vidieť vypočítané indikátory kompromitácie, ako sú MD5, SHA-1, SHA-256 heše alebo SSDEEP. SSDEEP je fuzzy hešovací algoritmus, ktorý vytvára heš súboru, ktorý sa pokúša zistiť úroveň podobnosti dvoch súborov [28]. Teda ak sú dva súbory dostatočne podobné, ich SSDEEP heš bude taktiež podobný. Tieto hodnoty sa dajú ďalej použiť napríklad v ochranných systémoch.



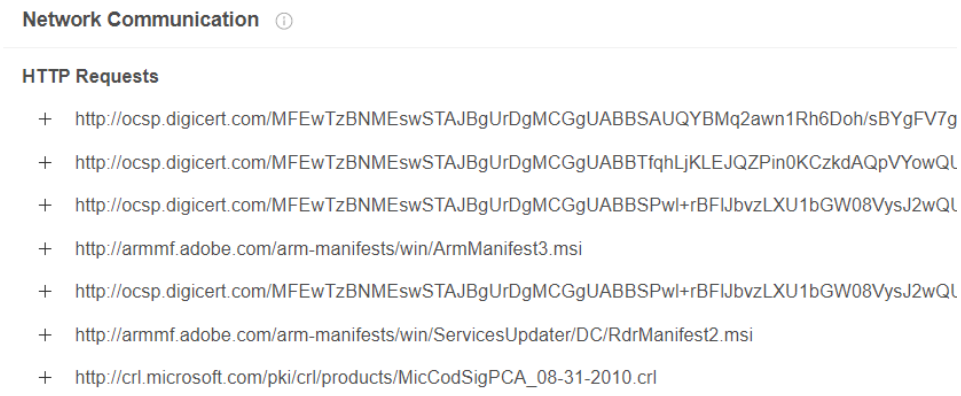
Basic Properties

MD5	4a4a6d26e6c8a7df0779b00a42240e7b
SHA-1	8072bada086040e07fa46ce8c12bf7c453c0e286
SHA-256	7ad9ed23a91643b517e82ad5740d24eca16bcae21cfe1c0da78ee80e0d1d3f02
Vhash	02402e655143z72z2f7z5025z3031z203dz
Authentihash	68824f7c7279401abb99016ad8039994fe1af3b172bee595e6c12ade62380437
Imphash	c3dba74b9c8c5852d9f79c7f4105f404
Rich PE header hash	90b515253779f2c9acd4ea8525fc5c8b
SSDEEP	384:akN70EPxlDesCUxvDuzbKGxc5X4LtOFV4U7vqydPNdG2l2Zk1mvlCnqA+PQ+O9G:vZPxluQunKGxJ44OdPNc2lEfCnqA+PQ+

Obr. 7 Príklad vypočítaných indikátorov kompromitácie malvéru Spora (prevzaté z [29])

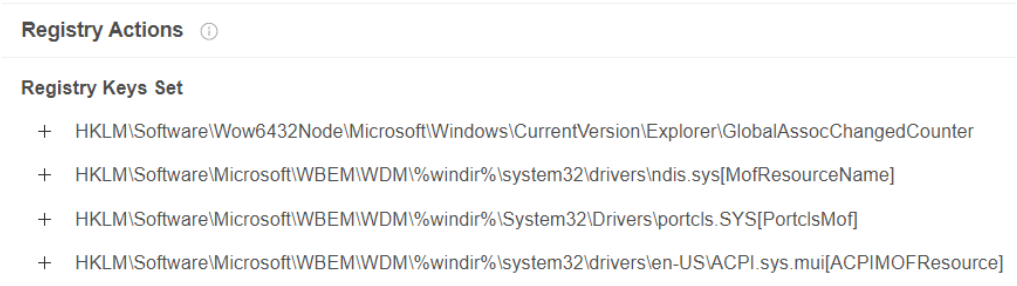
2.2.3 Behaviorálne indikátory kompromitácie

Behaviorálne indikátory kompromitácie sú kombináciou atomických a vypočítaných indikátorov kompromitácie a správaním sa útočníka [25]. Môžu pozostávať z viacerých atomických alebo vypočítaných indikátorov kompromitácie alebo kombináciou týchto dvoch typov. Atomické a vypočítané IoC boli použité ako súčasť narušenia systému, čo v skutočnosti znamená akýsi podpis útoku [26]. Príkladom môže byť už spomínaný malvér Spora. Na Obrázku č. 8 môžeme HTTP požiadavky, ktoré tento malvér vytvoril.



Obr. 8 HTTP request požiadavka (prevzaté z [29])

Ďalším príkladom behaviorálneho IoC sú zmeny v registri operačného systému Wundows. Výpis registra, pri ktorých nastala zmena môžeme vidieť na Obrázku č. 9.



Obr. 9 Zmeny v registroch (prevzaté z [29])

V tomto prípade posledným príkladom je spustený shell príkaz, ktorý je na Obrázku č. 10. Príkaz v tomto príklade znamená, že útočník spúšťa nástroj vssadmin na zmazanie volume shadow copy.

Shell Commands

%SAMPLEPATH%

"%windir%\System32\wbem\WMIC.exe" process call create "cmd.exe /c vssadmin.exe delete shadows /all /quiet & bcdedit.exe /set {default} recoveryenabled no & bcdedit.exe /set {default} bootstatuspolicy ignoreallfailures"

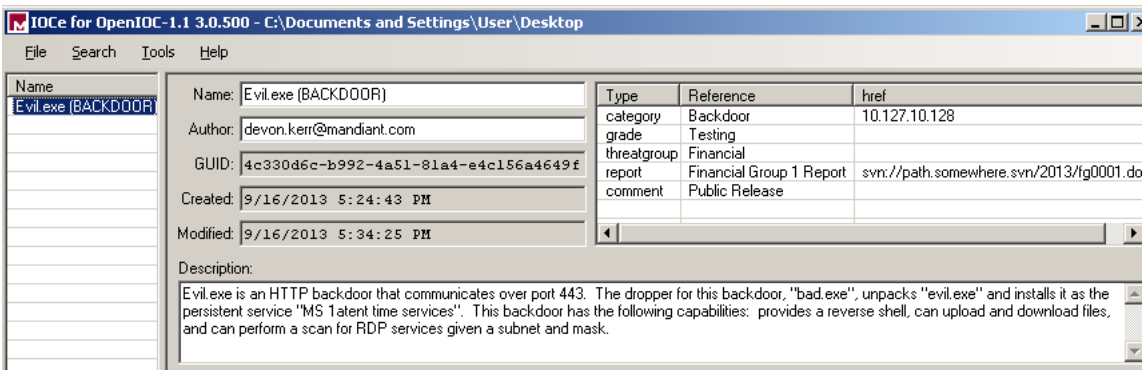
Obr. 10 Spustený príkaz v shell (prevzaté z [29])

Všetky tieto indikátory kompromitácie spolu tvoria behaviorálne indikátory kompromitácie, podľa ktorých vieme určiť konkrétne správanie útočníka.

2.3 Komponenty indikátorov kompromitácie

Indikátory kompromitácie sa skladajú z troch hlavných komponentov. Tieto komponenty sú rozdelené na metadáta, referencie a definície. V tejto časti si ukážeme, ako vyzerajú tieto komponenty v programe IOC Editor od spoločnosti FireEye [30].

Metadáta popisujú informácie o IoC [31]. Obrázok Obr. 11 je z programu IOC Editor, v ktorom sú zobrazené metadáta konkrétneho IoC, ako napríklad autor a názov IoC, dátum a čas kedy bol tento záznam vytvorený alebo modifikovaný a stručný popis.



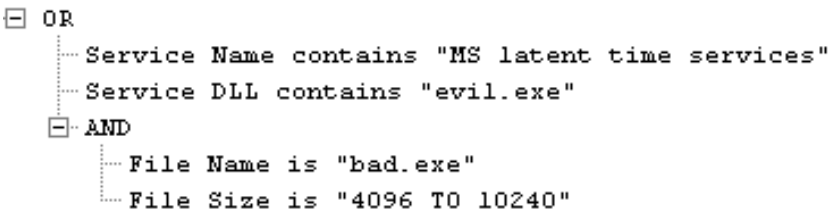
Obr. 11 Ukážka metadát indikátora kompromitácie v programe IOC Editor (prevzaté z [31])

Ďalším komponentom sú **referencie**. Referencie sa môžu využívať na asociáciu s konkrétnou skupinou ohrozenia. Medzi referenciami vieme nájsť informácie ako názov vyšetrovania, číslo prípadu alebo rôzne komentáre k danému IoC [31]. Na Obrázku č. 12 sú zobrazené ďalšie referencie, a to kategória, typ IoC, skupina ohrozenia, report a ďalšie komentáre.

Type	Reference	href
category	Backdoor	10.127.10.128
grade	Testing	
threatgroup	Financial	
report	Financial Group 1 Report	svn://path.somewhere.svn/2013/fg0001.docx
comment	Public Release	

Obr. 12 Ukážka referencií indikátora kompromitácie v programe IOC Editor (prevzaté z [31])

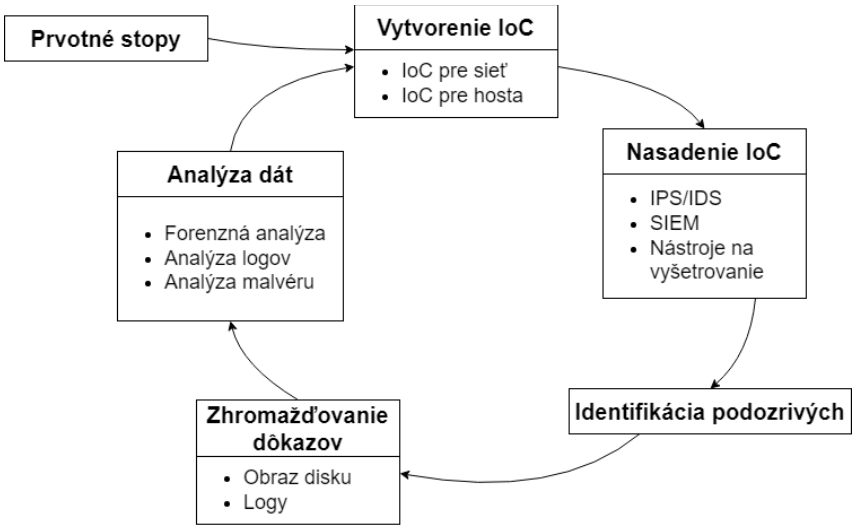
Tretím komponentom sú **definície**. Definície obsahujú forenzné artefakty, ktoré sa vyšetrovateľ rozhodol zakomponovať do daného IoC. V definíciách sú konkrétne ukazovatele, ktoré môžu byť kombinované z dvoch výrazov spojených pomocou výrazov booleovskej logiky [31]. Na Obrázku č. 13 je uvedený príklad definícií IoC. V tomto príklade to sú pravidlá pre službu, ktorá má názov “MS 1atent time services“ alebo názov ServiceDLL je “evil.exe“ alebo je názov súboru “bad.exe“ a má veľkosť v rozhraní od 4096 do 10240 bajtov.



Obr. 13 Ukážka definícií indikátora kompromitácie (prevzaté z [31])

2.4 Životný cyklus indikátorov kompromitácie

Tak ako threat intelligence má svoj životný cyklus, tak isto majú aj indikátory kompromitácie svoj životný cyklus, ktorý sa neustále opakuje. Životný cyklus vyšetrovania sleduje IoC a hľadá konkrétne položky v systéme napr. v registroch operačného systému Windows. Tieto registre poskytujú perfektnú platformu na generovanie a udržiavanie indikátorov kompromitácie, čo priamo pomáha pri forenznej analýze [18]. Životný cyklus IoC sa delí do šiestich fáz. Ako môžeme vidieť na Obrázku č. 14, tieto fázy na seba priamo nadväzujú.



Obr. 14 Životný cyklus indikátorov kompromitácie (prevzaté z [18])

- **Prvotné stopy** – Keď sa eviduje stopy o kompromitácii, incident respondenti preskúmajú a identifikujú konkrétne riešenie, ktoré je konkrétnym forenzným artefaktom.
- **Vytvorenie IoC pre hosta a sieť** – Po zozbieraní údajov sa vytvorí IoC napr. z registrov v operačnom systéme Windows.
- **Nasadenie IoC v spoločnosti** – Po vytvorení špecifického IoC sa nasadí do rôznych investigatívnych technológií napr. IPS/IDS, SIEM.
- **Identifikácia podozrivých aktivít a udalostí** – Nasadenie IoC pomáha pri identifikácii podozrivých aktivít a udalostí.

- **Zhromažďovanie a analýza stôp** – stopy o podozrivých udalostiach sa zbierajú napr. z logov alebo forenzného obrazu disku a adekvátne sa analyzujú.
- **Úprava a vytvorenie nových IoC** – Na základe nových stôp a dát nájdených pri vyšetrovaní môže vyšetrujúci tím vytvoriť nové IoC [18].

2.5 Prípady použitia indikátorov kompromitácie

Existuje niekoľko prípadov využitia na použitie indikátorov kompromitácie, ktoré sa spájajú s cieľmi analytika, resp. vyšetrovateľa.

Ako najbežnejší prípad využitia môžeme uviesť **identifikáciu malvéru** a potvrdenie, resp. vyvrátenie hypotézy o kompromitácii skúmaného zariadenia. Ide o IoC vytvorené na nájdenie určitého typu známeho škodlivého softvéru. Môže to vyhľadávať podľa atribútov binárneho súboru, súboru samotného alebo artefaktu vytvoreného pri spustení tohto súboru, ako napríklad kľúče registrov [31].

Ďalším prípadom využitia je **metodika**. Na rozdiel od IoC vytvoreného na hľadanie škodlivého softvéru, môžu tieto IoC nájsť veci, o ktorých nemusíme nevyhnutne vedieť [31]. Napríklad ak chceme identifikovať dll súbor, ktorý nebol podpísaný a bol načítaný z adresára, ktorý neobsahuje "windows/system32". Ďalším príkladom metodického IoC môže byť IoC, ktoré vyhľadáva reťazec končiaci ".jpg" v textových hodnotách registrov, ktoré obsahujú kľúč "Run". To ukazuje na podozrivé veci, ktoré po vyšetrení môžu viesť ku dôkazu o narušení systému.

Iným príkladom použitia sú IoC, ktoré **hromadia veľa artefaktov**. Ide o IoC, ktoré hromadia z threat intelligence kanálov MD5 heše súborov alebo podozrivé IP adresy. Tieto druhy IoC sú zvyčajne vhodné iba na presnú zhodu [31].

Zaujímavým prípadom využitia sú **investigatívne** IoC. Tieto IoC identifikujú digitálne stopy o škodlivej aktivite ako napríklad metadáta súvisiace s inštaláciou „backdoorov“ alebo súboroch pripravených na exfiltráciu. Sú podobné ako hromadné IoC, avšak investigatívne indikátory obsahujú artefakty iba z jedného vyšetrovania a môžu pomôcť pri prioritizácii, ktorý systém chceme analyzovať skôr [31].

2.6 Spôsoby ukladania a zdieľania indikátorov kompromitácie

Táto kapitola nadväzuje na definíciu a komponenty jednotlivých IoC. Keďže vo svete je mnoho organizácií, ktoré sa zaoberajú bezpečnosťou, každá si môže ukladať IoC vlastným spôsobom. Celosvetovo je niekoľko uznávaných formátov, akými sa IoC ukladajú a následne zdieľajú. My si uvedieme OpenIOC [33], STIX [34] a CybOX [35].

2.6.1 OpenIOC

Jedným z najpoužívanejších rozhraní je **OpenIOC** [33]. Predstavuje rozhranie na popis indikátorov kompromitácie vyvinutý spoločnosťou Mandiant v roku 2011. Je založený na XML formáte, takže je ľahko spracovateľný strojom a zároveň je pochopiteľný aj pre ľudí. Sám o sebe predstavuje iba definície forenzných artefaktov zanechaných v systéme po narušení. Na praktické využitie je potrebné použiť aplikácie, ktoré prehľadajú systém a snažia sa nájsť indikátory, ktoré sú zhodné s definíciami.

Príklad OpenIOC súboru je uvedený na Obrázku č. 15. Tento indikátor kompromitácie napísaný v rozhraní OpenIOC slúži na vyhľadávanie súborov s príponou exe, dll a rar.

```
<?xml version="1.0" encoding="us-ascii"?>
<ioc xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
id="88e454e9-f94d-4771-baf8-14fc625ea4e4"
last-modified="2014-08-06T06:52:49"
xmlns="http://schemas.mandiant.com/2010/ioc">
    <short_description>*New Unsaved Indicator*
    </short_description>
    <authored_date>2014-08-05T06:35:39</authored_date>
    <links /><ioc>
  <definition>
    <Indicator operator="AND">
      <Indicator operator="OR">
        <IndicatorItem condition="contains">
          <Context document="FileItem"
                   search="FileItem/FileExtension"/>
          <Content type="string">.exe</Content>
        </IndicatorItem>
        <IndicatorItem condition="contains">
          <Context document="FileItem"
                   search="FileItem/FileExtension"/>
          <Content type="string">.dll</Content>
        </IndicatorItem>
        <IndicatorItem condition="contains">
          <Context document="FileItem"
                   search="FileItem/FileExtension"/>
          <Content type="string">.rar</Content>
        </IndicatorItem>
        <Indicator operator="OR">
          <IndicatorItem condition="contains">
            <Context document="FileItem"
                     search="FileItem/FullPath"/>
            <Content type="string">Recycler</Content>
          </IndicatorItem>
          <IndicatorItem condition="contains">
            <Context document="FileItem"
                     search="FileItem/FullPath"/>
```

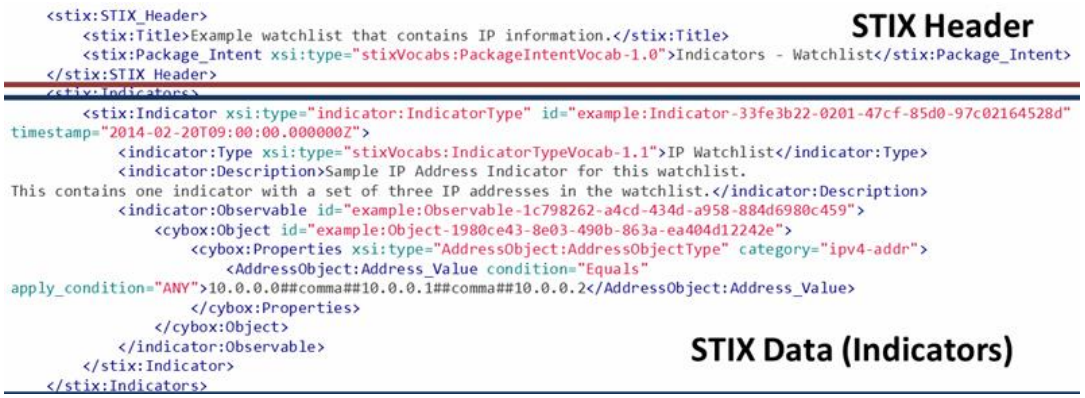
Obr. 15 Príklad OpenIOC indikátora kompromitácie (prevzaté z [33])

2.6.2 STIX

Structured Threat Information eXpression (STIX) je jednotný jazykový a serializačný formát na výmenu informácií o bezpečnostných hrozbách [34]. Mnoho bezpečnostných systémov môže importovať informácie o bezpečnostných hrozbách práve pomocou STIX formátu. STIX súbor musí mať pri sebe aj rozšírenie vo formáte xml aby mohlo byť čítané a parsované [36]. Pre systémy, ktoré nepodporujú OpenIOC formát je vytvorená Pythonová knižnica "openioc-to-stix", ktorá dokáže konvertovať tento formát na STIX formát [37]. Na Obrázku č. 16 môžeme vidieť xml časť STIX súboru a na Obrázku č. 17 je zobrazená hlavička STIX časti a dátovú časť.


```
<stix:STIX_Package
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:stix="http://stix.mitre.org/stix-1"
    xmlns:indicator="http://stix.mitre.org/Indicator-2"
    xmlns:cybox="http://cybox.mitre.org/cybox-2"
    xmlns:AddressObject="http://cybox.mitre.org/objects#AddressObject-2"
    xmlns:cyboxVocabs="http://cybox.mitre.org/default_vocabularies-2"
    xmlns:stixVocabs="http://stix.mitre.org/default_vocabularies-1"
    xmlns:example="http://example.com/"
    xsi:schemaLocation="
http://stix.mitre.org/stix-1 ../stix_core.xsd
http://stix.mitre.org/Indicator-2 ../indicator.xsd
http://cybox.mitre.org/default_vocabularies-2 ../cybox/cybox_default_vocabularies.xsd
http://stix.mitre.org/default_vocabularies-1 ../stix_default_vocabularies.xsd
http://cybox.mitre.org/objects#AddressObject-2 ../cybox/objects/Address_Object.xsd"
```

Obr. 16 xml časť STIX súboru (prevzaté z [38])




```
    <stix:STIX_Header>
        <stix:Title>Example watchlist that contains IP information.</stix:Title>
        <stix:Package_Intent xsi:type="stixVocabs:PackageIntentVocab-1.0">Indicators - Watchlist</stix:Package_Intent>
    </stix:STIX_Header>
    <stix:Indicators>
        <stix:Indicator xsi:type="indicator:IndicatorType" id="example:Indicator-33fe3b22-0201-47cf-85d0-97c02164528d"
timestamp="2014-02-20T09:00:00.000000Z">
            <indicator:Type xsi:type="stixVocabs:IndicatorTypeVocab-1.1">IP Watchlist</indicator:Type>
            <indicator:Description>Sample IP Address Indicator for this watchlist.
This contains one indicator with a set of three IP addresses in the watchlist.</indicator:Description>
            <indicator:Observable id="example:Observable-1c798262-a4cd-434d-a958-884d6980c459">
                <cybox:Object id="example:Object-1980ce43-8e03-490b-863a-ea404d12242e">
                    <cybox:Properties xsi:type="AddressObject:AddressObjectType" category="ipv4-addr">
                        <AddressObject:Address_Value condition="Equals"
apply_condition="ANY">10.0.0.0##comma##10.0.0.1##comma##10.0.0.2</AddressObject:Address_Value>
                    </cybox:Properties>
                </cybox:Object>
            </indicator:Observable>
        </stix:Indicator>
    </stix:Indicators>
```

Obr. 17 hlavička a dátová časť STIX súboru (prevzaté z [38])

2.6.3 CybOX

Cyber Observable eXpression (CybOX) je štandardizovaný jazyk na kódovanie a vymieňanie verných informácií o kybernetických pozorovateľných objektoch. Pri vyššie spomínaných OpenIOC a STIX išlo o ukladanie a zdieľanie výlučne indikátorov kompromitácie. CybOX sa zameriava na viacero použití a to napríklad ohodnotenie a charakterizácia hrozieb, charakterizácia malvérov, logovanie, povedomie o kybernetickej bezpečnosti, reakciu na incidenty a samozrejme aj ukladanie a zdieľanie indikátorov kompromitácie. Je tiež dostatočne flexibilný na to aby umožňoval vysoko presný popis kybernetických pozorovateľných objektov a zároveň dokázal sledovať aj abstraktnejšie vzory pre pozorovateľné objekty. Pomocou CybOX je možné zachytiť a zdieľať relevantné údaje, definovať ich v IoC a pravidlách. Taktiež aj reakcia na incidenty a ich riadenie ich môže využívať na zlepšenie detekcie, prevencie a reakcie na útoky [35]. Ako príklad si môžeme uviesť na Obrázku č. 18 uloženie indikátora kompromitácie, konkrétne IP adresy.

```
18 lines (18 sloc)  1.04 KB
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<cybox:Observables xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:cybox="http://docs.oasis-open.org/cti/ns/cybox/core-2"
    xmlns:cyboxCommon="http://docs.oasis-open.org/cti/ns/cybox/common-2"
    xmlns:AddressObj="http://docs.oasis-open.org/cti/ns/cybox/objects/address-2"
    xmlns:example="http://example.com/"
    xsi:schemaLocation="
    http://docs.oasis-open.org/cti/ns/cybox/core-2 ../core.xsd
    http://docs.oasis-open.org/cti/ns/cybox/objects/address-2 ../objects/Address_Object.xsd"
    cybox_major_version="2" cybox_minor_version="1" cybox_update_version="1">
    <cybox:Observable id="example:Observable-0b9af309-0d5a-4c44-bdd7-aea3d99f13b6">
        <cybox:Object id="example:Object-15be6630-c2df-4bf9-8750-3f45ca9e19cf">
            <cybox:Properties xsi:type="AddressObj:AddressObjectType" category="ipv4-addr">
                <AddressObj:Address_Value>192.168.0.5</AddressObj:Address_Value>
            </cybox:Properties>
        </cybox:Object>
    </cybox:Observable>
</cybox:Observables>
```

Obr. 18 Príklad indikátora kompromitácie (IP) v jazyku CybOX (prevzaté z [39])

3 Porovnanie nástrojov

V predchádzajúcich kapitolách sme sa venovali teoretickejšej časti, ktorá je dôležitá pre porozumenie základných pojmov a problémov threat intelligence a indikátorov kompromitácie. V tejto kapitole budeme porovnávať nástroje, ktoré sa zaoberajú indikátormi kompromitácie alebo celkovo threat intelligence.

Táto kapitola je rozdelená do troch podkapitol. V prvej podkapitole porovnáme extraktory indikátorov kompromitácie. Extraktory fungujú ako parsery, teda zo vstupu vyextrahujú potencionálne indikátory kompromitácie. V druhej podkapitole sa bližšie zameriame na IoC skenery. Skenery pomocou rôznych pravidiel a indikátorov kompromitácie skenujú systém, identifikujú hrozby a pomáhajú pri procese odpovede na bezpečnostný incident. V poslednej podkapitole sa pozrieme na threat intelligence platformy. Threat intelligence platformy sú komplexné aplikácie, ktoré agregujú a organizujú dáta o hrozbách z rôznych zdrojov a v rôznych formátoch.

3.1 Extraktory IoC

Extraktory IoC sú nástroje, ktoré dokážu z rôzneho vstupného súboru vyextrahovať indikátory kompromitácie. Fungujú na princípe parsovania textu a vyhľadávania potencionálnych indikátorov kompromitácie pomocou regulárnych výrazov. Na porovnanie sme vybrali päť nástrojov, a to InQuest - python-iocextract [40], ninoseki - ioc-extractor [41], sorberts – cacador [42], PaloAltoNetworks - ioc-parser [43] a IOCParser [44].

	InQuest python-iocextractor	Ninoseki - ioc-extractor	Sorberts cacador	PaloAltoNetwork ioc-parser	IOCPARSER
Otvorený kód	✓	✓	✓	✓	✗
Programovací jazyk	Python	JavaScript	Go	Python	?
Formát vstupu	Blok textu	Blok textu	Blok textu	PDF, TXT, HTML	Blogy, PDF, CSV, TXT,

					ZIP, Blok textu, JSON...
Podporované IoC	YARA	SSDEEP, MAC adresy, BTC/ETH/XMR adresy	SSDEEP, prípony súborov	Registre, súbory podľa prípon	MAC adresy, BTC adresy, MITRE ATT&CK id, YARA
Odobfuskovanie	IPv4, email, URL	IPv4, email, URL	IPv4, URL	-	Áno
Vlastné regulárne výrazy	✓	×	×	×	×
Formát výstupu	?	STIX	JSON	CSV, JSON, YARA, autofocus	JSON

Tab. 1 Porovnanie extraktorov IoC

Tabuľka č. 1 obsahuje porovnanie spomínaných nástrojov na extrakciu indikátorov kompromitácie. Tieto nástroje sme porovnávali z rôznych hľadísk. Jedným z prvých je, či ide o nástroj s otvoreným kódom. Prvé štyri spomínané nástroje sú s otvoreným kódom umiestnené na platforme GitHub. Nástroj IOCParser je online služba, na ktorú sa vieme dopytovať cez API, a teda nemá dostupný kód. Zvolené nástroje sú napísané prevažne v jazyku Python, Java Script a GO. Jedným z atribútov, ktorý sme porovnávali je formát vstupu. Prvé tri nástroje podporujú vstup iba ako blok textu. Zaujímavejšie z tohto hľadiska sú nástroj PaloAltoNetwork ioc-parser a služba IOCParser, ktoré ako vstup dokážu spracovať aj rôzne typy súborov. Jeden z najdôležitejších atribútov daných nástrojov je zoznam indikátorov kompromitácie, ktoré dokážu zo vstupu vyextrahovať. Základom v tejto oblasti sú IPv4 a IPv6 adresy, URL adresy, emailové adresy a MD5/SHA1/SHA256 heše súborov. Zaujímavosťou sú napríklad bitcoin (BTC) adresy a MAC adresy, ktoré dokážu vyextrahovať nástroje sorbets – cacador a IOCParser alebo SSDEEP heše, v ktorých vynikajú nástroje sorbets – cacador a Ninoseki – ioc-extractor. Ďalšou dôležitou časťou je odobfuskácia (refang)

indikátorov. V tejto časti vynikajú prvé tri nástroje, ktoré dokážu takýmto spôsobom upraviť IPv4 adresy, emailové adresy a URL. Služba IOCParser má túto schopnosť tiež, avšak nie je definované aké indikátory kompromitácie tak dokáže spracovať. Posledným sledovaným atribútom je formát výstupu. Pri nástroji InQuest – python-iocextractor nie je špecifikovaný formát výstupu. Pri ostatných to je všeobecný JSON alebo STIX formát, no zaujímavosťou je PaloAltoNetwork ioc-parser, ktorý dokáže exportovať aj YARA pravidlá.

Keďže náš nástroj má byť schopný extrahovať indikátory kompromitácie z obrazu disku, je dôležité pochopiť fungovanie týchto nástrojov. Dôležitými sledovateľmi sú druhy indikátorov kompromitácie, ktoré dokáže nástroj spracovať ale takisto aj vstup a výstup nástroja. Na rozdiel od týchto nástrojov, ktoré pracujú poväčšine s textom, náš nástroj pracuje s obrazom disku.

3.2 Skenery IoC

Skenery IoC sú nástroje, ktoré môžu skenovať systém, triediť získané informácie, identifikovať v ňom hrozby a tým uľahčiť forenznú analýzu v prípade incidentu. V skeneroch sú zakomponované YARA pravidlá, SIGMA pravidlá, pravidlá na početné anomálie a indikátory kompromitácie. Na porovnanie sme vybrali päť skenerov, a to LOKI [45], THOR Lite [46], THOR [47], Redline [48] a Kaspersky Log Scanner [49].

	LOKI	THOR Lite	THOR	Redline	KASPERSKY Log Scanner
Otvorený kód	✓	×	×	×	×
Cena	zadarmo	zadarmo/ registrácia	platené	zadarmo/ registrácia	súčasť Cybertrace
Platforma	Windows (skompilované) Linux/macOS (zdrojový kód)	Windows Linux macOS	Windows, Linux macOS AIX	Windows, macOS/Linux (obmedzenia)	Windows Linux

Podpora	Github README	Manual, vnútorná CI	Manual, SUPPORT portál, vnútorná CI	Manual, e-mail	Manual, SUPPORT portál
Hlavné použitie	Triáž	Triáž	Skenovanie, IR, živá FA	Triáž, analýza	Triáž
Sken živého systému	✗	✓/✗	✓	✗	✗
Typ reportu	Text log, CSV	Text log, SYSLOG, JSON	Text log, SYSLOG, JSON, HTML	CSV	Textový súbor

Tab. 2 Porovnanie skenerov IoC

V Tabuľke č. 2 môžeme vidieť porovnanie vyššie spomínaných skenerov indikátorov kompromitácie. Prvým atribútom, ktorý sme sledovali bolo, či ide o nástroj s otvoreným kódom. Z porovnávaných nástrojov bol jediný nástroj LOKI v otvorenom kóde. LOKI bol vytvorený ako prepis skenerov v nástroji THOR, ktorý je platený. Všetky nástroje podporujú operačné systémy Windows a Linux a okrem Log Scanneru aj MacOS. Ako môžeme vidieť, nástroj THOR podporuje aj operačný systém AIX, čo je proprietárny UNIXový operačný systém spoločnosti IBM. Spomedzi všetkých nástrojov majú THOR a Log Scanner (Kaspersky Cybertrace) okrem manuálu aj zákaznícky portál. Základnou vlastnosťou týchto nástrojov je triáž indikátorov kompromitácie, no nástroje THOR a THOR Lite ponúkajú aj skenovanie živého systému, pričom THOR Lite ponúka iba obmedzené funkcie oproti nástroju THOR. Ďalším dôležitým atribútom je formát, v akom sa dá vytvoriť report, resp. extrahovať údaje. Nástroj Redline ponúka report iba v CSV formáte a Kaspersky Log Scanner iba v textovom formáte. Najbohatší z pohľadu formátov je THOR, ktorý ponúka až štyri formáty. Posledným atribútom, aj keď nie je zobrazený v tabuľke, je možnosť pridať vlastné rozšírenia. Nástroj THOR môžeme rozšíriť o zoznam vlastných indikátorov kompromitácie vo formáte STIX. Keďže LOKI je nástroj s otvoreným kódom, je taktiež možné pridať vlastné indikátory kompromitácie ale aj YARA pravidlá, čo je veľká výhoda z pohľadu vytvárania špecifických hľadaní v systéme.

Ak porovnáme skenery indikátorov kompromitácie s návrhom nášho nástroja, líšia sa v hlavnom použití, kde funkcionalitou týchto nástrojov môže byť aj analýza alebo sken živého systému. Tieto nástroje majú spoločnú podobnú funkcionalitu s našim nástrojom a tou je triáž indikátorov kompromitácie. Pri návrhu nášho nástroja sme nerátali s možnosťou skenovania živého systému, zameranie je hlavne na sken a analýzu obrazu disku zaisteného pri bezpečnostnom incidente.

3.3 Threat intelligence platformy

Threat intelligence platformy sú technologické riešenia, ktoré zbierajú, agregujú a organizujú dáta o hrozbách z viacerých zdrojov. Tieto platformy dávajú analytikom doplňujúce informácie o hrozbách. Pomocou threat intelligence platforiem môžu analytici vytvárať rôzne analýzy a tým zvýšiť efektivitu vyšetrovania bezpečnostného incidentu. Taktiež si môžu bezpečnostné tými medzi sebou zdieľať informácie cez tieto platformy.

My sme na porovnanie vybrali päť threat intelligence platform, ktoré patria k celosvetovo najpoužívanejším platformám. Ide o platformu Yeti [50], Open-CTI [51], CyberTrace od spoločnosti Kaspersky [52], MISP [53] a TheHive [54].

Yeti [50] je platforma s otvoreným kódom napísaná prevažne v Pythone a JavaScripte, ktorá slúži na organizáciu pozorovateľov, indikátorov kompromitácie, TTP a vedomostí o bezpečnostných hrozbách na jednom mieste. Taktiež obohacuje tieto údaje o ďalšie informácie a dokáže ich vyexportovať alebo zobraziť v grafe, kde sú znázornené vzťahy medzi rôznymi hrozbami. Yeti používa ďalšie podsystémy a nástroje pre rôzne analýzy a obohacovanie údajov. Jedným z týchto podsystémov je napríklad FIR a FAME. FIR je platforma na riadenie bezpečnostných incidentov a FAME je platforma na analýzu malvéru. Yeti pomocou dátových kanálov zbiera údaje o hrozbách, ktoré sa analyzujú pomocou spomínaných podsystémov ako FIR a FAME. Ako zdroj dátových kanálov môže byť použitý napríklad MISP, ktorý si ešte spomenieme alebo rôzne malvér trakery a XML alebo JSON súbory. Yeti poskytuje používateľské API pre automatizáciu zbierania dát alebo automatizáciu analýz. Výsledky analýz môže zdieľať napríklad do SIEM alebo monitorovacích systémov alebo do rôznych treťostranových aplikácií. Taktiež umožňuje pridávať vlastné rozšírenia, napríklad na zdroje dátových kanálov alebo rôzne analýzy.

OpenCTI [51] je platforma s otvoreným kódom napísaná prevažne v JavaScripte, ktorá dovoľuje manažovať vedomosti o hrozbách. Bola vytvorená s cieľom štruktúrovať, uložiť, organizovať a vizualizovať technické a netechnické informácie o bezpečnostných hrozbách. Štruktúra dát je založená na STIX schéme. Platforma je dizajnovaná ako moderná webová aplikácia a môže byť integrovaná s inými platformami, ako napríklad MISP, The Hive alebo MITRE ATT&CK. Cieľom je vytvoriť komplexný nástroj, ktorý kapitalizuje technické a netechnické informácie, ktoré linkuje ku ich základnému zdroju. Z toho vie ukázať zaujímavé štatistiky, ako napríklad kedy bol indikátor prvýkrát a kedy poslednýkrát videný a pod. Používateľ môže extrahovať dáta v rôznych formátoch, napríklad obrázky grafov a štatistík, CSV alebo STIX formát. Tak ako YETI, aj OpenCTI poskytuje rozšírenia platformy, a to napríklad vo forme implementácie vlastných datasetov.

CyberTrace [52] je threat intelligence platforma, ktorá umožňuje integráciu dátových kanálov so SIEM systémom a pomáha analytikom efektívnejšie využívať informácie o hrozbách. Zdroje dátových kanálov môžu byť v JSON, STIX, CSV alebo XML formáte napríklad dátové kanály od spoločnosti Kaspersky alebo aj vlastné dátové kanály. CyberTrace využíva vlastný proces analýzy a parsovania údajov a tým dokáže generovať vlastné výstrahy pri detekcii hrozieb. Funguje na nasledujúcom princípe: SIEM zbiera logy z rôznych zariadení, to sa posiela do systému CyberTrace, ktorý vykoná vlastné analýzy a porovnania. Následne tieto údaje odosiela na CyberTrace web a späť do SIEM systému. Nakoniec vyšetrovateľ uvidí udalosti s kontextom a výstrahy založené na danom kontexte. CyberTrace poskytuje databázu indikátorov kompromitácie s full-textovým vyhľadávaním, stránku s detailnými informáciami o každom indikátore, výskumný graf pre vizuálny prieskum dát, možnosť exportovať IoC. Taktiež ukazuje rôzne štatistiky ako napríklad efektivitu dátových kanálov a taktiež poskytuje REST API.

MISP [53] (Malware Information Sharing Platform) je platforma s otvoreným kódom napísaná prevažne v jazyku PHP a JavaScripte, vytvorená na zbieranie, ukladanie, distribuovanie a zdieľanie bezpečnostných indikátorov a hrozbách o analýzach incidentov a malvérov. Cieľom MISPu je podporovať zdieľanie štruktúrovaných informácií o bezpečnostných hrozbách ale taktiež aj používanie týchto informácii ďalšími systémami ako napríklad NIDS, LIDS, SIEM alebo nástrojmi na analýzu logov. Medzi hlavné funkcionality systému MISP patrí efektívna databáza na ukladanie technických a netechnických informácií o malvéroch a incidentoch, automatizované hľadanie

vzťahov medzi atribútmi a indikátormi malvéru, útoku alebo analýzy. MISP poskytuje aj používateľské rozhranie ale taktiež funguje aj cez API. Dáta ukladá v štruktúrovanom formáte pre automatizované používanie. Tak ako predchádzajúce platformy, aj MISP sa dá rozšíriť. Podporuje pridávanie importových a exportových modulov, čo znamená, že dokáže importovať aj exportovať dáta v rôznych formátoch ako napríklad spomínané OpenIOC, STIX, CSV, XML, JSON. Taktiež podporuje pridávanie už existujúcich alebo vlastných rozširujúcich modulov písaných v Pythone.

TheHive [54] je platforma s otvoreným kódom napísaná prevažne v jazykoch Scala, JavaScript a HTML vytvorená pre jednoduchú odpoveď na incidenty. Cieľom TheHive je zjednodušiť prácu SOC, CSIRT a CERT tímom a preto sa dá jednoducho prepojiť s inými platformami. Jednou z platforiem, s ktorou sa dokáže TheHive prepojiť je už spomínaný MISP a to s jednou alebo viacerými inštanciami MISP-u pre začiatok vyšetrovania z udalostí z MISP-u. Taktiež je možné exportovať výsledky vyšetrovania ako MIPS udalosti a tak pomáhať iným spoločnostiam reagovať na podobné útoky. TheHive sa používa spolu s Cortexom a teda dokáže analyzovať desiatky pozorovateľných objektov. Podporuje rôzne metódy na ukladanie dát, súborov a indexov podľa potreby. Dokáže automaticky identifikovať sledovateľné objekty, videné v predchádzajúcich prípadoch, ktoré následne dokážu analytici označiť ako indikátory kompromitácie a izolovať ich. Taktiež dokáže odosielať upozornenia a prehľadávať SIEM systémy.

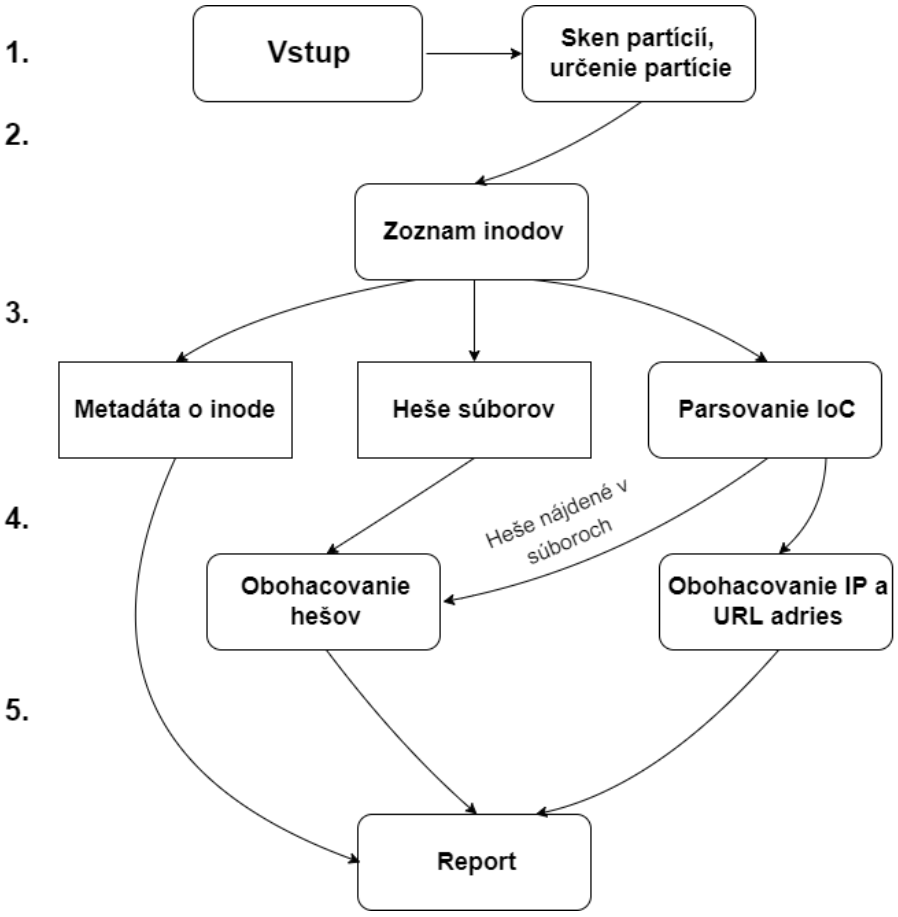
Threat intelligence platformy sú v porovnaní s naším nástrojom komplexnejšími nástrojmi. Ponúkajú zber, agregáciu a organizáciu dát o hrozbách, ktoré majú z rôznych zdrojov. Taktiež ponúkajú rôzne analyzátory na obohacovanie informácií o hrozbách. Náš nástroj sa čiastočne inšpiruje konceptom zberu indikátorov kompromitácie a následným obohatením o doplňujúce informácie.

4 Nástroj na spracovanie indikátorov kompromitácie

Hlavným cieľom tejto bakalárskej práce je navrhnúť a implementovať nástroj na spracovanie údajov o bezpečnostných hrozbách, najmä indikátorov kompromitácie pri riešení bezpečnostných incidentov. Nástroj by mal byť schopný analyzovať obraz disku a následne vyextrahovať rôzne indikátory kompromitácie. Následne by sa mali získané indikátory kompromitácie obohatiť o ďalšie informácie pomocou vybraných služieb. Výstupom by mal byť jednoduchý report so zoznamom indikátorov kompromitácie a prípadnými ďalšími dostupnými informáciami o nich.

4.1 Návrh riešenia

Navrhnutý nástroj je implementovaný v programovacom jazyku Python verzie 3 a na pozadí využíva viacero nástrojov a služieb ako napríklad Sleuthkit [55], Hybrid-Analysis [56] alebo Shodan [57]. Skladá sa z piatich hlavných častí, ktoré si bližšie popíšeme v nasledujúcom texte. Na Obrázku č. 19 môžeme vidieť diagram popisovaného nástroja.



Obr. 19 Diagram návrhu nástroja

Ako vstup nástroja sa stanovil obraz disku v surovom (raw) formáte. Nástroj na vstupe dostane obraz disku. Dôležité je, aby obraz disku obsahoval partície so súborovým systémom, ktorý je koncepčne postavený na tzv. inodoch. Inode je dátová štruktúra v súborovom systéme v Linuxe a iných Unixových systémoch, ktorá ukladá všetky informácie o súbore okrem jeho názvu a skutočných údajov [58]. Pôjde najmä o súborové systémy pre operačný systém Windows (NTFS, FAT16, FAT32), ako aj pre operačný systém Linux (EXT3, EXT4). Predpokladáme, že zaistený obraz disku nebude obsahovať RAID (Redundand Array of Independent Disks), LVM (Logical Volume Manager) ani rôzne typy šifrovania (napr. LUKS – Linux Unified Key Setup). Následne sa analyzuje obraz disku a získajú sa údaje o rozložení partícií. Po získaní offsetu výberom partície sa získajú metadáta o súboroch z obrazu disku. Z týchto metadát získame zoznam inodov, ktorý používame na prechádzanie súbormi a počítanie SHA1 hešov súborov. V tejto časti parsujeme evtx súbory pomocou python-evtx parsera a následne pomocou regulárnych výrazov hľadáme indikátory kompromitácie. V ďalšom kroku sa indikátory

kompromitácie obohacujú o doplňujúce informácie pomocou služieb Shodan a Hybrid-Analysis. Nakoniec sa vytvorí sumárny report so všetkými nájdenými indikátormi kompromitácie.

4.2 Použité technológie

Nástroj používa rôzne technológie, ktoré pomáhajú pri práci s dátami alebo analýzami. Hlavnou technológiou je zbierka nástrojov Sleuthkit, ktorá nám pomáha pri práci a analýzou obrazu disku. Ďalším použitým nástrojom je python-evtx, ktorý spracováva evtx súbory do čitateľného stavu. Následne sú použité služby ako Shodan a Hybrid-Analysis pre získanie dodatočných informácií o nájdených indikátoroch kompromitácie.

4.2.1 Sleutkit

Sleuthkit je zbierka nástrojov príkazového riadku a knižnica jazyku C. Tieto nástroje umožňujú analyzovať obrazy disku, získať informácie o zväzkoch a súborovom systéme. Nástroje súborového systému umožňujú skúmať súborové systémy nerušivým spôsobom. Keďže sa nástroje nespoliehajú na operačný systém, dokážu zobrazovať aj zmazané a skryté súbory. Knižnica je začlenená vo viacerých forenzných nástrojoch ako napríklad Autopsy. Táto zbierka nástrojov je napísaná v jazykoch C a Perl. Je kompatibilná s operačnými systémami Linux, MacOS a Windows. Nástroje zväzkového systému umožňujú preskúmať rozloženie diskov a iných médií. Sleuthkit podporuje DOS partície, BSD partície, MAC partície, "Sun slices" (obsah zväzkov) a GPT disky. Pomocou týchto nástrojov môžeme identifikovať umiestnenie partícií a extrahovať ich, pre analýzu pomocou nástrojov súborového systému.

4.2.2 EVTX Parser

Python-evtx [59] je parser pre súbory denníka udalostí pre Windows (Windows Event Log). Tieto súbory sa vyznačujú tým, že sú vo formáte proprietárneho binárneho XML súboru. Označujú sa koncovkou súboru .evtx. Tento modul poskytuje prístup ku súborovým hlavičkám a hlavičkám chunk-ov, šablónam záznamov a záznamov udalostí.

Výsledkom spracovania evtx súboru je kompletný XML dokument s variabilnou schémou.

4.2.3 Shodan

Shodan je vyhľadávač zariadení pripojených na internet. Zhromažďuje informácie o zariadeniach priamo pripojených k internetu, požiadaním zariadenia o verejne dostupné informácie. Môžu to byť informácie o softvéri na danom serveri, "správa na privítanie", aké služby server poskytuje alebo dostupné porty. Shodan bol navrhnutý pre inžinierov a vývojárov, čomu odpovedá aj zložitejšia syntax.

4.2.4 Hybrid-Analysis

Hybrid-Analysis je nezávislá služba, ktorú poháňa Falcon Sandbox. Poskytuje prístup ku analýze súborov kombináciou údajov za behu programu a analýzou výpisu pamäte. Všetky dáta, ktoré sú extrahované z nástroja Hybrid-Analysis sa automaticky analyzujú a spracovávajú do správ o analýze malvéru. Používatelia môžu nahrávať nové vzorky, ktoré budú spracované a analyzované, prehľadávať databázu existujúcich vzoriek alebo si tieto vzorky sťahovať.

4.3 Implementácia riešenia

Pre implementáciu skriptu nástroja sme sa rozhodli použiť programovací jazyk Python verzie 3, kvôli veľkému množstvu dostupných knižníc. Taktiež používame príkazy príkazového riadku Linuxu z dôvodu používania nástrojov Sleuthkitu a jednoduchému narábaniu s dátami.

4.3.1 Výber partície

V **prvom kroku** sa použije nástroj "mmls" zo skupiny nástrojov Sleuthkit [60]. Tento nástroj zobrazuje rozloženie partícií (tabuľku partícií) v obraze disku. Vďaka tomu si môže užívateľ vybrať konkrétnu partíciu, ktorú chce skenovať. Následne sa na základe tejto voľby získa z tabuľky partícií offset, na ktorom daná partícia začína. Offset je

dôležitý pre nástroje v ďalších krokoch. Ukážka nástroja v tomto kroku je znázornená na Obrázku č. 20.

```
#ponuka particii pre uzivatela
os.system("mmls " + image_name + " | awk '/^[0-9]/{print $0}'")
```

Obr. 20 Ukážka použitia nástroja "mmls"

4.3.2 Vytvorenie „body file“

V **druhom kroku** sa použije nástroj “tsk_gettimes“ zo skupiny nástrojov Sleuthkit [61]. Tento nástroj zbiera MAC (modified, access, change) časové pečiatky z obrazu disku do takzvaného “body file“. MAC časové pečiatky sú metadáta súborového systému, ktoré zaznamenávajú isté udalosti týkajúce sa súborov. “Body file“ je prechodný súbor, ktorý sa vytvorí pri vytváraní časovej osi aktivít súboru. Keďže “tsk_gettimes“ získa metadáta o všetkých partíciách, potrebujeme pomocou príkazu “grep“ získať iba metadáta o vybranej partícii. To dokážeme vyfiltrovať na základe názvu súboru, keďže názov obsahuje celú cestu súboru a teda aj partíciu, na ktorej sa nachádza. Výstup nástroja “tsk_gettimes“ je teda “body file“, ktorý si uložíme do súboru. Tento výstup má nasledujúci formát:

```
MD5|name|inode|mode_as_string|UID|GID|size|atime|mtime|ctime|crtime
```

Tieto údaje znamenajú: MD5 heš súboru, názov, inode, mód v akom daný súbor je (r - read, w - write), id používateľa (user id), id skupiny (group id), veľkosť súboru, čas posledného prístupu (acces time - atime), čas poslednej úpravy obsahu (modified time - mtime), čas poslednej zmeny metadát (change time - ctime) a čas vytvorenia súboru (creation time - crtime). Následne tento súbor otvoríme a pomocou for cyklu ním prejdeme. Keďže ďalšie nástroje, s ktorými budeme pracovať, prehľadávajú obraz disku na základe inodov, musíme si zoznam inodov extrahovať pomocou separátora “|“ a uložiť. Ukážka nástroja v tomto kroku a extrahovania inodov je znázornená na Obrázku č. 21.

```
#vytvorenie zoznamu suborov s metadatami
os.system("tsk_gettimes " + image_name + " | grep vol" + str(partition_number) + " > subory.txt")

#vytvorenie zoznamu inodov
inodes = []
with  open('subory.txt') as f:
        for line in f.readlines():
                inode = line.split("|")[2]
                inodes.append(inode)
```

Obr. 21 Ukážka použitia nástroja "tsk_gettimes" a práce s jeho výstupom

4.3.3 Extrakcia IoC

Tretí krok sa zaoberá vytváraním hešov súborov a parsovaním metadát a indikátorov kompromitácie zo súborov. Ako sme už spomínali v predchádzajúcom kroku, z výstupu nástroja “tsk_gettimes“ vieme jednoducho parsovať pomocou separátora “|“. O každom súbore si uložíme v štruktúre jazyka python “dictionary“ vyššie spomínané zaujímavé údaje. Následne prechádzame všetkými súbormi, a to cez zoznam všetkých inodov. Na zobrazenie obsahu súboru používame nástroj “icat“ zo Sleuthkitu s prepínačom “-o“, ktorý slúži na určenie offsetu partície a atribútmi názov obrazu disku a inode súboru, ktorého obsah chceme zobraziť. Obsah súboru pošleme pomocou linuxovej rúry do nástroja sha1sum, ktorý vytvorí SHA1 heš súboru. Následne pomocou príkazu “grep“ s prepínačom “-v“, odstránime heše prázdnych súborov. Na Obrázku č. 22 môžeme vidieť prácu s nástrojom “icat“, následné odstránenie hešov prázdnych súborov a vloženie dvojice inode – heš do dictionary.

```
#metoda na vypocitanie SHA1 hesu suboru
def count_SHA1(image_name, offset, dictionary, inode):
        hash = os.popen("icat -o " + offset + " " + image_name + " " + str(inode) +\
" | sha1sum | awk '{print $1}' | grep -v " + EMPTY_STRING_SHA1).read().strip('\n')
        set_key(dictionary, inode, hash)
```

Obr. 22 Ukážka použitia nástroja "icat", vypočítania hešu a vloženia do dictionary

Keďže súbory windowsových logov (.evtx) nie sú čitateľné nástrojom “icat“, na ich parsovanie musíme použiť externý nástroj python-evtx. Tento nástroj zo vstupného evtx súboru vytvorí xml súbor s logmi, ktorý už je čitateľný a teda sa dá bežne parsovať. Dôležitou časťou je zistenie, či je súbor formátu evtx. To overíme pomocou takzvaných magic bajtov. Magic bajty (File signatures) je niekoľko prvých bajtov súboru, pomocou

ktorých môžeme rozoznať, o aký druh súboru ide [62]. Pri tomto overovaní sa používa nástroj “xxd“, ktorý vytvorí hexadecimálne zobrazenie súboru. Následne pomocou príkazu “grep“ vyberieme iba súbory ktoré začínajú s nasledujúcimi bajtami: 45 6C 66 46 69 6C 65 00. Toto overenie a použitie nástroja “xxd“ je znázornené na Obrázku č. 23.

```
#metoda na overenie ci je subor evtx
def is_evtx(image_name, offset, inode):
        x = os.popen("icat -o " + offset + " " + image_name + " " + inode +\
" | xxd -l 8 | grep '456c 6646 696c 6500'").read()
        if x != '':
                return 1
        else:
                return 0
```

Obr. 23 Ukážka overenia "magic bajtov" pomocou nástroja "xxd"

Ak sa nejedná o evtx súbor, nad obsahom súboru hľadáme pomocou regulárnych výrazov IPv4, URL a emailové adresy a md5 heše. Na prácu s regulárnymi výrazmi používame pythonovskú knižnicu “re“, ktorá obsahuje metódu “findall“. Táto metóda potrebuje regulárny výraz, na základe ktorého bude vyhľadávať a text, v ktorom to má vyhľadávať. Na Obrázku č. 24 je príklad použitia metódy “findall“ z knižnice “re“ pre nájdenie indikátora kompromitácie pomocou regulárneho výrazu.

Všetky zhody ukladáme do dictionary formou IoC, ktoré sme našli a inody, v ktorých sa tieto IoC objavili. Takýmto spôsobom sa budeme vedieť spätne dohľadať, v ktorých súboroch boli problémové indikátory kompromitácie.

```
#vseobecna metoda na parsovanie IoC
def parse_ioc(file, inode, pattern, dictionary):
        iocs = re.findall(pattern, file)
        for ioc in iocs:
                set_key(dictionary, ioc, inode)
```

Obr. 24 Ukážka použitia metódy "findall" na nájdenie IoC

4.3.4 Obohacovanie IoC

V **štvrtom kroku** sa zameriavame na obohacovanie indikátorov kompromitácie. Obohacovanie v tomto prípade znamená získanie dodatočných informácií o konkrétnych indikátoroch kompromitácie. Ako sme už spomínali v predchádzajúcich krokoch, všetky indikátory kompromitácie máme uložené v štruktúre jazyka python dictionary. Pomocou metódy “keys()“ získame list indikátorov kompromitácie.

V prvej časti overíme IP adresy pomocou knižnice Shodan pre python na komunikáciu s ich API. Ako prvé musíme pomocou volania “SHODAN_API=shodan.Shodan(API_KEY)“ získať prístup do API s vlastným API kľúčom. API Kľúč sa vygeneruje každému zaregistrovanému užívateľovi. Následne vo for cykle prechádzame cez list IP adries. Ako môžeme vidieť na Obrázku č. 25 pomocou metódy “host“ zašleme požiadavku na overenie danej IP adresy. Táto metóda vráti súbor, z ktorého vyberieme požadované prvky.

```
for ip in ips:
        try:
                if type(dict_ipv4s[ip]) == list:
                        print("Inode: " + set(dict_ipv4s.get(ip, 0)) + ":", file = ipv4_file)
                else:
                        print("Inode: " + dict_ipv4s.get(ip, 0) + ":", file = ipv4_file)
                host = SHODAN_API.host(ip)
                print("""       IP: {}
        Organization: {}
        Country Code: {}
        Country Name: {}
        City: {}
        Latitude: {}
        Longtitude: {}
        Host Names: {}
        Ports: {}
        Operating System: {}
                """.format(host['ip_str'], host.get('org', 'n/a'), host.get('country_code', 'n/a'),\
host.get('country_name', 'n/a'), host.get('city', 'n/a'), host.get('latitude', 'n/a'),\
host.get('longitude', 'n/a'), host.get('hostnames', 'n/a'), host.get('ports', 'n/a'),\
host.get('os', 'n/a')), file = ipv4_file)
        except shodan.APIError as e:
                print(' {}: {}\n'.format(ip, e), file = ipv4_file)
```

Obr. 25 Použitie metódy "host" a výber požadovaných prvkov

V druhej časti overíme heše súborov pomocou VxAPI, ktoré dokáže komunikovať so službou Hybrid-Analysis. Na konfiguráciu VxAPI je potrebné v súbore config.py vložiť vlastný API kľúč, ktorý je dostupný po zaregistrovaní sa do služby. Taktiež treba skontrolovať, či je názov servera nastavený na domovský stránku služby Hybrid-Analysis. Ako môžeme vidieť na Obrázku č. 26, službu použijeme pomocou

metódy “os.system()“ v ktorej spustíme skript “vxapi.py“. Z viacerých možností sme vybrali parameter “search_hash“, ktorý v databáze Hybrid-Analysis vyhľadá všetky dostupné informácie o danom heši. Odpoveď od API uložíme do súboru pre následné vybratie dôležitých informácií a vytvorenie reportu.

```
#overovanie hesov najdenych v suboroch a vypis zistenych informacii
for i in dict_md5s:
        os.system("python3 VxAPI-master/vxapi.py search_hash " + i + " > ha.txt")
```

Obr. 26 Použitie VxAPI s parametrom search_hash

4.3.5 Vytvorenie zoznamu nájdených IoC

Piaty krok sa zaoberá vytvorením zoznamov nájdených indikátorov kompromitácie. Rozhodli sme sa vytvoriť jednotlivé zoznamy pre metadáta o súboroch, IP adresy, URL adresy, emailové adresy a heše. Tieto zoznamy sú ukladané do textových súborov. Na Obrázku č. 27 je uvedený príklad vytvárania zoznamu emailových adries. Analogicky sa vytvárajú zoznamy s ďalšími indikátormi kompromitácie.

```
emails = dict_emails.keys()
email_file = open('report_email.txt', 'w')
for email in emails:
        if type(dict_emails[email]) == list:
                print("""Inode: {}:
        Email: {}""".format(set(dict_emails.get(email, 0)), email), file = email_file)
        else:
                print("""Inode: {}:
        Email: {}""".format(dict_emails.get(email, 0), email), file = email_file)
email_file.close()
```

Obr. 27 Vytváranie zoznamu emailových adries

Záver

Keďže počet zariadení pripojených do siete Internet neustále rastie a s tým rastie aj počet bezpečnostných hrozieb, je potrebné sa informačnej a kybernetickej bezpečnosti. Jedným zo spôsobov ako znížiť bezpečnostné riziko je poznať nepriateľa, aby sme sa vedeli lepšie chrániť. K tomu napomáha rýchle spracovanie a analýza predchádzajúcich kybernetických bezpečnostných incidentov. Z toho vyplýva nutnosť vyvíjať efektívnejšie postupy a nástroje na spracovanie údajov o bezpečnostných hrozbách, a teda aj indikátorov kompromitácie.

Prvým cieľom tejto bakalárskej práce je analýza možností spracovania údajov o bezpečnostných hrozbách, najmä indikátorov kompromitácie prostredníctvom threat intelligence. Tomuto cieľu sme sa venovali v prvých dvoch kapitolách. V prvej kapitole sme priblížili základné definície a princípy threat intelligence, ako aj modely používané v tejto oblasti. Dôležitou časťou bolo pochopenie základného princípu fungovania a využitia threat intelligence.

V druhej kapitole sme sa venovali indikátorom kompromitácie. Keďže spomedzi údajov, ktoré skúma threat intelligence, sme sa zamerali na indikátory kompromitácie, bolo dôležité ich pochopiť do hĺbky. Ako prvé rozoberáme model pyramídy bolesti a následne sa venujeme ich typom a komponentom. V poslednej časti porovnávame rôzne spôsoby uloženia a zdieľania indikátorov kompromitácie.

Druhý cieľ je spracovaný v rámci tretej kapitoly, kde analyzujeme a porovnávame nástroje a prístupy k analýze bezpečnostných hrozieb. Táto kapitola je rozdelená do troch častí, z ktorých prvá je zameraná na nástroje ktoré extrahujú indikátory kompromitácie. Tieto nástroje sú dôležitou súčasťou porovnania, keďže aj v návrhu nášho nástroja sa používa extrakcia indikátorov kompromitácie. Druhá časť skúma nástroje na skenovanie systému, ktoré dokážu triediť získané informácie a odhaľovať bezpečnostné hrozby. V poslednej časti sa zaoberáme najkomplexnejšími nástrojmi, a teda threat intelligence platformami. Tieto platformy dokážu zbierať, agregovať a organizovať dáta o bezpečnostných hrozbách z viacerých zdrojov a zároveň pridávajú týmto dátam doplňujúce informácie.

Posledným cieľom tejto práce je navrhnúť, implementovať a vyhodnotiť nástroj na spracovanie indikátorov kompromitácie pri riešení kybernetických bezpečnostných incidentov. Fungovanie nástroja je rozdelené do piatich častí: vstup, skenovanie obrazu

disku, extrakcia indikátorov kompromitácie, obohacovanie indikátorov kompromitácie a vytvorenie výstupu. Pre prácu s obrazom disku sme sa rozhodli využiť sadu nástrojov Sleuthkit. Problematickou časťou boli evtx súbory, ktoré tieto nástroje nevedia prečítať. Tento problém sme vyriešili evtx parserom [59]. Následne na analýzu indikátorov kompromitácie používame online služby Shodan [57] a Hybrid Analysis [56].

V budúcnosti je možné tento nástroj rozšíriť o viac druhov extrahovaných indikátorov kompromitácie alebo obohacovanie indikátorov kompromitácie o dodatočné informácie pomocou viacerých online služieb.

Zoznam použitej literatúry

1. Threat Intelligence Definition. Why Threat Intelligence Is Important for Your Business and How to Evaluate a Threat Intelligence Program. [online] 12.4.2022 Dostupné z: https://www.recordedfuture.com/threat-intelligence/
2. What the 6 Phases of the Threat Intelligence Lifecycle Mean for Your Team. [online] 1.5.2022 Dostupné z: https://www.recordedfuture.com/threat-intelligence-lifecycle-phases/
3. Threat Intelligence Lifecycle. [online] 1.5.2022 Dostupné z: https://info-savvy.com/threat-intelligence-lifecycle/
4. The Threat Intelligence Lifecycle: A Complete Guide. [online] 1.5.2022 Dostupné z: https://securityscorecard.com/blog/threat-intelligence-lifecycle-guide
5. Recorded Future. [online] 10.3.2022 Dostupné z: https://www.recordedfuture.com/
6. Types of Threat Intelligence. [online] 11.4.2022 Dostupné z: https://info-savvy.com/types-of-threat-intelligence/
7. What Is Threat Intelligence? Definition and Types. [online] 3.5.2022 Dostupné z: https://www.dnsstuff.com/what-is-threat-intelligence
8. MITRE ATT&CK [online] 18.4.2022 Dostupné z: https://attack.mitre.org/
9. I. You and K. Yim, "Malware Obfuscation Techniques: A Brief Survey," 2010 International Conference on Broadband, Wireless Computing, Communication and Applications, 2010, pp. 297-300, doi: 10.1109/BWCCA.2010.85. [online] 16.5.2022 Dostupné z: https://ieeexplore.ieee.org/abstract/document/5633410
10. https://apps.dtic.mil/sti/pdfs/ADA586960.pdf
11. Sergio Caltagirone, Andrew Pendergast, Christopher Betz. The Diamond Model of Intrusion Analysis. [online] 3.5.2022 Dostupné z: https://www.activeresponse.org/wp-content/uploads/2013/07/diamond.pdf
12. The Cyber Kill Chain [online] 26.4.2022 Dostupné z: https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html
13. Maciej Makowski. 2020. The Cyber Kill Chain explained – along with some 2020 examples. [online] 3.5.2022 Dostupné z: https://eforensicsmag.com/the-cyber-kill-

chain-explained-along-with-some-2020-examples-by-maciej-makowski/?fbclid=IwAR2IGZqGPfbIcqE6cbg8-tORAKfinYO0bkGXLB7c0b1repWqJ1iULiFXnfM

14. CyCraft Classroom: MITRE ATT&CK vs. Cyber Kill Chain vs. Diamond Model. [online] 26.4.2022 Dostupné z: https://medium.com/cycraft/cycraft-classroom-mitre-att-ck-vs-cyber-kill-chain-vs-diamond-model-1cc8fa49a20f
15. Wiem Tounsi, Helmi Rais. A Survey on technical threat intelligence in the age of sophisticated cyber attacks. [online] 15.4.2022 Dostupné z: https://www.sciencedirect.com/science/article/pii/S0167404817301839?casa_token=2c5XmnEAXxgAAAAA:Kcno9zQBYcDmb8LXF074NfKCP11DZoHjrx0OQMKZNdJYoJT876J_kQ08lMcLaICjVUF1C0ARPA
16. What is Security Operations Center (SOC)? [online] 15.4.2022 Dostupné z: https://www.trellix.com/en-us/security-awareness/operations/what-is-soc.html
17. Understanding Threat Intelligence Use Cases. [online] 3.5.2022 Dostupné z: https://www.snapt.net/blog/understanding-threat-intelligence-use-cases
18. Python Forensic – Indicators of Compromise. [online] 14.4.2022 Dostupné z: https://www.tutorialspoint.com/python_forensics/python_forensics_compromise_indicators.htm
19. Indicators of Compromise. [online] 23.2.2022 Dostupné z: https://www.trendmicro.com/vinfo/us/security/definition/indicators-of-compromise
20. Indicators of Compromise (IoC). [online] 28.4.2022 Dostupné z: https://encyclopedia.kaspersky.com/glossary/indicator-of-compromise-ioc/
21. Haber, M.J., Rolls, D. (2020). Indicators of Compromise. In: Identity Attack Vectors. Apress, Berkeley, CA. [online] 18.4.2022 Dostupné z: https://link.springer.com/chapter/10.1007/978-1-4842-5165-2_9
22. Xiaojing Liao, Kan Yuan, XiaoFeng Wang, Zhou Li, Luyi Xing (2016). Acing the IOC Game: Toward Automatic Discovery and Analysis of Open-Source Cyber Threat Intelligence [online] 1.5.2022 Dostupné z: https://dl.acm.org/doi/abs/10.1145/2976749.2978315

23. The Concept of Pyramid of Pain [online] 1.5.2022 Dostupné z: https://cyware.com/educational-guides/cyber-threat-intelligence/the-concept-of-pyramid-of-pain-f358

24. David J. Bianco. The Pyramid of Pain [online] 1.5.2022 Dostupné z: https://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html

25. Marsha D. Rowell. 2017. Cyber indicators of compromise: a domain ontology for security information and event management [online] 3.5.2022 Dostupné z: https://apps.dtic.mil/sti/pdfs/AD1046101.pdf

26. What are Indicators of Compromise (IoCs) Used For [online] 22.4.2022 Dostupné z: https://www.lifars.com/2020/05/what-are-indicators-of-compromise-iocs-used-for/

27. Malwerbytes Labs, 2017. Explained: Spora ransomware. [online] 16.5.2022 Dostupné z: https://blog.malwarebytes.com/threat-analysis/2017/03/spora-ransomware/

28. Fuzzy Hashing with SSDEEP. [online] 16.5.2022 Dostupné z: https://dfir.science/2017/07/How-To-Fuzzy-Hashing-with-SSDEEP-(similarity-matching).html

29. VirusTotal [online] 1.5.2022 Dostupné z: https://www.virustotal.com/gui/file/7ad9ed23a91643b517e82ad5740d24eca16bcae21cfe1c0da78ee80e0d1d3f02/behavior

30. IOC Editor. [online] 3.5.2022 Dostupné z: https://www.fireeye.com/services/freeware/ioc-editor.html

31. Will Gibb, Devon Kerr. 2013. OpenIOC: Back to Basics [online] 25.3.2022 Dostupné z: https://www.mandiant.com/resources/openioc-basics

32. What is Open Indicators of Compromise (OpenIOC) framework?. [online] 15.4.2022. Dostupné z: https://cyware.com/educational-guides/cyber-threat-intelligence/what-is-open-indicators-of-compromise-openioc-framework-ed9d

33. OpenIOC. [online] 15.4.2022. Dostupné z: https://github.com/fireeye/OpenIOC_1.1

34. Introduction to STIX. [online] 3.5.2022. Dostupné z: https://oasis-open.github.io/cti-documentation/stix/intro.html#why-should-you-care

35. About CybOX. [online] 17.5.2022. Dostupné z: https://cybox.mitre.org/about/

36. Indicators of compromise as way to reduce risk. [online] 3.5.2022. Dostupné z: https://securelist.com/indicators-of-compromise-as-a-way-to-reduce-risk/71915/

37. OpenIOC to STIX. [online] 3.5.2022. Dostupné z: https://github.com/STIXProject/openioc-to-stix

38. STIX sample. [online] 13.5.2022. Dostupné z: http://saisa.eu/blogs/Guidance/wp-content/uploads/2014/11/stix-sample.png

39. CybOX_IPv4Adress_Instance.xml. [online] 17.5.2022. Dostupné z: https://github.com/CybOXProject/schemas/blob/master/samples/CybOX_IPv4Address_Instance.xml

40. InQuest – python-iocextract. [online] 12.5.2022. Dostupné z: https://github.com/inquest/python-iocextract

41. Ninoseki – ioc-extractor. [online] 12.5.2022. Dostupné z: https://github.com/ninoseki/ioc-extractor

42. Sroberts - cacador. [online] 12.5.2022. Dostupné z: https://github.com/sroberts/cacador

43. PaloAltoNetworks – ioc-parser. [online] 12.5.2022. Dostupné z: https://github.com/PaloAltoNetworks/ioc-parser

44. IOCParser. [online] 12.5.2022. Dostupné z: https://iocparser.com/

45. LOKI. [online] 12.5.2022. Dostupné z: https://github.com/Neo23x0/Loki

46. THOR Lite. [online] 12.5.2022. Dostupné z: https://www.nextron-systems.com/thor-lite/

47. THOR. [online] 12.5.2022. Dostupné z: https://www.nextron-systems.com/thor/

48. Redline. [online] 12.5.2022. Dostupné z: https://www.fireeye.com/services/freeware/redline.html

49. Kaspersky Log Scanner. [online] 12.5.2022. Dostupné z: https://support.kaspersky.com/CyberTrace/1.0/en-US/171647.htm

50. Yeti. [online] 12.5.2022. Dostupné z: https://yeti-platform.github.io/

51. OpenCTI. [online] 12.5.2022. Dostupné z: https://github.com/OpenCTI-Platform/opencti

52. Kaspersky CyberTrace. [online] 12.5.2022. Dostupné z: https://www.kaspersky.com/enterprise-security/cybertrace-threat-intelligence

53. MISP. [online] 12.5.2022. Dostupné z: https://www.misp-project.org/

54. TheHive. [online] 12.5.2022. Dostupné z: https://thehive-project.org/

55. TheSleuthkit. [online] 16.5.2022. Dostupné z: https://www.sleuthkit.org/

56. Hybrid-Analysis. [online] 16.5.2022. Dostupné z: https://www.hybrid-analysis.com/

57. Shodan. [online] 16.5.2022. Dostupné z: https://www.shodan.io

58. David Trounce. 2020. What Are Inodes in Linux and How Are They Used? [online] 15.5.2022. Dostupné z: https://helpdeskgeek.com/linux-tips/what-are-inodes-in-linux-and-how-are-they-used/

59. Williballenthin - Python-evtx. [online] 13.5.2022. Dostupné z: https://github.com/williballenthin/python-evtx

60. mmls. [online] 17.5.2022. Dostupné z: http://www.sleuthkit.org/sleuthkit/man/mmls.html

61. tsk_gettimes. [online] 17.5.2022. Dostupné z: http://www.sleuthkit.org/sleuthkit/man/tsk_gettimes.html

62. GCK´S FILE SIGNATURE TABLE. [online] 13.5.2022. Dostupné z: https://www.garykessler.net/library/file_sigs.html

Prílohy

Príloha A: Bakalárska práca v elektronickej podobe, prílohy v elektronickej podobe.

Príloha B: Zdrojový kód nástroja na spracovanie indikátorov kompromitácie

Príloha C: Návod na použitie