SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Evidenčné číslo: FEI-16607-92065

DETEKCIA COMMAND AND CONTROL (C2) KOMUNIKÁCIE

DIPLOMOVÁ PRÁCA

2023 **Bc. Martin Kubečka**

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Evidenčné číslo: FEI-16607-92065

DETEKCIA COMMAND AND CONTROL (C2) KOMUNIKÁCIE

DIPLOMOVÁ PRÁCA

Študijný program:	Aplikovaná informatika
Názov študijného odboru:	Informatika
Školiace pracovisko:	Ústav informatiky a matematiky FEI STU
Vedúci záverečnej práce:	Mgr. Martin Bečka, PhD.
Konzultant:	doc. Ing. Milan Vojvoda, PhD.

Bratislava 2023 **Bc. Martin Kubečka**

Slovenská technická univerzita v Bratislave
Ústav informatiky a matematiky

Fakulta elektrotechniky a informatiky
Akademický rok: 2022/2023
Evidenčné číslo: FEI-16607-92065

ZADANIE DIPLOMOVEJ PRÁCE

Študent:	**Bc. Martin Kubečka**
ID študenta:	92065
Študijný program:	aplikovaná informatika
Študijný odbor:	informatika
Vedúci práce:	Mgr. Martin Bečka, PhD.
Vedúci pracoviska:	doc. Ing. Milan Vojvoda, PhD.
Konzultant:	doc. Ing. Milan Vojvoda, PhD.
Miesto vypracovania:	Ústav informatiky a matematiky FEI STU

Názov práce: **Detekcia Command and Control (C2) komunikácie**

Jazyk, v ktorom sa práca vypracuje: slovenský jazyk

Špecifikácia zadania:

Infraštruktúra Command and Control, tiež známa ako C2 alebo C&C, je súbor nástrojov a techník, ktoré útočníci používajú na udržanie komunikácie s napadnutými zariadeniami po fáze počiatočnej exploitácie. C2 je dostupná v mnohých rôznych formách, z ktorých množstvo techník pozorujeme pri súčasných kybernetických útokoch.

Úlohy:
1. Analyzujte voľne dostupné Command and Control (C2) frameworky a ich techniky na komunikáciu.
2. Implementujte vybrané vhodné existujúce techniky a pravidlá ako aj vlastné pravidlá do programu na detekciu indikátorov C2 komunikácie.
3. Vyhodnoťte realizované riešenie, jeho výhody a nevýhody v porovnaní s inými riešeniami.

Zoznam odbornej literatúry:

1. C. Sanders, J. Smith: Applied Network Security Monitoring: Collection, Detection, and Analysis, Elsevier Science 2013, ISBN 9780124172166.
2. C. Sanders: Practical Packet Analysis, 3rd Edition: Using Wireshark to Solve Real-World Network Problems, No Starch Press 2017, ISBN 9781593278021.
3. R. Bejtlich: The Practice of Network Security Monitoring: Understanding Incident Detection and Response, No Starch Press 2013, ISBN 9781593275099.

Termín odovzdania diplomovej práce:	12. 05. 2023
Dátum schválenia zadania diplomovej práce:	05. 05. 2023
Zadanie diplomovej práce schválil:	prof. Dr. Ing. Miloš Oravec – garant študijného programu

SÚHRN

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Študijný program:	Aplikovaná informatika
Autor:	Bc. Martin Kubečka
Diplomová práca:	Detekcia Command and Control (C2) komunikácie
Vedúci záverečnej práce:	Mgr. Martin Bečka, PhD.
Konzultant:	doc. Ing. Milan Vojvoda, PhD.
Miesto a rok predloženia práce:	Bratislava 2023

Zámerom práce bolo navrhnúť aplikáciu a implementovať vhodné existujúce techniky ako aj vlastné metódy na detekciu indikátorov Command and Control komunikácie. V úvodných kapitolách práce sme priblížili indikátory kompromitácie a ich kategorizáciu pomocou diagramu Pyramid of Pain. Ďalej sme popísali procesy Cyber Threat Intelligence, ktorých výstupom sú informácie o kybernetických hrozbách, ktoré môžeme okrem iného použiť aj na detekciu Command and Control komunikácie. Životný cyklus kybernetického útoku, ktorého súčasťou je fáza Command and Control, sme predstavili prostredníctvom rámca kybernetickej bezpečnosti s názvom Cyber Kill Chain a MITRE ATT&CK matice. Následne sme popísali techniky, ktoré poskytujú tri vybrané voľne dostupné Command and Control frameworky s otvoreným zdrojovým kódom, Sliver, Merlin a Mythic. V ďalšej časti práce sme analyzovali dostupné riešenia, na základe ktorých sme definovali používateľské požiadavky pre našu aplikáciu C2Detective. Vychádzajúc z analytickej časti a definovaných funkcionálnych a nefunkcionálnych požiadaviek sme predstavili návrhovú a systémovú špecifikáciu aplikácie C2Detective, pričom implementačnú časť práce sme venovali jednotlivým detekčným technikám tejto aplikácie. V neposlednom rade sme testovali implementované detekčné techniky na vzorkách, ktoré obsahujú odchytenú sieťovú prevádzku Command and Control komunikácie. Na záver tejto práce sme vyhodnotili proces vývoja a implementácie našej aplikácie C2Detective a jej celkový prínos.

Kľúčové slová: Command and Control, Domain Generation Algorithms, DNS Tunneling, JA3, Cyber Kill Chain, MITRE ATT&CK, Pyramid of Pain

ABSTRACT

SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA
FACULTY OF ELECTRICAL ENGINEERING AND INFORMATION TECHNOLOGY

Study Programme:	Applied Informatics
Author:	Bc. Martin Kubečka
Master's thesis:	Detection of Command and Control (C2) communication
Supervisor:	Mgr. Martin Bečka, PhD.
Consultant:	doc. Ing. Milan Vojvoda, PhD.
Place and year of submission:	Bratislava 2023

The aim of this thesis was to design an application and implement appropriate existing techniques as well as our own methods for detecting Command and Control communication indicators. In the introductory chapters, we introduced the Indicators of Compromise and their categorization using the Pyramid of Pain diagram. Next, we described Cyber Threat Intelligence processes, which provide information about cyber threats that can be used to detect Command and Control communication, among other things. We presented the cyber attack lifecycle, which includes the Command and Control phase, through a cybersecurity framework called Cyber Kill Chain and the MITRE ATT&CK Matrix. We then described the techniques provided by three selected freely available open-source Command and Control frameworks, Sliver, Merlin and Mythic. In the next part of this thesis, we analyzed the available solutions, based on which we defined the user requirements for our C2Detective application. Based on the analytical part and the defined functional and non-functional requirements, we presented the design and system specification of the C2Detective application, while the implementation part of this thesis was dedicated to the individual detection techniques of this application. Finally, we tested the implemented detection techniques on samples containing captured network traffic of Command and Control communication. In the end, we concluded this work by evaluating the development and implementation process of our C2Detective application and its overall contribution.

Keywords: Command and Control, Domain Generation Algorithms, DNS Tunneling, JA3, Cyber Kill Chain, MITRE ATT&CK, Pyramid of Pain

Vyhlásenie autora

Podpísaný Bc. Martin Kubečka čestne vyhlasujem, že som diplomovú prácu Detekcia Command and Control (C2) komunikácie vypracoval samostatne na základe poznatkov získaných počas štúdia a informácií z dostupnej literatúry uvedenej v práci.

Diplomovú prácu som vypracoval pod vedením Mgr. Martina Bečku, PhD.

Bratislava, dňa 12. 05. 2023

_____

podpísaný autor

Poďakovanie

Ďakujem vedúcemu práce Mgr. Martinovi Bečkovi, PhD. a konzultantovi doc. Ing. Milanovi Vojvodovi, PhD. za cenné rady počas písania tejto práce, za priebežné stretnutia, podnety a pripomienky.

Obsah

Zoznam obrázkov a tabuliek

Zoznam skratiek

API	Application Programming Interface
APT	Advanced Persistent Threat
ASCII	American Standard Code for Information Interchange
BSD-3	Berkeley Software Distribution 3-Clause
C2	Command and Control
CLI	Command Line Interface
CPU	Central Processing Unit
CSV	Comma Separated Values
CTI	Cyber Threat Intelligence
DFIR	Digital Forensics and Incident Response
DGA	Domain Generation Algorithms
DNS	Domain Name System
EDR	Endpoint Detection and Response
GPLv3	GNU General Public License version 3
gRPC	Google Remote Procedure Call
GUI	Graphical User Interface
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IDS	Intrusion Detection System
IoCs	Indicators of Compromise
IPS	Intrusion Prevention System
JSON	JavaScript Object Notation
MaaS	Malware-as-a-Service
MD5	Message Digest Method 5
MITRE	Massachusetts Institute of Technology Research & Engineering
mTLS	Mutual Transport Layer Security
OISF	Open Information Security Foundation
OSINT	Open Source Intelligence Techniques
P2P	Peer-to-peer
PCAP	Packet Capture
PDF	Portable Document Format
PyPI	Python Package Index

RITA	Real Intelligence Threat Analysis
SHA	Secure Hash Algorithm
SIEM	Security Information and Event Management
SOC	Security Operations Center
SPI	Stateful Packet Inspection
SQL	Structured Query Language
SSH	Secure Shell
TCP	Transmission Control Protocol
TLD	Top-Level Domain
TLS	Transport Layer Security
Tor	The Onion Router
TTPs	Tactics, Techniques and Procedures
UDP	User Datagram Protocol
UI	User Interface
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
YAML	Yet Another Markup Language

Úvod

V kontexte kybernetickej bezpečnosti je *hrozbou* (z angl. threat) akákoľvek činnosť, ktorá predstavuje riziko narušenia ***dôvernosti*** (z angl. confidentiality), ***integrity*** (z angl. integrity) a ***dostupnosti*** (z angl. availability) informácií alebo systémov [1]. Kybernetické hrozby môžu pochádzať z rôznych zdrojov [2], ktorými sú napríklad kyberzločinci, štátom sponzorované skupiny škodlivých aktérov, tretie strany, resp. dodávatelia a v neposlednom rade aj zamestnanci, ktorí majú prístup k citlivým informáciám. Títo aktéri majú rôzne motívy pre ich škodlivé činnosti, ktoré môžu zahŕňať finančný zisk, špionáž a politické ciele, hacktivizmus (z angl. hacktivism), krádež duševného vlastníctva a v neposlednom rade kyberterorizmus (z angl. cyber terrorism).

Pri mnohých kybernetických útokoch používajú škodliví aktéri **Command and Control** (ďalej len „C2“) komunikáciu [3] na udržiavanie kontroly nad kompromitovanými systémami, alebo zariadeniami. C2 komunikácia umožňuje útočníkom vykonávať vzdialené príkazy na infikovaných systémoch a prijímať z nich údaje späť, bez vedomia používateľa daného systému. Takáto komunikácia môže prebiehať prostredníctvom rôznych kanálov [4] vrátane známych sieťových protokolov HTTP, alebo DNS, pomocou platforiem sociálnych sietí, alebo dokonca aj s využitím cloudových úložísk. Pokročilí útočníci využívajú sofistikovanejšie metódy, aby sa vyhli odhaleniu, ako napríklad používanie šifrovania, alebo maskovania ich škodlivých aktivít takými spôsobmi, aby sa daná sieťová komunikácia javila ako legitímna. Okrem toho môžu útočníci využívať na komunikáciu s infikovanými systémami viacero C2 kanálov súčasne, čo analytikom značne sťažuje identifikáciu a blokovanie takejto škodlivej sieťovej prevádzky. Detekcia Command and Control komunikácie by mala byť dôležitou súčasťou každej stratégie kybernetickej bezpečnosti z dôvodu, že napomáha rýchlejšie identifikovať a reagovať na prebiehajúce kybernetické útoky.

Techniky pre C2 komunikáciu a rôzne ďalšie funkcie, poskytujú škodlivým aktérom aplikácie, resp. softvérové nástroje, ktoré nazývame **C2 frameworky**. V našej práci analyzujeme vybrané voľne dostupné C2 frameworky a ich techniky, ktoré útočníci využívajú na komunikáciu s kompromitovanými systémami a na iné škodlivé účely. Na základe analyzovaných techník C2 frameworkov a spracovaných otvorených zdrojov implementujeme do vlastného nástroja s názvom **C2Detective** funkcionality na detekciu potencionálnych indikátorov C2 komunikácie.

Pre lepšie porozumenie rozsahu popisovanej problematiky sa v prvých častiach našej práce venujeme životnému cyklu kybernetických útokov, indikátorom, ktoré sú dôležitými vedľajšími produktmi takýchto útokov a informáciám o kybernetických hrozbách, ktoré

môžeme použiť na detekciu C2 komunikácie a všeobecne škodlivej aktivity. Následne si priblížime po technickej stránke C2 komunikáciu, analyzujeme dostupné techniky vybraných C2 frameworkov a popisujeme teoretický prístup k detekcii C2 komunikácie. Na základe analyzovaných techník vybraných C2 frameworkov, spracovaných otvorených zdrojov a v neposlednom rade definovaných funkcionálnych a nefunkcionálnych požiadaviek, popisujeme implementované metódy na detekciu potencionálnych indikátorov C2 komunikácie, ktoré sú obsiahnuté v našej aplikácii C2Detective. V závere tejto práce testujeme a vyhodnocujeme jednotlivé popísané a implementované metódy na detekciu C2 komunikácie.

1 Indikátory kompromitácie

Indikátory kompromitácie (z angl. Indicators of Compromise, IoCs) [5, 6] sú digitálny údaj forenzného charakteru, ktorý naznačuje, že mohlo dôjsť ku kompromitácii počítačovej siete, alebo koncového bodu. Tieto digitálne stopy napomáhajú odborníkom identifikovať škodlivú činnosť, alebo iné bezpečnostné hrozby, akými sú napríklad úniky citlivých informácii a malvérové útoky.

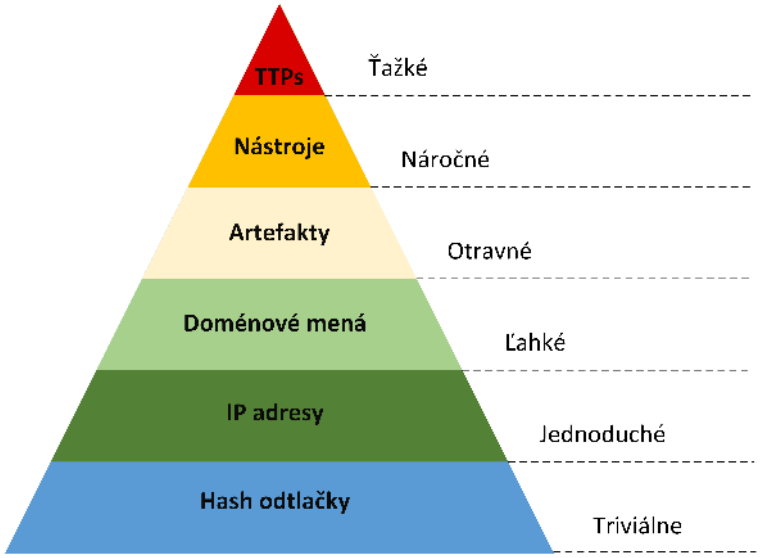
Monitorovanie indikátorov kompromitácie má *reaktívny* charakter. To znamená, že ak bol odhalený takýto indikátor, s veľkou istotou môžeme tvrdiť, že daný systém bol kompromitovaný. Rýchle odhalenie indikátorov kompromitácie môže napomôcť zastaviť kybernetické útoky v skoršej fáze ich životného cyklu, čo má za následok obmedzenie dopadu takéhoto útoku.

Taktiež poznáme **Indikátory útoku** (z angl. Indicators of Attack), ktoré sa na rozdiel od indikátorov kompromitácie, zameriavajú na identifikáciu prebiehajúcej aktivity útočníka. Ďalej skúmajú motiváciu a samotnú identitu škodlivého aktéra, pričom indikátory kompromitácie napomáhajú organizáciám len pochopiť udalosti, ktoré sa už odohrali. *Proaktívny* prístup k detekcii využíva oba typy uvedených indikátorov na odhalenie kompromitácie systémov, za čo najkratší čas.

Nárast sofistikovanosti škodlivých aktérov v kybernetickom priestore má za následok, že odhalenie indikátorov kompromitácie je čoraz náročnejšie. Medzi najčastejšie indikátory kompromitácie patrí napríklad hash odtlačok súboru, C2 IP adresa, alebo doménové meno, kľuč registra (z angl. registry key) a iné. Spomenuté indikátory sa neustále menia, čo sťažuje ich detekciu. Ako reakcia na túto skutočnosť vznikol diagram Pyramid of Pain, ktorý si ďalej bližšie priblížime.

1.1 Pyramid of Pain

Diagram s názvom **Pyramid of Pain** [7] predstavil v roku 2013 výskumník David J. Blanco po tom, ako v ten istý rok spoločnosť Mandiant zverejnila svoje zistenia ohľadne APT skupiny, konkrétne APT1. Správa o pôsobení tejto skupiny obsahovala niekoľko strán technických informácii a indikátorov, konkrétne *indikátorov kompromitácie*. Výskumník uvádza, že v dobe vytvorenia spomínaného diagramu, podľa jeho názoru, neboli tieto indikátory dostatočne efektívne používané.



Obr. 1: Diagram Pyramid of Pain

Uvedený diagram, ktorý môžeme vidieť na obrázku 1, znázorňuje vzťah medzi typmi indikátorov, ktoré môžeme použiť na odhalenie aktivít škodlivého aktéra a úrovne, akou skomplikujeme takémuto útočníkovi prácu, v prípade, že sa nám podarí tieto indikátory úspešne zablokovať. Najširšia časť pyramídy je sfarbená zelenou farbou a vrchol pyramídy je červený. Šírka aj farba danej časti diagramu sú veľmi dôležité pre pochopenie významu týchto typov ukazovateľov. Na úvod si definujeme typy ukazovateľov, ktoré tvoria popisovaný diagram.

1.1.1 Hash odtlačky

Väčšina hašovacích algoritmov vypočíta **hash odtlačok** správy z celého vstupu a na výstupe vytvorí hodnotu s pevnou dĺžkou, ktorá je jedinečná pre daný vstup. Inak povedané, ak sa obsah dvoch súborov líši čo i len v jedinom bite, výsledné hodnoty odtlačkov týchto dvoch súborov sú úplne odlišné. SHA-1 a MD5 algoritmy sú dva najbežnejšie používané príklady tohto typu. Popísané odtlačky považujeme za najpresnejšie typy indikátorov. Ako sme uviedli, výstupné hodnoty hašovacích funkcií sa dajú veľmi jednoducho zmeniť, a teda vo vybraných prípadoch je ich monitorovanie nevyhovujúce. V súčasnosti nie je odporúčané používať algoritmy SHA-1 a MD5 v oblasti kryptografie, pretože v minulosti boli objavené zraniteľnosti v predmetných algoritmoch. Avšak stále môžu byť použité na vytváranie hash odtlačkov pre uvedené účely.

Taktiež sa môžeme stretnúť s takzvanými *fuzzy hash* odtlačkami, ktoré sa tento problém snažia vyriešiť výpočtom hodnôt takých odtlačkov, ktoré zohľadňujú podobnosti na vstupe. Inak povedané, dva súbory s čo i len malými, alebo stredne veľkými rozdielmi,

by mali výstupné hodnoty hašovacej funkcie značne podobné, čo by analytikovi umožnilo zaznamenať možný vzťah medzi týmito súbormi.

1.1.2 IP adresy

IP adresy považujeme za najzákladnejší ukazovateľ. Na to aby mohol škodlivý aktér vykonať útok v prostredí počítačových sietí potrebuje sieťové pripojenie, teda IP adresu. Umiestnenie tohto ukazovateľa je v druhej najširšej časti pyramídy, pretože počet IP adries je príliš veľký. Priemerne pokročilý útočník mení IP adresy podľa potreby s veľmi malým úsilím, alebo používa anonymizované proxy služby akou je napríklad sieť Tor. Z uvedených dôvodov sú IP adresy v pyramíde označené zelenou farbou. Ak protivníkovi znemožníme používať jednu z jeho IP adries, zvyčajne vykoná spomenuté úkony bez prerušenia jeho činnosti.

1.1.3 Doménové mená

O stupeň vyššie v pyramíde sú bledo-zelenou farbou **doménové mená**. Ich zmena je o niečo náročnejšia ako pri IP adresách z dôvodu, že je potrebná ich registrácia, zaplatenie a v neposlednom rade ich umiestnenie. Vo svete existuje veľké množstvo poskytovateľov DNS služieb s nepostačujúcimi štandardmi pri procese registrácie, čo má za následok jednoduchší proces ich zmeny.

1.1.4 Artefakty hostiteľského systému a počítačovej siete

V strede pyramídy, ktorá začína bledo-žltou farbou, sa nachádzajú takzvané **artefakty hostiteľského systému a počítačovej siete** (z angl. Network & Host Artifacts). Na tejto úrovni začínajú mať naše postupy určitý negatívny vplyv na aktivity útočníka. Takéto pozorovateľné objekty, teda artefakty, sú spôsobené aktivitami protivníka v kompromitovanom prostredí. Z technického hľadiska by mohol byť sieťovým artefaktom každý bajt, ktorý pretečie počítačovou sieťou v dôsledku interakcií protivníka. V praxi takéto artefakty majú tendenciu odlišovať škodlivú činnosť útočníka od činnosti legitímnych používateľov. Typickými príkladmi pre sieťové artefakty môžu byť vzory URI, C2 informácie vložené do sieťových protokolov, charakteristické hodnoty HTTP hlavičky (napríklad User-Agent), alebo vybrané polia TLS certifikátu. Na druhej strane sú príklady artefaktov hostiteľského počítača, akými sú kľúče registrov, hodnoty, o ktorých je známe, že ich vytvárajú konkrétne časti škodlivého softvéru, súbory, alebo adresáre spustené na určitých miestach v systéme a v neposlednom rade názvy škodlivých služieb.

V prípade, že sa nám podarí odhaliť indikátory na tejto úrovni a zároveň sme schopní na ne reagovať, prinútime útočníka zmeniť jeho stratégiu, prekonfigurovať, alebo

prekompilovať jeho nástroje a podobne. Vhodným príkladom je, keď útočníkov nástroj určený na skenovanie systému používa pri vyhľadávaní obsahu špecifickú hodnotu HTTP hlavičky User-Agent. Ak zablokujeme všetky požiadavky, ktoré obsahujú túto špecifickú hodnotu uvedenej HTTP hlavičky, zamedzíme útočníkovi vo vykonávaní jeho aktivity. Zmena, alebo oprava HTTP hlavičky môže byť triviálna, ale napriek tomu musí útočník vynaložiť nejaké úsilie na identifikáciu a prekonanie našej prekážky.

1.1.5 Nástroje

Ďalšiu žltú úroveň predstavujú **nástroje**. Konkrétne hovoríme o softvéri, ktorý protivník používa na splnenie svojich úloh. Vo väčšine prípadov ide o programy, ktoré si útočník prinesie do kompromitovaného prostredia so sebou, a teda nie taký softvér, alebo iné nástroje, ktoré sú už dostupné v tomto prostredí. Do tejto kategórie patria napríklad komponenty pre C2 komunikáciu, nástroje na lámanie hesiel a ďalšie.

Na tejto úrovni odoberáme útočníkovi možnosť používať jeden, alebo viac konkrétnych nástrojov z dôvodu, že sme odhalili ich artefakty, teda útočník musí použiť, alebo vytvoriť nový nástroj, prípadne modifikovať taký nástroj, ktorý sme detegovali. Medzi príklady ukazovateľov škodlivých nástrojov patria signatúry antivírusových programov, alebo takzvané Yara pravidlá, pomocou ktorých sme schopní nájsť variácie tých istých súborov aj s miernymi zmenami. Do tejto úrovne môžu patriť aj sieťovo orientované nástroje s charakteristickým komunikačným protokolom, kde by si zmena protokolu vyžadovala podstatné prepracovanie pôvodného nástroja.

1.1.6 Taktiky, techniky a postupy

Na samotnom vrchole pyramídy sa nachádzajú **taktiky, techniky a postupy** (z angl. Tactics, Techniques & Procedures, ďalej len „TTPs“). Keď reagujeme na tejto úrovni, pôsobíme priamo na postupy útočníka, nie proti nástrojom, ktoré používa. Ako príklad si môžeme uviesť prípad, kedy analyzujeme samotné kroky pri použití útoku s názvom *Pass-the-Hash* a nie nástroje, ktoré boli použité na vykonávanie tohto typu útoku. Ďalším príkladom je cielený phishingový útok (z angl. spearphishing) so škodlivou prílohou. Aj v tomto prípade nehovoríme o konkrétnych nástrojoch, pretože existuje ľubovoľný počet spôsobov, ako vytvoriť škodlivú prílohu emailovej správy. Ak dokážeme na tejto úrovni dostatočne rýchlo reagovať na protivníka, prinútime ho urobiť to najnáročnejšie, čím je zmena jeho taktík, techník a postupov.

2 Cyber Threat Intelligence

Threat Intelligence, alebo konkrétne **Cyber Threat Intelligence** (ďalej len „CTI“), zahŕňa údaje, ktoré prešli procesmi agregácie, spracovania a analýzy s cieľom pomôcť bezpečnostným tímom pochopiť správanie, motívy a zámery škodlivých aktérov. Tento typ údajov môže zahŕňať aj informácie, ktoré sú zhromažďované z rôznych zdrojov, ako sú napríklad SIEM, DFIR a v neposlednom rade OSINT. V boji proti kybernetickým hrozbám, sú informácie o ďalšom postupe útočníkov kľúčové pre zmenu našich postupov z *reaktívnych* na *proaktívne* [8, 9].

2.1 Životný cyklus Cyber Threat Intelligence

Životný cyklus CTI predstavuje proces transformácie nespracovaných údajov na *kontextualizované informácie*, ktoré sú zamerané na riešenie bezpečnostných incidentov.

Transformačný proces prebieha v šesťfázovom cykle [8, 10], ktorý poskytuje rámec pre optimalizáciu a zefektívňovanie reakcie na neustále sa meniace hrozby v kybernetickom priestore. V tejto časti si priblížime transformačný proces informácií, ktorý nám poskytuje CTI.

2.1.1 Požiadavky

Prvá fáza životného cyklu je kľúčová, pretože definuje plán pre konkrétnu operáciu. Počas tejto fázy sa určujú ciele a metodiky, na základe potrieb zúčastnených strán. Ako naše ciele si môžeme stanoviť napríklad, kto sú útočníci a aká je ich motivácia, aká je plocha útočných vektorov (z angl. attack surface), alebo aké konkrétne opatrenia by sa mali prijať na posilnenie obrany proti budúcim útokom.

2.1.2 Zhromažďovanie

Po definovaní požiadaviek môžeme začať so zhromažďovaním informácií potrebných na splnenie našich cieľov. V závislosti od stanovených cieľov, vyhľadávame informácie v záznamoch o sieťovej prevádzke, verejne dostupné údaje z internetových fór, sociálnych sietí a podobne.

2.1.3 Spracovanie

Po zozbieraní nespracovaných údajov je potrebné pripraviť dáta do formátu vhodného na analýzu. Proces spracovania môže zahŕňať napríklad usporiadanie dát do tabuliek, preklad informácií zo zahraničných zdrojov a vyhodnotenie údajov z hľadiska relevantnosti a spoľahlivosti.

2.1.4 Analýza

Po spracovaní zhromaždených údajov sa vykoná dôkladná analýza s cieľom nájsť odpovede na otázky, ktoré boli definované v prvej fáze cyklu. Rozhodnutia, ktoré sa môžu prijať, zahŕňajú napríklad analýzu potenciálnej hrozby prostredníctvom odhalenia známych vzorov kybernetických útokov.

2.1.5 Šírenie

Fáza šírenia informácií si vyžaduje, aby bola analýza pretransformovaná do formátu, ktorý prehľadne prezentuje výsledky zainteresovaným stranám. Prezentácia analýzy by mala byť vo väčšine prípadov stručná, bez nadmerného použitia technických odborných názvov.

2.1.6 Spätná väzba

Záverečná fáza životného cyklu zahŕňa získavanie spätnej väzby na pretransformovanú analýzu s cieľom určiť, či je potrebné vykonať nejaké úpravy pre budúce operácie. Napríklad môže nastať zmena priorít zainteresovaných strán, zmena požadovaného spôsobu šírenia, alebo prezentácie údajov.

2.2 Klasifikácia Cyber Threat Intelligence

Informácie, ktoré nám poskytujú procesy pre CTI môžu byť na jednej strane komplexné, akými je napríklad detailný profil škodlivého aktéra, alebo na druhej strane menej komplexné, ako napríklad známa škodlivá IP adresa, doménové meno, URL adresa a podobne. V tejto časti si CTI rozdelíme na 3 kategórie [8, 11], ktoré si bližšie priblížime.

2.2.1 Taktické informácie

Taktické informácie majú technický charakter a predstavujú *indikátory kompromitácie*, akými sú napríklad škodlivé IP adresy, doménové mená, hash odtlačky škodlivých súborov, ale aj známe škodlivé URL adresy. Tento typ informácií je často spracovaný vo formáte, ktorý je možné ľahko integrovať do našich bezpečnostných riešení, čo v praxi znamená, že vybrané bezpečnostné produkty dokážu prijímať tieto informácie napríklad pomocou dostupného API rozhrania.

Kategória taktických informácií patrí k jednoduchším na vytváranie, pričom tento proces býva vo väčšine prípadov *automatizovaný*. Informácie tohto charakteru mávajú kratšiu životnosť z dôvodu, že škodliví aktéri dokážu relatívne rýchlo meniť detegované indikátory. Taktiež je potrebné uviesť, že odoberanie veľkého množstva *threat feedov* môže

viesť ku spracovávaniu menej relevantných informácií pre naše prostredie, ako aj navýšeniu výskytu *falošných pozitív* (z angl. false positive). Dôvodom tejto skutočnosti môže byť neaktuálny, alebo menej dôveryhodný zdroj informácií.

2.2.2 Operatívne informácie

Faktory, akými je **kto** (atribúcia), **ako** (TTPs) a **prečo** (zámer) vykonal daný útok, spolu vytvárajú *kontext*, ktorý poskytuje prehľad o tom, ako škodliví aktéri plánujú, vykonávajú a udržiavajú svoje operácie. Takýto prehľad následne predstavuje **operatívne informácie**.

Na rozdiel od taktických, operatívne informácie vyžadujú analýzu bezpečnostným odborníkom, ktorý spracuje tieto informácie do formátu, ktorý je ľahko použiteľný koncovými príjemcami. Napriek tomu, že si operatívne informácie vyžadujú viacero zdrojov, majú dlhšiu životnosť ako uvedené taktické informácie. Dôvodom je, že útočníci nedokážu za krátky čas meniť svoje TTPs, tak ľahko ako IP adresu, alebo doménové meno.

Operatívne informácie sú najužitočnejšie pri vykonávaní reakcie na kybernetické incidenty, monitorovaní bezpečnostných hrozieb, ako aj pri každodenných operáciách v bezpečnostnom centre SOC.

2.2.3 Strategické informácie

Škodliví aktéri môžu vykonávať svoju aktivitu, ako reakciu na faktory, medzi ktoré patrí napríklad geopolitická situácia. **Strategické informácie** poukazujú na to, ako môžu globálne udalosti a zahraničné politiky ovplyvňovať kybernetický priestor.

Tento typ informácií napomáha osobám s rozhodovacími právomocami, lepšie pochopiť riziká, ktoré predstavujú kybernetické hrozby. Na základe pochopenia týchto rizík, dokážu dané osoby realizovať kroky v ich vlastný alebo spoločenský prospech, ktoré sú v súlade s ich strategickými prioritami. Spomedzi uvedených kategórií, sú strategické informácie najnáročnejšie na vytvorenie. Tieto informácie si vyžadujú zber a analýzu dát z oblasti kybernetickej bezpečnosti a geopolitickej situácie vo svete špecializovanými odborníkmi, pričom strategické informácie sa zvyčajne získavajú vo forme správ.

2.3 Prípady použitia Cyber Threat Intelligence

Popisované informácie sú využívané v rôznych odvetviach kybernetickej bezpečnosti [12]. V bezpečnostných centrách SOC, analytici využívajú informácie, konkrétne taktické informácie, integrované do rôznych bezpečnostných riešení. Takáto kombinácia umožňuje identifikovať kybernetické hrozby v reálnom čase, ako napríklad C2 komunikáciu a taktiež

zlepšiť schopnosť odhaľovania a reakcie na kybernetické útoky. Ďalším praktickým využitím je riešenie kybernetických incidentov, kedy môžeme tieto informácie využiť na identifikáciu zdroja daného útoku, ako aj na prijatie vhodných opatrení voči podobným hrozbách v budúcnosti. V oblasti riadenia rizík, využívame popisované informácie na posudzovanie rizík, ktoré sú spojené s konkrétnou hrozbou a následne pri vypracovávaní stratégie na zmiernenie dopadu daného rizika. CTI je taktiež možné použiť na identifikáciu systémových zraniteľností a pri určovaní priority ich eliminácie. V neposlednom rade môžeme takéto informácie využiť pri vzdelávaní používateľov o potenciálnych kybernetických hrozbách a spôsoboch obrany voči nim.

3 Rámce kybernetickej bezpečnosti

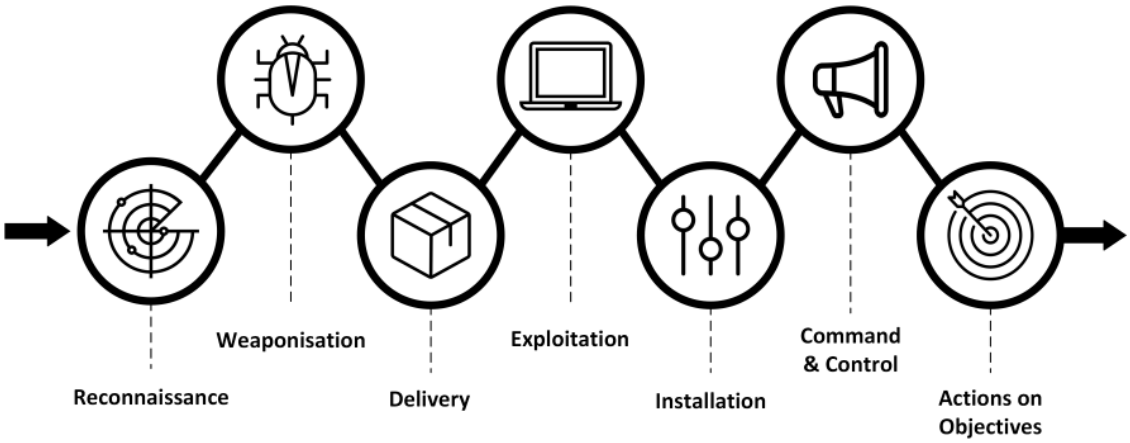
Pre lepšie porozumenie samotného procesu detekcie C2 komunikácie, je vopred potrebné definovať jednotlivé fázy kybernetického útoku pomocou vybraného rámca **Cyber Kill Chain** a priblížiť taktiky škodlivých aktérov prostredníctvom rámca **MITRE ATT&CK**. Vo všeobecnosti rámce kybernetickej bezpečnosti predstavujú štruktúrovaný prístup k identifikácii, vyhodnocovaniu a riadeniu kybernetických hrozieb a zraniteľností, pričom nám môžu taktiež poskytovať spôsoby, ako efektívne zavádzať kontrolné mechanizmy kybernetickej bezpečnosti.

3.1 Cyber Kill Chain

Cyber Kill Chain [13] je adaptáciou vojenského modelu, ktorý postupnými krokmi identifikuje a eliminuje aktivity nepriateľa. Tento cyklus pôvodne navrhla spoločnosť *Lockheed Martin* v roku 2011, pričom jeho obsah popisuje fázy kybernetického útoku a je určený na obranu proti APT skupinám. Tieto skupiny využívajú sofistikované techniky na získanie prístupu do systému a na zotrvanie v kompromitovanej infraštruktúre, čo najdlhší čas. Ich útoky najčastejšie zahŕňajú kombináciu malvéru, ransomvéru, trójskych koní, spoofingu a techník sociálneho inžinierstva na uskutočnenie svojho plánu.

3.1.1 Anatómia modelu Cyber Kill Chain

Pôvodný model Cyber Kill Chain od spoločnosti Lockheed Martin obsahuje sedem krokov [13, 14], ktoré môžeme vidieť na obrázku 2.



Obr. 2: Anatómia modelu Cyber Kill Chain

1. **Reconnaissance**: Počas prvej fázy kybernetického útoku, škodlivý aktér identifikuje

cieľ a skúma zraniteľnosti a slabiny, ktoré môže použiť ako potenciálny útočný vektor. V rámci tohto procesu útočník získava prihlasovacie údaje, alebo zhromažďuje iné informácie, akými sú emailové adresy, identifikátory používateľov, fyzické umiestnenie cieľa, využívané softvérové aplikácie a v neposlednom rade podrobnosti o operačnom systéme. Čím viac informácií je útočník schopný zhromaždiť počas prvej fázy kybernetického útoku, tým sofistikovanejší a presvedčivejší bude tento útok, a teda aj jeho vyššia pravdepodobnosť úspechu.

2. **Weaponisation**: V druhej fáze cyklu útočník vytvorí vektor útoku, napríklad malvér so vzdialeným prístupom, ktorý dokáže zneužiť identifikovanú zraniteľnosť z prvej fázy cyklu. Počas tejto fázy si môže útočník vytvoriť takzvaný *backdoor*, s cieľom pokračovať v prístupe do kompromitovaného systému aj v prípade, že jeho prvotný prístup bol detegovaný.

3. **Delivery**: V treťom kroku útočník iniciuje útok. Konkrétny postup závisí od typu útoku, ktorý má škodlivý aktér v úmysle vykonať. Útočník môže napríklad rozoslať emailové správy so škodlivými prílohami. Táto činnosť môže byť kombinovaná s technikami sociálneho inžinierstva, aby sa zvýšila účinnosť danej kampane.

4. **Exploitation**: Škodlivý kód je v štvrtej fáze spustený na cieľových systémoch. Po prelomení bezpečnostného perimetra má útočník možnosť ďalej kompromitovať systémy prostredia, napríklad inštaláciou dodatočných nástrojov, spúšťaním škodlivých skriptov alebo úpravou kryptografických certifikátov.

5. **Installation**: Bezprostredne po fáze exploitácie, útočník nainštaluje škodlivý softvér, alebo použije iné útočné vektory. V tomto bode cyklu, škodlivý aktér vstúpil do systému a ďalej sa snaží prevziať úplnú kontrolu nad kompromitovaným prostredím.

6. **Command & Control**: Vo fáze Command & Control, útočník prevzal vzdialenú kontrolu nad zariadením, alebo identitou v cieľovej počítačovej sieti. Taktiež môže škodlivý aktér v tejto fáze pracovať na *laterálnom pohybe* (z angl. lateral movement) v kompromitovanom prostredí a zároveň vytvárať ďalšie body vzdialeného prístupu.

7. **Actions on Objectives**: V poslednej fáze škodlivý aktér vykonáva činnosti týkajúce sa jeho plánovaných cieľov, ktoré môžu zahŕňať krádež, šifrovanie, zničenie, alebo exfiltráciu citlivých údajov.

Popísaný cyklus bol časom rozšírený o ôsmy krok, ktorý bol nazvaný **Monetization**. V tejto fáze sa škodlivý aktér zameriava na získanie príjmu z vykonaného útoku, či

už prostredníctvom výkupného, ktoré má obeť zaplatiť, alebo predajom exfiltrovaných citlivých informácií, ako sú osobné údaje, alebo obchodné tajomstvá.

Útoky, ktoré sa dostanú do fázy **Command & Control**, si vyžadujú pokročilejšie nápravné opatrenia, vrátane hĺbkovej kontroly počítačovej siete a koncových bodov s cieľom určiť rozsah takéhoto útoku. Organizácie by mali podniknúť kroky na identifikáciu kybernetických hrozieb, čo najskôr v ich životnom cykle z dôvodu, aby minimalizovali riziko útoku, ako aj náklady na riešenie takejto udalosti.

Kroky, resp. techniky útočníka, ktoré tvoria definovaný model Cyber Kill Chain, sme pre prehľad zhrnuli v tabuľke 1, pričom uvádzame názov techniky, jej účel a pre každú túto techniku taktiež niekoľko príkladov.

3.1.2 Využitie Cyber Kill Chain v praxi

Napriek niektorým nedostatkom zohráva popísaný model Cyber Kill Chain dôležitú úlohu pri definovaní stratégie kybernetickej bezpečnosti [13]. Dostupné služby a riešenia, ktoré implementujú popísaný model nám umožňujú

- reagovať na kybernetické útoky v reálnom čase, teda odhaliť útočníkov v každej fáze životného cyklu pomocou výstupov zo CTI procesov,
- vo všeobecnosti zabrániť prístupu do systémov neoprávneným používateľom,
- zabrániť zdieľaniu, ukladaniu, zmene, exfiltrácii, alebo šifrovaniu citlivých údajov neoprávnenými používateľmi,
- zastaviť laterálny pohyb útočníka v rámci počítačovej siete.

Technika	Účel	Príklady
Reconnaissance	Identifikácia cieľa a skúmanie jeho zraniteľností a bezpečnostných slabín.	Zber emailových adries, enumerácia sociálnych sietí, skenovanie cieľových systémov a iné.
Weaponisation	Vytvorenie útočného vektoru na základe potrieb a zámerov útočníka.	Malvér so vzdialeným prístupom, škodlivá príloha emailovej správy, prihlasovacie údaje pre vzdialený prístup a ďalšie.
Delivery	Spôsob doručenia škodlivého kódu do cieľového prostredia.	Emailová správa, pamäťové médium, škodlivá webová adresa a iné.
Exploitation	Zneužitie zraniteľností systému obete s cieľom spustenia škodlivého kódu.	ProxyLogon (CVE-2021-26855), ZeroLogon (CVE-2020-1472), Log4Shell (CVE-2021-44228), Follina (CVE-2022-30190) a ďalšie.
Installation	Inštalácia škodlivého softvéru a iných nástrojov na získanie prístupu do kompromitovaného systému.	Trójsky kôň so vzdialeným prístupom, technika *password dumping* pomocou nástroja Mimikatz a iné.
Command & Control	Vzdialené ovládanie kompromitovaného systému, inštalácia ďalších škodlivých programov, eskalácia systémových oprávnení.	Cobaltstrike, Empire, Covenant, Metasploit, Merlin, Sliver, Mythic, SilentTrinity a mnoho ďalších.
Actions on Objectives	Uskutočnenie plánovaných cieľov škodlivého aktéra.	Krádež, šifrovanie, zničenie alebo exfiltrácia citlivých údajov.

Tabuľka 1: Techniky modelu Cyber Kill Chain

3.2 MITRE ATT&CK

MITRE je nezisková organizácia, ktorá vznikla so zámerom poskytovať technické poradenstvo vláde Spojených štátov amerických. Rámec **MITRE ATT&CK** [15] vznikol v roku 2013 ako výsledok experimentu s názvom *MITRE Fort Meade Experiment*. V rámci tohto experimentu, výskumníci napodobňovali správanie útočných a obranných tímov s cieľom zlepšiť efektívnosť detekcie hrozieb po fáze kompromitácie, prostredníctvom analýzy správania škodlivých aktérov. Kľúčovou otázkou pre výskumníkov bolo, s akou úspešnosťou sme schopní odhaľovať zdokumentované správanie útočníkov. Na zodpovedanie tejto otázky výskumníci vyvinuli rámec s názvom ATT&CK, ktorý bol použitý ako nástroj určený na kategorizáciu správania škodlivých aktérov.

Rámec MITRE ATT&CK je databáza poznatkov, ktorá agreguje taktiky a techniky používané škodlivými aktérmi vo všetkých fázach modelu Cyber Kill Chain. Tento rámec bol verejnosti bezplatne sprístupnený v roku 2015 a v súčasnosti napomáha bezpečnostným tímom v rôznych odvetviach zabezpečiť organizácie proti známym a novo vznikajúcim hrozbám v kybernetickom priestore. Napriek tomu, že sa rámec MITRE ATT&CK pôvodne zameriaval výhradne na hrozby proti operačnému systému Windows, aktuálne jeho databáza zahŕňa aj operačný systém macOS a Linux, ako aj mobilné zariadenia a priemyselné riadiace systémy (z angl. Industrial control system) [16].

3.2.1 Taktiky a techniky

Konkrétni škodliví aktéri majú tendenciu používať špecifické *techniky*. Rámec MITRE ATT&CK obsahuje informácie, ktoré spájajú skupiny takýchto útočníkov s kampaňami, čo umožňuje bezpečnostným tímom lepšie pochopiť protivníkov, vyhodnotiť obranné postupy a posilniť bezpečnosť na relevantných miestach. **Matica MITRE ATT&CK** (z angl. MITRE ATT&CK Matrix) obsahuje súbor techník používaných protivníkmi na dosiahnutie konkrétneho cieľa. Tieto ciele sú v matici kategorizované ako *taktiky*. Ciele útočníkov sú prezentované lineárne od počiatočnej fázy **Reconnaissance**, až po konečnú fázu **Exfiltration**, resp. po fázu **Impact** [16].

Taktiky protivníkov sú špecifické technické ciele, ktoré chce protivník dosiahnuť. Tieto taktiky sú rozdelené do kategórií podľa ich cieľov. Ako príklad, nižšie uvádzame štrnásť taktík [17], ktoré tvoria **Enterprise ATT&CK maticu** [18], ktorá je nadmnožinou matíc pre systémy Windows, macOS a Linux.

1. **Reconnaissance**: zhromažďovanie informácií o cieľovej organizácii.

2. **Resource Development**: vytvorenie prostriedkov na podporu útočných operácií,

teda príprava nástrojov a infraštruktúry škodlivého aktéra.

3. **Initial Access**: snaha preniknúť do počítačovej siete, napríklad prostredníctvom škodlivej prílohy v cielenej phishingovej emailovej správe.

4. **Execution**: pokus o spustenie škodlivého kódu, napríklad spustenie trójskeho koňa so vzdialeným prístupom na systéme obete.

5. **Persistence**: snaha udržať si prístup v kompromitovanej počítačovej sieti, napríklad zmenou špecifických konfigurácii.

6. **Privilege Escalation**: pokus o eskaláciu systémových oprávnení, napríklad prostredníctvom využitia zraniteľnosti v kompromitovanom prostredí.

7. **Defense Evasion**: snaha vyhnúť sa odhaleniu obrannými tímami, napríklad pomocou používania dôveryhodných procesov a nástrojov, na ukrytie škodlivého softvéru, alebo časti škodlivého kódu.

8. **Credential Access**: krádež prihlasovacích údajov, teda prihlasovacích mien a hesiel, napríklad prostredníctvom techniky *keylogging*.

9. **Discovery**: snaha preskúmať kompromitované prostredie, s cieľom získať poznatky o systéme a internej počítačovej sieti, pričom sú často používané natívne nástroje daného operačného systému.

10. **Lateral Movement**: pohyb medzi viacerými systémami v kompromitovanom prostredí s použitím legitímnych prihlasovacích údajov.

11. **Collection**: zbieranie údajov, ktoré sú zaujímavé pre útočníka, ako napríklad prístupy k údajom v cloudovom úložisku, pomocou metód ako zachytávanie snímok obrazovky, alebo spomenutou technikou keylogging.

12. **Command and Control**: proces, ktorý umožňuje komunikáciu a vzdialené riadenie kompromitovaných systémov škodlivým aktérom, s cieľom napodobňovania bežnej očakávanej sieťovej prevádzky, napríklad prostredníctvom HTTP protokolu.

13. **Exfiltration**: krádež citlivých údajov z kompromitovaného prostredia, ktorá môže zahŕňať kompresiu, šifrovanie a prenos takýchto údajov prostredníctvom C2 komunikačného kanálu s cieľom vyhnutia sa detekcie.

14. **Impact**: manipulácia, prerušenie dostupnosti, alebo narušenie integrity údajov, procesov, alebo celých systémov, so zámerom dosiahnutia cieľov útočníka.

Každá taktika matice MITRE ATT&CK obsahuje viacero techník útočníka. Je to z dôvodu, že škodliví aktéri používajú rôzne techniky v závislosti od faktorov, ako sú ich zručnosti, dostupnosť vhodných nástrojov, alebo konfigurácia systému, ktorý je cieľom daného útoku. Vybraná technika opisuje konkrétny spôsob, akým sa útočník môže pokúsiť dosiahnuť svoj cieľ.

Jednotlivé techniky obsahujú popis metódy a systémy, ktorých sa daná technika týka a taktiež, ktoré skupiny škodlivých aktérov ju používajú (v prípade ak je známa). Ďalej techniky zahŕňajú spôsoby zmiernenia dopadu danej aktivity a v neposlednom rade referencie na jej použitie v praxi.

Taktiky v matici sú označené jedinečným šesťmiestnym identifikátorom, ktorého počiatočné dve hodnoty sú *„TA“*, za ktorými nasleduje poradová hodnota danej taktiky. Ako príklad označenia si uvedieme taktiku s názvom **Command and Control**, ktorej identifikátor má hodnotu **TA0011**. Po vzore taktík, sú označované aj jednotlivé techniky útočníkov s rozdielom, že identifikátor je päťmiestny a jeho počiatočná hodnota je iba *„T“*. V prípade, že daná technika obsahuje aj takzvané *podtechniky* (z angl. sub-technique), tie sú označené pridaním poradového čísla oddeleného bodkou za identifikátorom príslušnej techniky. Ako príklad označenia si pre taktiku TA0011 uvedieme techniku s názvom **Application Layer Protocol**, s identifikátorom **T1071** a podtechnikou, ktorá má názov **Web Protocols** a identifikačné označenie **T1071.001**.

Po taktikách a technikách útočníkov je potrebné uviesť, že rámec MITRE ATT&CK taktiež obsahuje postupy, ktoré predstavujú detailné popisy krokov, ako protivník plánuje dosiahnuť svoje ciele.

Matica (Enterprise) ATT&CK verzia 12 v čase písania tejto práce obsahuje celkovo štrnásť taktík, sto deväťdesiat tri techník a štyristo jedna podtechník, ktorá ako sme uviedli obsahuje matice pre systémy Windows, macOS a Linux.

3.2.2 Využitie matice MITRE ATT&CK v praxi

Nižšie uvádzame niektoré zo spôsobov [19], pri ktorých môžu bezpečnostné tímy využiť rámec MITRE ATT&CK, resp. maticu MITRE ATT&CK.

1. **Emulácia škodlivého aktéra** (z angl. Adversary Emulation): Hodnotenie bezpečnosti použitím informácií o škodlivom aktérovi a jeho taktikách, technikách a postupoch s cieľom emulácie vybranej hrozby, teda konkrétneho útočníka.

2. **Posúdenie nedostatkov obranyschopnosti**: Nadobudnutie znalosti, ktoré časti infraštruktúry majú dostatočnú úroveň obrany, posúdenie existujúcich nástrojov, alebo

testovanie nových nástrojov pred ich zakúpením s cieľom určenia ich efektívnosti.

3. **Obohacovanie informácii**: Umožňuje posúdiť, či sme schopní brániť sa proti konkrétnym pokročilým pretrvávajúcim hrozbám, alebo škodlivému správaniu iných typov útočníkov.

4. **Vývoj behaviorálnej analýzy**: Zjednodušenie vzorov činnosti, ktorú považujeme za škodlivú, pomocou prepojenia medzi podozrivými aktivitami pri monitorovaní prostredia a činnosťami útočníka.

3.3 Porovnanie popísaných rámcov kybernetickej bezpečnosti

Prvý popísaný rámec s názvom **Cyber Kill Chain** využívame pri detekcii a vyhľadávaní hrozieb počas celého životného cyklu kybernetického útoku. Na druhej strane, rámec **MITRE ATT&CK**, ktorý je reprezentovaný ako matica techník, definuje postupnosť udalostí pri kybernetickom útoku.

V porovnaní s inými staršími rámcami kybernetickej bezpečnosti, ktoré v tejto práci neuvádzame, rámec MITRE ATT&CK indexuje všetko od útoku zo strany škodlivého aktéra, až po postupy pri obrane voči nim. Tieto zmapované scenáre útokov môžu replikovať *ofenzívne tímy* (z ang. red teams) a zároveň testovať efektivitu *defenzívnych tímov* (z angl. blue team) [16].

Taktiež je potrebné uviesť, že rámec MITRE ATT&CK je pravidelne aktualizovaný, čo by malo mať za následok, že aj defenzívne tímy aktualizujú svoje vlastné postupy, ako aj modelovanie vybraných útokov. V neposlednom rade uvádzame typy útokov, napríklad útoky cielené na cloudové riešenia, ktorých potenciálne útočné vektory sú menej relevantné pre staršie rámce kybernetickej bezpečnosti, z čoho vyplýva, že na rozdiel od rozšírení rámcu MITRE ATT&CK, nie sú takéto taktiky útočníkov zohľadnené.

4 Command and Control komunikácia

C2 frameworky, ako bolo spomenuté v úvode, predstavujú nástroje, ktoré umožňujú útočníkom vzdialene ovládať kompromitované systémy, teda umožňujú jednostrannú, alebo obojstrannú komunikáciu. **C2 komunikácia** zvyčajne zahŕňa jeden, alebo viacero skrytých kanálov, v závislosti od mechanizmov použitých pri konkrétnom útoku. C2 frameworky ďalej poskytujú funkcie na stiahnutie ďalšieho škodlivého kódu, alebo softvéru, funkcie na eskaláciu systémových privilégií, laterálny pohyb, alebo exfiltráciu údajov [20].

4.1 Architektúra Command and Control infraštruktúry

V základnej **centralizovanej architektúre** je C2 infraštruktúra [21] tvorená z dvoch komponentov a to *klienta* a *servera*. Komponent klienta je nainštalovaný v kompromitovanom prostredí a umožňuje škodlivému aktérovi komunikovať s infikovanými systémami na diaľku, zatiaľ čo komponent servera riadi komunikáciu medzi klientom a systémom útočníka. Z dôvodu, že príkazy pochádzajú z jedného zdroja, je takýto model architektúry jednoduchšie odhaliť a zablokovať. Škodliví aktéri preto využívajú proxy služby, alebo iný spôsob presmerovania sieťovej komunikácie, s cieľom maskovať pôvodnú zdrojovú IP adresu riadiaceho servera.

Iným druhom C2 infraštruktúry je takzvaná **P2P architektúra**, ktorá predstavuje *decentralizovaný* model. Na rozdiel od centralizovanej architektúry a riadiaceho servera, sú príkazy rozposielané medzi *uzlami*, čo poskytuje značnú redundanciu pri pokusoch o vyradenie danej infraštruktúry škodlivého aktéra. Architektúru P2P je možné využiť s centralizovaným modelom na vytvorenie *hybridnej konfigurácie architektúry*, kedy P2P zložka funguje ako záložný variant, v prípade, že riadiaci server je napadnutý, alebo vyradený z prevádzky.

V neposlednom rade je potrebné uviesť, že škodlivý aktér si môže zvoliť *ľubovoľný* spôsob komunikácie s kompromitovaným prostredím. Takýto model môže s infikovaným systémom komunikovať prostredníctvom rôznych zdrojov, ako sú napríklad chatovacie miestnosti, komentáre na sociálnych sieťach, alebo prostredníctvom platforiem na zdieľanie súborov. Využívaním dôveryhodných služieb škodliví aktéri zvyšujú svoje šance na úspech a znižujú šance na odhalenie ich škodlivej činnosti.

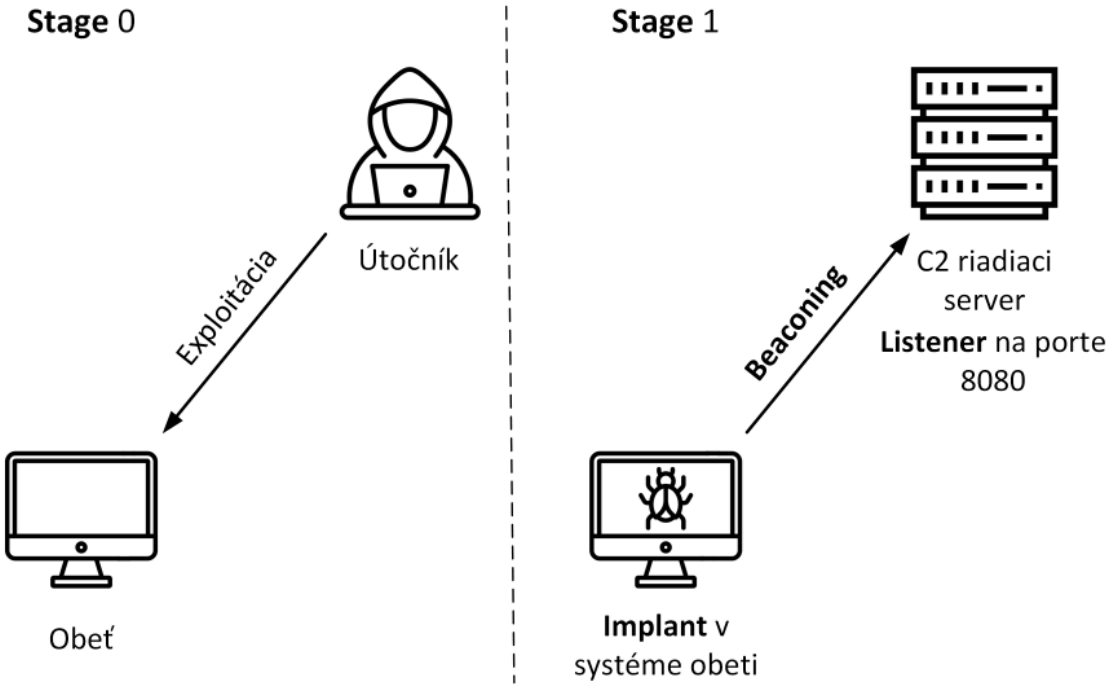
4.2 Porovnanie dostupných techník vybraných C2 frameworkov

C2 frameworky poskytujú rôzne techniky a funkcionality, ktoré napomáhajú útočníkom vyhnúť sa odhaleniu a vykonávať škodlivé aktivity bez detekcie obrannými tímami. Pri analýze otvorených zdrojov sme zistili, že vo všeobecnosti sú za najrozšírenejšie a najčastejšie používanie C2 frameworky považované **Cobalt Strike** [22], **Metasploit** [23], a **Empire** [24]. Použitie, ako aj možné postupy detekcie uvedených troch C2 frameworkov sú vo veľkej miere zdokumentované.

Z dôvodu, že sa v našej práci venujeme detekcii C2 komunikácie v počítačových sieťach, pri popise vybraných C2 frameworkom sa nebudeme venovať technikám a funkcionalitám, ktoré nie sú priamo spojené s komunikáciou medzi riadiacim serverom a kompromitovaným systémom.

Predtým, ako si priblížime techniky a funkcionality, ktoré poskytujú vybrané C2 frameworky je potrebné definovať viaceré pojmy [25], ktoré budeme ďalej v našej práci používať a zároveň ich môžeme vidieť aj na obrázku 3.

- **Listener**: program, alebo služba, ktorá je nakonfigurovaná tak, aby čakala na prichádzajúce spojenia z útočníkovho systému, resp. z kompromitovaného systému.
- **Implant**: časť softvéru, ktorá je používaná na udržiavanie prístupu do kompromitovaného prostredia, alebo systému, teda jedná sa o kód, ktorý vykonáva útok na cieľovom systéme, ako aj udržiava vzdialený prístup (pojem implant je často zameniteľný s pomenovaním **agent**).
- **Beacon/Beaconing**: označuje komunikačný vzor, pri ktorom sa implant v pravidelných, alebo nepravidelných intervaloch pripája k C2 riadiacemu serveru, na rozdiel od použitia sieťového spojenia v reálneho časom. V inom kontexte C2 frameworku môže pojem beacon taktiež zahŕňať aj definíciu komunikačného štýlu.
- **Stage**: metóda načítania časti škodlivého kódu, zvyčajne prostredníctvom počítačovej siete, do vzdialeného systému. **Staging** sa používa v spojení s exploitami, ktoré majú obmedzenú veľkosť, čo znamená, že je cieľom vykonať malú časť kódu, takzvaný *stager*, ktorý následne načíta väčšiu časť škodlivého kódu, pričom jednotlivé fázy načítavania sú niekedy číslované od čísla 0.



Obr. 3: Komponenty Command and Control komunikácie

4.2.1 C2 framework Sliver

Spoločnosť s názvom Bishop Fox, ktorá je zameraná na kybernetickú bezpečnosť, vytvorila v programovacom jazyku Go C2 framework s názvom **Sliver** [26], ktorý neustále získava čoraz väčšiu pozornosť škodlivých aktérov, pretože predstavuje vhodnú alternatívu s otvoreným zdrojovým kódom ku spomenutým populárnym C2 frameworkom [27].

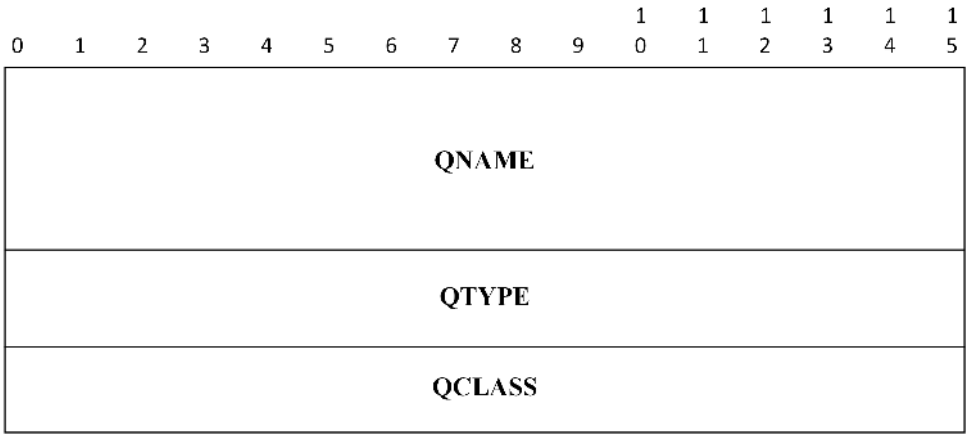
Architektúru C2 frameworku Sliver tvoria štyri hlavné komponenty [28]. **Serverová konzola** (z angl. Server Console) je hlavné rozhranie, ktoré sa aktivuje po spustení spustiteľného súboru *sliver-server*. Serverová konzola predstavuje nadmnožinu klientskej konzoly, pričom obe zdieľajú rovnaký kód až na príkazy špecifické pre server, ktoré sa týkajú správy klienta, teda operátora. Konzola servera komunikuje so serverom prostredníctvom rozhrania *gRPC*. **Sliver C2 server** je tiež súčasťou spustiteľného súboru *sliver-server*. Tento komponent spravuje internú databázu a taktiež spúšťa a zastavuje listenery. Hlavným rozhraním používaným na komunikáciu so serverom je už spomínané rozhranie *gRPC*, prostredníctvom ktorého sú implementované všetky funkcie. **Klientska konzola** predstavuje hlavné používateľské rozhranie, ktoré sa využíva na interakciu so Sliver C2 serverom. V neposlednom rade tvorí architektúru C2 frameworku Sliver komponent **implant**, ktorý reprezentuje škodlivý kód spustený na cieľovom systéme, ku ktorému chce škodlivý ak-

tér získať vzdialený prístup. Protokoly HTTP, HTTPS, DNS, Wireguard a mTLS sú podporované dostupným listenerom.

Zaujímavú techniku, ktorú ponúka Sliver, je takzvaná technika **DNS Tunneling**. Ide o metódu, ktorú škodliví aktéri používajú na ukrytie komunikácie s kompromitovanými systémami pomocou protokolu DNS. Súčasná implementácia tejto techniky vo frameworku Sliver nie je navrhnutá so zámerom úplného ukrytia prenášaných dát, čo znamená, že nie je vhodná na exfiltráciu údajov.

Uvedená technika funguje spôsobom, že sa do subdomény vložia požadované údaje a odošle sa dotaz na túto subdoménu autoritatívnemu mennému serveru, čo implantu umožní nadviazať spojenie so systémom kontrolovaným útočníkom. Toto spojenie je možné nadviazať aj v prípadoch, kedy kompromitovaný systém nemôže smerovať TCP, alebo UDP prevádzku do internetu. DNS dotazy sú vo všeobecnosti malé, čo znamená, že DNS komunikácia je pomalá. Táto skutočnosť má za následok, že ak je správa príliš veľká, na odoslanie údajov bude potrebných veľa dotazov, čo má negatívny dopad na rýchlosť komunikácie.

Na vytvorenie rýchleho DNS tunela je potrebné do každej požiadavky vložiť čo najviac údajov, aby sa znížil počet dotazov odoslaných pre danú správu. Najväčšou časťou v DNS požiadavke, ktorej štruktúru vidíme na obrázku 4, je pole **QNAME**, ktoré obsahuje doménu, na ktorú sa dopytujeme. DNS je protokol, ktorý nerozlišuje veľké a malé písmená, čo znamená, že napríklad písmená „b“ a „B“ sa považujú za rovnaké. Binárne údaje nie je možné posielať prostredníctvom DNS, preto musia byť zakódované pomocou vyhovujúceho kódovania. Na zakódovanie ľubovoľných binárnych údajov do ASCII sa zvyčajne používa *Base64* kódovanie. Toto kódovanie rozlišuje veľké a malé písmená, ale ako sme uviedli, protokol DNS nerozlišuje medzi veľkými a malými písmenami, čo môže spôsobiť problém pri dekódovaní dát, v prípade, že všetky znaky boli zmenené na malé písmená pri procese prekladu doménového mena na IP adresu. Z tohto dôvodu sa používa hexadecimálne kódovanie, ktoré nerozlišuje veľkosť písmen a používa len číslice 0 až 9 a písmená A až F. Táto metóda je veľmi neefektívna, čo má za následok zdvojnásobenie veľkosti kódovaných údajov. *Base32* kódovanie je efektívnejšia možnosť, ale najefektívnejšie kódovanie nám poskytuje kódovanie *Base58*, kde rozlišujeme veľké a malé písmená, rovnako ako pri Base64, ale používajú sa iba znaky, ktoré sú povolené v poli QNAME. Riešenie vo frameworku Sliver spočíva v tom, že sa najprv zistí, či sa Base58 dá použiť na spoľahlivé kódovanie údajov, a v prípade, že je detegovaný nejaký problém, program zvolí Base32 kódovanie.



Obr. 4: Štruktúra DNS požiadavky

Technika DNS Tunneling poskytuje škodlivým aktérom spôsob ako vytvoriť komunikačné kanále s kompromitovanými systémami. Napriek tomu, že prenášané údaje nie sú úplné ukryté, ide o užitočnú techniku v obmedzených prostrediach, kde môžu byť iné formy komunikácie blokované. Rýchlosť komunikácie je limitujúcim faktorom, pričom použitá metóda kódovania musí byť efektívna, aby sa znížil počet odoslaných dotazov.

4.2.2 C2 framework Merlin

Ďalším populárnym C2 frameworkom s otvoreným zdrojovým kódom je **Merlin** [29], ktorý bol taktiež napísaný v programovacom jazyku Go. Merlin funguje na základe *klient-server* architektúry a na komunikáciu medzi serverom a hostiteľskými agentmi je možné mimo bežné protokoly využiť aj novší protokol **HTTP/2** [30]. Použitím protokolu HTTP/2 počas komunikácie s kompromitovanými systémami Merlin dosahuje lepšie využitie sieťových zdrojov, pričom umožňuje viaceré súbežné výmeny dát v rámci jedného spojenia. Uvedený protokol HTTP/2 je *binárny* protokol, čo znamená, že je kompaktnejší a zároveň nie je čitateľný analytikom bez použitia interpretačného nástroja. Taktiež je potrebné uviesť, že tak ako voľne dostupné nástroje na detekciu škodlivej aktivity v počítačových sieťach aj niektoré komerčné riešenia nedokážu odhaliť škodlivú komunikáciu prostredníctvom protokolu HTTP/2.

Architektúru C2 frameworku Merlin tvoria dve hlavné časti, a to **server** a **agenti**. Vďaka programovaciemu jazyka Go je možné oba tieto komponenty skompilovať tak, aby fungovali na akejkoľvek platforme, čo poskytuje podporu operačných systémov Linux, Windows a macOS. Ďalšou výhodou je, že server skompilovaný na nasadenie na systéme Linux môže obsluhovať agentov skompilovaných pre všetky ostatné platformy.

Mimo iné, zaujímavá metóda, ktorú Merlin obsahuje, umožňuje agentovi vytvoriť TLS

klienta, ktorý sa vydáva za niekoho iného, len na základe poskytnutého **JA3 odtlačku** [31]. Na začatie TLS relácie, po nadviazaní TCP spojenia s cieľovým systémom, klient odošle *TLS Client Hello* paket. Tento paket a spôsob jeho generovania závisí od balíkov a metód použitých pri vytváraní klientskej aplikácie. Server, ak prijíma spojenia TLS, odpovie *TLS Server Hello* paketom, ktorý je formulovaný na základe knižníc a konfigurácií na strane servera, ako aj na základe podrobností v pakete TLS Client Hello. V TLS verzii 1.3 sú správy Client Hello a Server Hello, ktoré sa používajú na iniciovanie TLS spojenia a stanovenie parametrov šifrované, teda nie je možné ich ako v predošlých TLS verziách voľne čítať z odchytených paketov. Z verzií TLS, v ktorých vieme voľne prečítať hodnoty paketov Client Hello a Server Hello, dokážeme získať odtlačok konfigurácii, na základe ktorého následne vieme identifikovať klientske aplikácie.

Metóda **JA3** sa používa na zhromažďovanie desiatkových hodnôt bajtov vybraných polí *TLSVersion*, *Ciphers*, *Extensions*, *EllipticCurves*, *EllipticCurvePointFormats* v pakete **Client Hello**. Následne sa tieto hodnoty spoja v definovanom poradí, pričom na oddelenie každého poľa sa použije znak čiarky a na oddelenie hodnôt v každom poli znak pomlčky. Ak v pakete Client Hello nie sú žiadne rozšírenia protokolu TLS, príslušné polia zostanú prázdne. Zo spojeného reťazca sa v poslednom kroku vytvorí **MD5** hash odtlačok, s cieľom vytvoriť ľahko použiteľný a zdieľateľný tridsaťdva znakový odtlačok klientskeho systému. Ako príklad si môžeme ukázať konfiguráciu, ktorej extrahované hodnoty definovaných polí sú reprezentované nasledujúcim reťazcom.

```
769,47-53-5-10-49161-49162-49171-49172-50-56-19-4,0-10-11,23-24-25,0
```

Po aplikácii MD5 hash metódy dostávame odtlačok, resp. JA3 odtlačok daného klientskeho systému s hodnotou *„ada70206e40642a3e4461f35503241d5“*. Tento odtlačok môžeme následne použiť, tak ako na vytvorenie TLS Merlin klienta, aj na detekciu danej aplikácie, resp. jej konfigurácie.

Pre úplnosť predstavíme taktiež metódu **JA3S**, ktorá spočíva v zhromažďovaní desiatkových hodnôt bajtov vybraných polí *TLSVersion*, *Cipher*, *Extensions* z paketu **Server Hello**. Následne, tak ako pri metóde JA3, sa tieto hodnoty spoja v definovanom poradí, pričom na oddelenie každého poľa sa použije znak čiarky a na oddelenie hodnôt v každom poli znak pomlčky. Rovnako ako pri pakete Client Hello, ak v pakete Server Hello nie sú žiadne rozšírenia protokolu TLS, príslušné polia zostanú prázdne. V poslednom kroku vytvoríme zo spojeného reťazca **MD5** hash odtlačok, ktorý predstavuje JA3S odtlačok.

Funkcia MD5 bola zvolená z dôvodu spätnej kompatibility s už existujúcimi techno-

lógiami. Tak ako aj pri iných detekčných technikách aj pri procese detekcie škodlivých systémov pomocou odtlačkov JA3 a JA3S existuje riziko falošných pozitív. O JA3 otlačkoch môžeme uvažovať ako o ekvivalente reťazca HTTP hlavičky *User-Agent*. To, že jeden systém, resp. škodlivý systém, má určitý reťazec, neznamená, že bude vždy jedinečný pre daný softvér. Je možné, že rovnaký reťazec používa aj iný softvér. Neexistuje však dôvod, prečo nepoužívať tento reťazec na rozšírenie procesov analýzy a detekcie. Rovnako ako iné sieťové metadáta a artefakty, JA3 odtlačok je dodatočnou informáciou, ktorú môžeme použiť pri obohacovaní extrahovaných údajov. JA3S odtlačok, ak sa používa v spojení s odtlačkom JA3, môže výrazne znížiť úroveň falošne pozitívnych výsledkov v prípade, že vyhľadávame špecifické systémy.

4.2.3 C2 framework Covenant

V neposlednom rade si predstavíme ďalší populárny C2 framework s otvoreným zdrojovým kódom, ktorý sa nazýva **Covenant** [32]. Tento framework vznikol s cieľom spopularizovať plochu útočných vektorov a zároveň uľahčiť používanie útočných prostriedkov súboru technológií **.NET** [33]. C2 framework Covenant predstavuje multiplatformovú ASP.NET Core aplikáciu, ktorá poskytuje plnohodnotné webové rozhranie umožňujúce spoluprácu viacerých používateľov.

Tak ako predošlé predstavené C2 frameworky, aj Covenant funguje na základe *klient-server* architektúry, pričom jeho infraštruktúru tvoria tri hlavné komponenty **Covenant**, **Elite** a **Grunt**. Komponent architektúry s názvom Covenant predstavuje riadiaci server. Ďalšia zložka s názvom Elite je rozhranie príkazového riadka, ktoré operátori používajú na interakciu so serverom Covenant a na vykonávanie operácií. V neposlednom rade komponent Grunt slúži ako implant, ktorý je spúšťaný na kompromitovanom systéme. Všetky tri uvedené komponenty sú napísané v programovacom jazyku C#. Komponenty Covenant a Elite sú zamerané na ASP.NET Core a zároveň podporujú aj technológiu Docker, zatiaľ čo implant Grunt je zameraný na súbor technológií .NET.

C2 framework Covenant obsahuje niekoľko funkcionalít, ktorými sa pokúša odlíšiť od ostatných frameworkov. Jednou z hlavných funkcionalít je už spomínané plnohodnotné webové rozhranie, ktoré umožňuje spoločnú, alebo nezávislú prácu viacerých používateľov s jedným Covenant riadiacim serverom. ASP.NET Core je multiplatformový, čo poskytuje natívnu podporu na operačných systémoch Linux, Windows a macOS a taktiež ako bolo spomenuté je podporovaná aj technológia Docker. Ďalšou výhodou C2 frameworku Covenant je implementácia šifrovanej výmeny kryptografických kľúčov medzi implantmi Grunt a listenermi Covenant, ktorá je do veľkej miery založená na podobnej výmene, akú podporuje

spomínaný C2 framework Empire. Dynamická kompilácia pri každom generovaní nového implantu Grunt a listenera Covenant, alebo pri pridávaní nových úloh je taktiež podporovaná, pričom táto funkcionalita je aj rozšírená o proces obfuskácie kódu pomocou nástroja *ConfuserEx*. V neposlednom rade C2 framework Covenant poskytuje voľne dostupné API, pomocou ktorého je danú inštanciu možné ľahko rozširovať o nové funkcionality. Rozšírenie funkcionalít taktiež poskytuje možnosť meniť, alebo pridávať nové šablóny kódu pre implanty, ktoré definujú akým spôsobom bude komunikovať kompromitovaný systém s riadiacim serverom Covenant.

Možnosť pridávať vlastné šablóny, resp. rozšírenia, ponúka príležitosť vytvoriť modul, ktorý by generoval takzvané **DGA doménové mená** [34]. DGA sa používa na dynamické generovanie veľkého počtu pseudonáhodných doménových mien a následný výber malej podmnožiny domén pre C2 komunikačný kanál. Hlavnou myšlienkou dynamickej povahy DGA bolo vyhnúť sa pevne zakódovaným názvom doménových mien, alebo zoznamov IP adries v škodlivých binárnych súboroch, čo komplikuje extrakciu týchto informácií procesmi reverzného inžinierstva. DGA je zvyčajne tvorené tromi hlavnými zložkami, a to **časovým odtlačkom**, resp. časovým seedom, **generátorom doménového mena**, ktorý využíva časový seed a **súborom TLD domén**. Časový seed môže byť reprezentovaný napríklad ako aktuálny dátum, alebo ako aktuálny čas v preddefinovanom formáte. Generátor doménových mien predstavuje hlavnú časť DGA a jeho obsah môže byť napríklad náhodný reťazec znakov, spojenie náhodných slov, konštantná časť nasledovaná meniacou sa príponou a iné. V neposlednom rade súbor TLD domén musí obsahovať reálne hodnoty, ktoré definujú, pod ktorými entitami sú vygenerované domény registrované. Nasledujúca tabuľka uvádza niekoľko príkladov DGA doménových mien.

Doménové meno	**Malvérová rodina**
xxcnirvbqivbucfsbliu.com	Zeus
agabgtdhgsbspwsq.ru	Magecart
disaalallowdisallow.me	Dridex
xvrvdsuhphjg.online	Mirai
kffunqtohreoaejntwov.com	Zloader

Tabuľka 2: Príklady DGA doménových mien

4.2.4 Zhrnutie porovnania vybraných C2 frameworkov

V tejto časti sme si priblížili čo sú C2 frameworky a aké komponenty ich tvoria. Taktiež sme si uviedli tri príklady C2 frameworkov a ich vybrané funkcionality, ktorými sa odlišujú

od iných nástrojov. V tabuľkách nižšie uvádzame prehľad poskytovaných funkcionalít popísaných C2 frameworkov v porovnaní s tromi najpopulárnejšími C2 frameworkami, Cobalt Strike, Empire a Metasploit. Uvedené údaje sme korelovali s informáciami z dokumentácii jednotlivých C2 frameworkov, ako aj s projektom **C2 Matrix** [35], ktorý poskytuje kategorizované C2 frameworky a s nimi súvisiace informácie. V tabuľke 4, ktorá obsahuje prehľad dostupných protokolov, ktoré podporujú jednotlivé C2 frameworky, uvádzame iba ak daný protokol je podporovaný, teda táto skutočnosť je explicitne definovaná v dokumentácii, alebo priamo v programe.

C2 framework	**Licencia**	**Viacero používateľov**	**UI**	**API**
Cobalt Strike	Komerčný	Áno	GUI	Nie
Covenant	GPLv3	Áno	GUI	Áno
Empire	BSD-3	Nie	GUI, CLI	Áno
Merlin	GPLv3	Nie	CLI	Nie
Metasploit	BSD-3	Nie	CLI	Áno
Sliver	GPLv3	Áno	CLI	Nie

Tabuľka 3: Porovnanie vybraných C2 frameworkov

C2 framework	**TCP**	**HTTP**	**HTTPS**	**HTTP/2**	**HTTP/3**	**DNS**	**SMB**
Cobalt Strike	Áno	Áno	Áno			Áno	Áno
Covenant	Áno	Áno	Áno				Áno
Empire	Áno	Áno	Áno			Áno	Áno
Merlin		Áno	Áno	Áno	Áno		
Metasploit	Áno	Áno	Áno				Áno
Sliver	Áno	Áno	Áno			Áno	

Tabuľka 4: Podporované protokoly pre C2 komunikáciu vybraných C2 frameworkov

4.3 Teoretický prístup k detekcii C2 komunikácie

Pri odhaľovaní C2 komunikácie by sme sa mali v prvom rade zamerať na detekciu techník, ktoré sú využívané pri tejto komunikácii. V prípade, že sa výhradne nezameriavame na detekciu indikátorov konkrétnych C2 aplikácii, náš prístup nám poskytuje *flexibilnejšie*, *komplexnejšie* a *účinnejšie* odhaľovanie potenciálnych hrozieb. Pokročilí škodliví aktéri neustále menia svoje TTPs, s ktorými je aj spájaný výber aplikácie a

techník pre C2 komunikáciu. Z tohto dôvodu, zameriavanie sa na odhaľovanie konkrétnych aplikácii poskytujúcich nástroje pre C2 komunikáciu, nemusí byť z dlhodobého hľadiska účinné. Na druhej strane, menej pokročilí škodliví aktéri využívajú predvolené nastavenia predmetných nástrojov, čo nám umožňuje detegovať potenciálnu C2 komunikáciu s menším výpočtovým zaťažením. Analýza sieťovej prevádzky s cieľom detekcie techník C2 komunikácie môže odhaliť neznáme hrozby, ktoré výskumníci v oblasti kybernetickej bezpečnosti ešte neidentifikovali.

Popísané rámce kybernetickej bezpečnosti, **Cyber Kill Chain** a **MITRE ATT&CK**, predstavujú užitočné podporné nástroje pri odhaľovaní C2 komunikácie. Analýzou jednotlivých fáz kybernetického útoku môžeme identifikovať činnosti C2 a prijať požadované opatrenia na prevenciu, alebo zmiernenie dopadu daného útoku. Taktiež dokážeme zefektívniť identifikáciu C2 komunikácie, zameraním sa na TTPs, ktoré sú s ňou spájané.

V tejto časti práce čerpáme informácie z vybraných taktík, resp. techník matice MITRE ATTA&CK, konkrétne z príslušných častí *Mitigations* a *Detection* [18]. V praxi je taktiež možné využiť rámec **MITRE D3FEND** [36], ktorý je doplnkom k popísanému rámcu MITRE ATT&CK. Uvedený rámec MITRE D3FEND, resp. matica D3FEND (z angl. D3FEND Matrix) mapuje vzťahy medzi TTPs škodlivých aktérov a obrannými protiopatreniami.

C2 komunikácia môže byť detegovaná v každej fáze popísaného cyklu Cyber Kill Chain. V počiatočnej fáze **Reconnaissance** je možné použiť monitorovanie sieťovej prevádzky s cieľom detegovať neobvyklé, alebo podozrivé vzory a aktivity. Takáto aktivita môže predstavovať prichádzajúcu, alebo odchádzajúcu sieťovú komunikáciu ku škodlivým IP adresám, alebo škodlivým doménovým menám. Na odhalenie činnosti C2 vo fáze **Weaponisation** môžeme použiť nástroje, ktoré skenujú systémy pre výskyt podozrivého a škodlivého obsahu. V ďalšej fáze cyklu Cyber Kill Chain, ktorou je **Delivery**, mailové filtre na blokovanie škodlivých príloh a škodlivých externých odkazov dokážu detegovať indikátory, ktoré sú spájané s aktivitami C2. Vo štvrtej fáze cyklu, **Exploitation**, môžeme opäť použiť nástroje na monitorovanie sieťovej prevádzky. V nasledujúcej fáze, ktorou je **Installation**, dokážu detegovať C2 komunikáciu EDR nástroje, ktoré monitorujú správanie systémov a vyhľadávajú škodlivú aktivitu, akou sú neobvyklé systémové procesy, alebo neobvyklé sieťové spojenia. V predposlednej fáze cyklu, ktorá je samotne pomenovaná **Command and Control**, môžeme použiť na detekciu všetky doteraz uvedené techniky, teda monitorovanie a analýza sieťovej prevádzky, použitie skenerov na výskyt škodlivých súborov, využitie EDR nástroja s cieľom odhaliť komunikáciu so známym C2 serverom. V neposlednom rade, vo fáze **Actions on Objectives**, je opäť možné použiť všetky

spomenuté techniky a nástroje s rozdielom, že v tejto fáze sa zameriame na neobvyklé, alebo neoprávnené prenosy súborov, zmeny konfigurácie systémov, alebo na inú aktivitu, ktorá naznačuje škodlivú činnosť.

V našej práci sa zameriavame výhradne na detekciu C2 komunikácie v *počítačových sieťach*. Pomocou matice MITRE ATT&CK vieme definovať konkrétne techniky pre taktiku s názvom **Command and Control (TA0011)** [4], na ktoré sa zameriavame pri detekcii C2 komunikácie. Nasledujúca tabuľka poskytuje prehľad vybraných techník, resp. podtechník matice MITRE ATT&CK, ktorým sa venujeme v tejto práci.

Technika	**Podtechnika**
Application Layer Protocol (T1071)	Web Protocols (T1071.001)
Application Layer Protocol (T1071)	Domain Name System (T1071.004)
Data Encoding (T1132)	Standard Encoding (T1132.001)
Dynamic Resolution (T1568)	Domain Generation Algorithms (T1568.002)
Proxy (T1090)	Multi-hop Proxy (T1090.003)

Tabuľka 5: Vybrané techniky taktiky Command and Control (TA0011)

Ďalej rozširujeme uvedené techniky, resp. podtechniky o detekčné procesy, ktoré sme vytvorili na základe analýzy C2 frameworkov a dodatočných otvorených zdrojov. Jednotlivé detekčné techniky, ako aj ich implementáciu a testovanie si priblížime bližšie v ďalších častiach tejto práce.

5 Analýza riešení a definícia požiadaviek

Pri výbere vhodného nástroja by sme mali porovnávať dostupné možnosti na základe ich poskytovaných funkcií, možnosti rozširovania ich schopností (napríklad pomocou rozšírení), podpory komunity, transparentnosti, ale aj celkovej účinnosti. Napriek tomu, že by sme nemali porovnávať nástroje iba na základe dostupnosti zdrojového kódu, alebo na základe toho, či sa jedná, alebo nejedná o platenú službu, pri analýze dostupných riešení sme sa výhradne zamerali na bezplatné riešenia s otvoreným zdrojovým kódom. Dôvodom je, že náš nástroj je alternatívou k malému množstvu bezplatných nástrojov, ktoré majú otvorený zdrojový kód a ich poskytované funkcionality vieme využiť pri detekcii C2 komunikácie.

Skutočnosť, že daný program má otvorený zdrojový kód vnímame ako pozitívny faktor, pokiaľ máme znalosť programovacieho jazyka, v ktorom je takýto program vyvinutý a našim cieľom je dostupný program modifikovať, prípadne doplniť o nové funkcionality.

5.1 Analýza dostupných nástrojov

Pred definovaním konkrétnych používateľských funkcionalít nášho programu si porovnáme voľne dostupné nástroje s otvoreným zdrojovým kódom, ktorých vývoj je stále udržiavaný a ktoré je možné použiť pri detekcii C2 komunikácie.

Arkime [37] je komplexný analytický nástroj, ktorý bol navrhnutý tak, aby pomáhal bezpečnostným tímom monitorovať a analyzovať sieťovú prevádzku. Tento nástroj poskytuje viacero funkcií vrátane odchytávania paketov v reálnom čase, spätného zostavovania relácií a pokročilých možností vyhľadávania, ako aj podporu pre rozšírenia, ktoré dopĺňajú základné poskytované funkcionality. Arkime je navrhnutý tak, aby sa dal nasadiť na rozličných systémoch, pričom dokáže spracovať desiatky gigabitov sieťovej prevádzky za sekundu. Odchytené pakety sú ukladané na dostupnom mieste na pamäťovom disku daného senzora. Systém Arkime sa skladá z komponentov **capture**, **viewer** a **elasticsearch**. Prvý uvedený komponent capture, predstavuje aplikáciu napísanú v programovacom jazyku C, ktorá poskytuje odchytávanie sieťovej prevádzky, zapisovanie paketov na pamäťový disk a odosielanie údajov z rozhrania do komponentu elasticsearch, ktorý slúži ako vyhľadávacia databázová technológia. Komponent viewer je Node.js aplikácia, ktorá spravuje webové rozhranie nástroja Arkime. Po inštalácii používateľ pristupuje k dátam, ktoré tvorí odchytená sieťová prevádzka, pomocou spomenutého webového rozhrania.

Nástroj Arkime poskytuje rôzne spôsoby interpretácie odchytených dát. Primárnym

zobrazením je stránka obsahujúca kompletný zoznam relácií, pričom pre jednotlivé relácie je možné zobraziť ich metadáta a údaje príslušných paketov. Ďalším spôsobom zobrazenia odchytených dát, je *SPI prehľadová stránka*, ktorá umožňuje používateľovi zobraziť všetky jedinečné hodnoty pre každé pole spracovaných dát.

Zeek [38] je pasívny analytický nástroj pre sieťovú prevádzku, vyvinutý so zámerom podpory vyšetrovania podozrivých, alebo škodlivých aktivít v počítačových sieťach. Tento nástroj taktiež poskytuje funkcionality mimo oblasť kybernetickej bezpečnosti, ako napríklad meranie výkonu a riešenie technických problémov. Zeek sa špecificky zameriava na vysokorýchlostné a veľkoobjemové monitorovanie sietí. Okrem bežného spracovania odchytených paketov tento nástroj poskytuje extrakciu údajov z aplikačnej vrstvy. Tieto údaje môžu obsahovať dáta HTTP relácií, extrahované súbory, detekciu zraniteľných verzií pozorovaných aplikácii v monitorovanej počítačovej sieti, detekciu útoku hrubou silou, napríklad na protokol SSH, overovanie hodnôt v TLS certifikátoch a ďalšie.

Nástroj Zeek predstavuje alternatívu k finančne nákladným riešeniam a k analytickým programom, ktoré sú zvyčajne limitované na pevne definované detekčné pravidlá. Napriek tomu, že tento nástroj podporuje aj detekciu založenú na signatúrach, naviac poskytuje aj skriptovací jazyk, ktorý rozširuje možnosti vyhľadávania škodlivých aktivít. Škálovateľnosť systému Zeek zabezpečuje *klastrová architektúra*, ktorá umožňuje dynamicky pridávať, alebo odoberať potrebné výpočtové zdroje.

RITA [39] je nástroj na analýzu sieťovej prevádzky zameraný na detekciu indikátorov C2 komunikácie a na odhaľovanie iných škodlivých aktivít v počítačovej sieti. Detekcia uvedených aktivít nie je založená primárne na základe signatúr, ale prostredníctvom *štatistickej analýzy* odchytených sieťových dát. Nástroj RITA na vstupe spracováva výhradne výstupy z popísaného systému Zeek a na výstupe je možné vygenerovať správu obsahujúcu zistenia z vykonanej analýzy vo formáte HTML. Skutočnosť, že používanie programu RITA je podmienené používaním systému Zeek pokladáme za negatívnu vlastnosť.

Medzi primárne funkcie uvedeného nástroja patrí detekcia indikátorov C2 komunikácie, detekcia techniky DNS Tunneling, identifikácia dlhých spojení, extrakcia HTTP hlavičiek User-Agent, analýza dĺžky URL adries, detekcia príznakov skenovania portov a v neposlednom rade vyhľadávanie známych škodlivých domén z voľne dostupných zoznamov, ktoré iniciovali, alebo prijali spojenia.

Suricata [40] je nástroj na detekciu kybernetických hrozieb, ktorý vyvinula spolu s komunitou a uverejnila v roku 2010 nezisková organizácia *OISF*. Tento nástroj pri

monitorovaní sieťovej prevádzky deteguje škodlivé aktivity v počítačovej sieti na základe rozsiahlej sady pravidiel. Jednou z alternatív popisovaného nástroja je aplikácia **Snort** [41]. Hlavnou výhodou Suricata je, že pri analýze sieťovej prevádzky dokáže spracovávať niekoľko udalostí súčasne prostredníctvom viacerých vlákien systému, čo znamená, že nemusí prerušovať vykonávanie iných požiadaviek. Výpočtová zložka tohto nástroja je navrhnutá tak, aby využívala najnovšie viacjadrové čipové sady CPU ako aj hardvérovú akceleráciu na zvýšenie efektivity. Suricata používa štandardné vstupné a výstupné súborové formáty ako YAML a JSON, ktoré umožňujú jednoduchú integráciu s ďalšími nástrojmi akými sú Elasticsearch, Kibana a Splunk. Ďalšou výhodou Suricata je, že má rozsiahlu komunitu vývojárov, ktorá vytvorila rôzne zdroje vrátane inštalačných a používateľských príručiek, často kladených otázok a popisu riešení rozličných problémov. Po základnej inštalácii aplikácie je potrebné vytvoriť konfiguračné súbory a stiahnuť a nastaviť detekčné pravidlá.

Popisovaný nástroj je možné použiť ako pasívne **IDS**, kedy Suricata monitoruje sieťovú prevádzku a v prípade, že deteguje podozrivú, alebo škodlivú aktivitu, upozorní používateľa. Druhou možnou konfiguráciou tohto nástroja je nastavenie ako aktívne **IPS**. V tomto prípade Suricata nielen monitoruje prichádzajúcu a odchádzajúcu sieťovú prevádzku, ale aj blokuje detegovanú škodlivú aktivitu. Správna implementácia IPS je náročná z dôvodu, že ak nie sú vhodne nakonfigurované detekčné pravidlá, resp. signatúry, takýto systém blokuje aj legitímnu sieťovú prevádzku, ktorá sa môže javiť ako podozrivá.

Napriek tomu, že popísané nástroje poskytujú množstvo funkcionalít detekcie rozličných škodlivých aktivít, nemali by sme tieto nástroje vnímať ako jediný prostriedok obrany voči škodlivým aktérom, ale skôr ako jednu vrstvu v komplexnej kybernetickej bezpečnostnej stratégii.

5.2 Používateľské požiadavky

Na základe analýzy voľne dostupných programov s otvoreným zdrojovým kódom, definujeme náš nástroj ako ich alternatívu. Našim cieľom je, aby aplikácia bola jednoduchá na používanie, zaručujúca efektivitu a spoľahlivosť pri práci a s možnosťou aktualizácie existujúcich detekčných techník a rozširovania nových detekčných funkcionalít v budúcnosti. Ďalej si v tejto časti presne definujeme funkcionálne a nefunkcionálne používateľské požiadavky.

5.2.1 Funkcionálne požiadavky

- Softvér musí používať rozhranie príkazového riadku, pričom používateľ definuje úlohy prostredníctvom dostupných argumentov.
- Softvér musí detailne informovať používateľa o jednotlivých vykonávaných krokoch.
- Softvér musí byť konfigurovateľný pomocou jedného samostatného konfiguračného súboru.
- Softvér musí vedieť detegovať svoje chýbajúce komponenty a nesprávne, alebo neúplné údaje v konfiguračnom súbore.
- Softvér musí vedieť spracovať odchytenú sieťovú prevádzku z PCAP súborov.
- Softvér musí taktiež poskytovať funkcionalitu odchytávania sieťovej prevádzky, ktorej parametre sa nastavia prostredníctvom konfiguračného súboru aplikácie.
- Zo spracovanej odchytenej sieťovej prevádzky musí softvér mimo iné vedieť extrahovať
  - čas začatia a ukončenia odchytávania sieťovej prevádzky,
  - jedinečné externé TCP sieťové spojenia,
  - verejné zdrojové a cieľové IP adresy,
  - doménové mená,
  - obsah HTTP relácii vrátane HTTP hlavičiek,
  - dopytované URL adresy,
  - vybrané polia z TLS certifikátov.
- Extrahované dáta zo vstupného PCAP súboru musí ďalej softvér vedieť exportovať do súboru formátu JSON.
- Softvér musí umožňovať zobraziť v rozhraní príkazového riadku rôzne informačné a štatistické údaje ako
  - SHA-256 hash odtlačok analyzovaného PCAP súboru,
  - začiatok a koniec odchytávania sieťovej prevádzky,
  - počet externých TCP sieťových spojení,
  - počet jedinečných extrahovaných doménových mien,
  - počet jedinečných verejných IP adries,

  - prehľad najčastejšie sa vyskytujúcich zdrojových a cieľových verejných IP adries,
  - počet HTTP relácií,
  - počet jedinečných extrahovaných URL adries,
  - počet spracovaných paketov, ktoré obsahujú údaje z TLS certifikátov.
- Softvér musí vedieť detegovať
  - techniku DNS Tunneling,
  - DGA doménové mená,
  - známe škodlivé JA3 odtlačky,
  - sieťové spojenia s nadmernou frekvenciou,
  - dlhé sieťové spojenia,
  - neobvykle veľké HTML odpovede,
  - známe hodnoty TLS certifikátov vybraných C2 frameworkov,
  - Tor komunikáciu a zároveň aj konkrétne sieťovú komunikáciu k výstupným uzlom (z angl. exit nodes) Tor siete,
  - dopytované doménové mená, ktoré sú spájané s ťažbou kryptomien.
- Softvér musí vedieť detegovať pomocou integrácie rozšírenia
  - známe IP adresy C2 riadiacich serverov, ktoré prijali, alebo iniciovali sieťové spojenie,
  - známe doménové mená C2 riadiacich serverov, ktoré prijali, alebo iniciovali sieťové spojenie,
  - známe škodlivé URL adresy, ktoré sú spájané s C2 komunikáciou,
  - sieťovú komunikáciu s potenciálnymi C2 riadiacimi servermi.
- Detegované potenciálne indikátory C2 komunikácie musí softvér vedieť exportovať do súboru formátu JSON.
- Po vykonaní všetkých požadovaných úloh musí program vygenerovať správu o analýze vo formáte HTML a PDF, ktorá obsahuje detailný prehľad detegovaných IoCs, ako aj celkové vyhodnotenie detekcie indikátorov C2 komunikácie.

5.2.2 Nefunkcionálne požiadavky

- Softvér musí byť efektívny a jednoduchý na používanie.
- Softvér musí byť navrhnutý tak, aby bolo zabezpečené jednoduché pridávanie nových detekčných funkcionalít v budúcnosti a zároveň musí byť zabezpečená jednoduchá aktualizácia existujúcich detekčných techník.
- Softvér musí byť prenositeľný, bez zložitej konfigurácie a s minimálnymi požiadavkami.
- Softvér musí byť spoľahlivý a pri vzniknutej chybe vykonávania, relevantne informovať používateľa o danom probléme.
- Zdrojový kód softvéru musí byť otvorený a voľne dostupný.

6 Návrhová a systémová špecifikácia aplikácie C2Detective

Vychádzajúc z analytickej časti a definovaných funkcionálnych a nefunkcionálnych požiadaviek, si v tejto časti práce predstavíme hlavné komponenty aplikácie **C2Detective** a proces od spracovania vstupného súboru až po proces zapísania výsledkov analýzy do výstupných súborov. V čase písania tejto práce naša aplikácia vyžaduje operačný systém Linux, primárne z dôvodu zabezpečenia podpory funkcionality odchytávania sieťovej prevádzky.

Pre vývoj aplikácie C2Detective bol zvolený programovací jazyk **Python** (verzia 3.10) z dôvodu, že tento programovací jazyk poskytuje množstvo modulov, ktoré sú voľne dostupné prostredníctvom repozitára s názvom **PyPI** [42] a ktoré je možné použiť pre naše analytické potreby. Pre správu projektu sme využili platformu **GitHub**, na ktorej je možné voľne prehliadať zdrojový kód našej aplikácie [43].

Nasledujúce zobrazenie 5 definuje požadovanú hierarchickú štruktúru priečinkov a súborov našej aplikácie. Jednotlivé komponenty a fázy, v ktorých sú dané časti aplikácie využívané si ďalej v tejto časti bližšie priblížime.

```
C2Detective
├── c2detective.py
├── config
│   ├── config.yml
│   └── domain_whitelist.txt
├── iocs
│   ├── c2_tls_certificate_values.json
│   ├── crypto_domains
│   │   ├── crypto_domains.json
│   │   └── update_crypto_domains.py
│   ├── ja3
│   │   ├── ja3_rules.json
│   │   └── update_ja3_rules.py
│   └── tor
│       ├── tor_nodes.json
│       └── update_tor_nodes.py
├── logs
├── src
│   ├── analyst_profile.py
│   ├── detection_engine.py
│   ├── detection_reporter.py
│   ├── enrichment_engine.py
│   ├── packet_capture.py
│   └── packet_parser.py
└── templates
    └── report_template.html
```

Obr. 5: Hierarchická štruktúra projektu C2Detective

6.1 Predstavenie komponentov aplikácie

Procesy, resp. komponenty našej aplikácie sme rozdelili do ôsmych častí, ktoré uvádzame nižšie, pričom fázy odchytávanie sieťovej prevádzky a obohacovanie detegovaných indikátorov kompromitácie sú voliteľné na základe potrieb používateľa.

Konfigurácia aplikácie

Bezprostredne po spustení aplikácie C2Detective prebieha konfigurácia, pri ktorej sa načítavajú údaje z poskytnutého konfiguračného súboru formátu YAML. Tieto údaje sú obsiahnuté v samostatnej triede s názvom **AnalystProfile** (*analyst_profile.py*) a ďalej sú využívané v ďalších komponentoch prostredníctvom inštancie tejto triedy, resp. jej atribútov. Konfiguračný súbor aplikácie C2Detective obsahuje nasledovné sekcie

- ***api_keys***: API kľuče služieb, ktoré sú využívané pri obohacovaní IoCs,
- ***api_urls***: URL adresy API rozhraní služieb, ktoré sú využívané pri obohacovaní IoCs,
- ***settings***: všeobecné nastavania aplikácie,
- ***feeds***: URL adresy externých zdrojov, ktorých dáta sú spracovávané pre detekčné účely,
- ***file_paths***: relatívne systémové cesty vyžadovaných súborov,
- ***enrichment_services***: povoľovanie, alebo zakazovanie jednotlivých služieb, ktoré sú využívané pri obohacovaní IoCs,
- ***sniffing***: nastavenia odchytávania sieťovej prevádzky,
- ***thresholds***: prahové hodnoty využívané pri detekcii indikátorov C2 komunikácie,
- ***plugins***: konfigurácia rozšírení aplikácie.

Overenie integrity aplikácie

Fáza overovania integrity aplikácie zisťuje, či je dodržaná požadovaná hierarchická štruktúra priečinkov a súborov našej aplikácie. V prípade, že niektorý z požadovaných priečinkov, alebo súborov sa nenachádza v projekte, je o tejto skutočnosti informovaný koncový používateľ správou v rozhraní príkazového riadku. Zobrazená správa obsahuje predmetný priečinok, resp. súbor a relevantnú referenciu na jeho vytvorenie, resp. stiahnutie z verejného repozitára našej aplikácie.

Aktualizácia dát z externých zdrojov

Spracovávané dáta z externých zdrojov je potrebné priebežne aktualizovať. V tejto fáze aplikácia overuje, kedy boli naposledy upravené predmetné spracované súbory a následne porovnáva tieto časy s aktuálnym časom. Zoznam IP adries siete Tor je odporúčané aktualizovať každých 30 minút, zoznam JA3 pravidiel a zoznam doménových mien, ktoré sú spájané s ťažbou kryptomien je odporúčané aktualizovať každých 24 hodín. Ak je rozdiel času medzi aktuálnym časom a časom poslednej úpravy daných súborov väčší ako uvedené odporúčané časové intervaly, aplikácia informuje o tejto skutočnosti používateľa. Pri opätovnom spustení aplikácie má používateľ možnosť aktualizovať jednotlivé súbory pomocou definovania príslušného argumentu, resp. argumentov.

Odchytávanie sieťovej prevádzky

Odchytávanie sieťovej prevádzky predstavuje voliteľnú funkcionalitu aplikácie C2Detective, ktorá je reprezentovaná triedou **PacketCapture** (*packet_capture.py*). V prípade voľby tejto funkcionality je na základe definovanej konfigurácie zo sekcie ***sniffing*** odchytávaná sieťová prevádzka, ktorej dáta sú zapísané do výstupného súboru formátu PCAP. Tento súbor je následne ďalej spracovaný nasledovaným komponentom našej aplikácie.

Spracovanie odchytenej sieťovej prevádzky

Jedným z hlavných komponentov aplikácie je spracovanie odchytenej sieťovej prevádzky, ktorej atribúty a metódy sú obsiahnuté v triede s názvom **PacketParser** (*packet_parser.py*). Prvou úlohou tohto komponentu je načítanie paketov zo vstupného PCAP súboru pomocou modulu **scapy** [44] a jeho funkcie s názvom *rdpcap*. Táto funkcia nám vracia zoznam objektov, ktoré reprezentujú jednotlivé pakety. Taktiež pre ďalšie analytické procesy využívame aj ďalšiu funkciu modulu scapy s názvom *sessions*. Táto funkcia vytvára a vracia slovník, ktorý obsahuje usporiadané pakety podľa poradia ich prenosu zo všetkých sieťových relácií, ktoré sú identifikované v danom zozname paketov.

Najdôležitejšou funkciou tohto komponentu je extrakcia vybraných dát z načítaných paketov. V tejto funkcii prehľadávame úplný zoznam paketov, ktorý sme získali pomocou spomínanej funkcie rdpcap a extrahujeme, resp. spracovávame nasledovné údaje

- ***start_time***: časový odtlačok začiatku odchytávania sieťovej prevádzky,
- ***end_time***: časový odtlačok konca odchytávania sieťovej prevádzky,
- ***connection_frequency***: sieťové spojenia a ich príslušné frekvencie,

- ***public__src__ip__list***: zoznam verejných zdrojových IP adries,
- ***public__dst__ip__list***: zoznam verejných cieľových IP adries,
- ***public__ip__list***: zoznam všetkých verejných IP adries,
- ***external__tcp__connections***: zoznam TCP sieťových spojení,
- ***dns__packets***: zoznam DNS paketov,
- ***domain__names***: zoznam jedinečných doménových mien,
- ***http__payloads***: zoznam objektov dát, ktoré sú prenášané prostredníctvom HTTP protokolu,
- ***http__sessions***: zoznam HTTP spojení, ktoré reprezentujú vybrané hodnoty,
- ***unique__urls***: zoznam jedinečných URL adries.

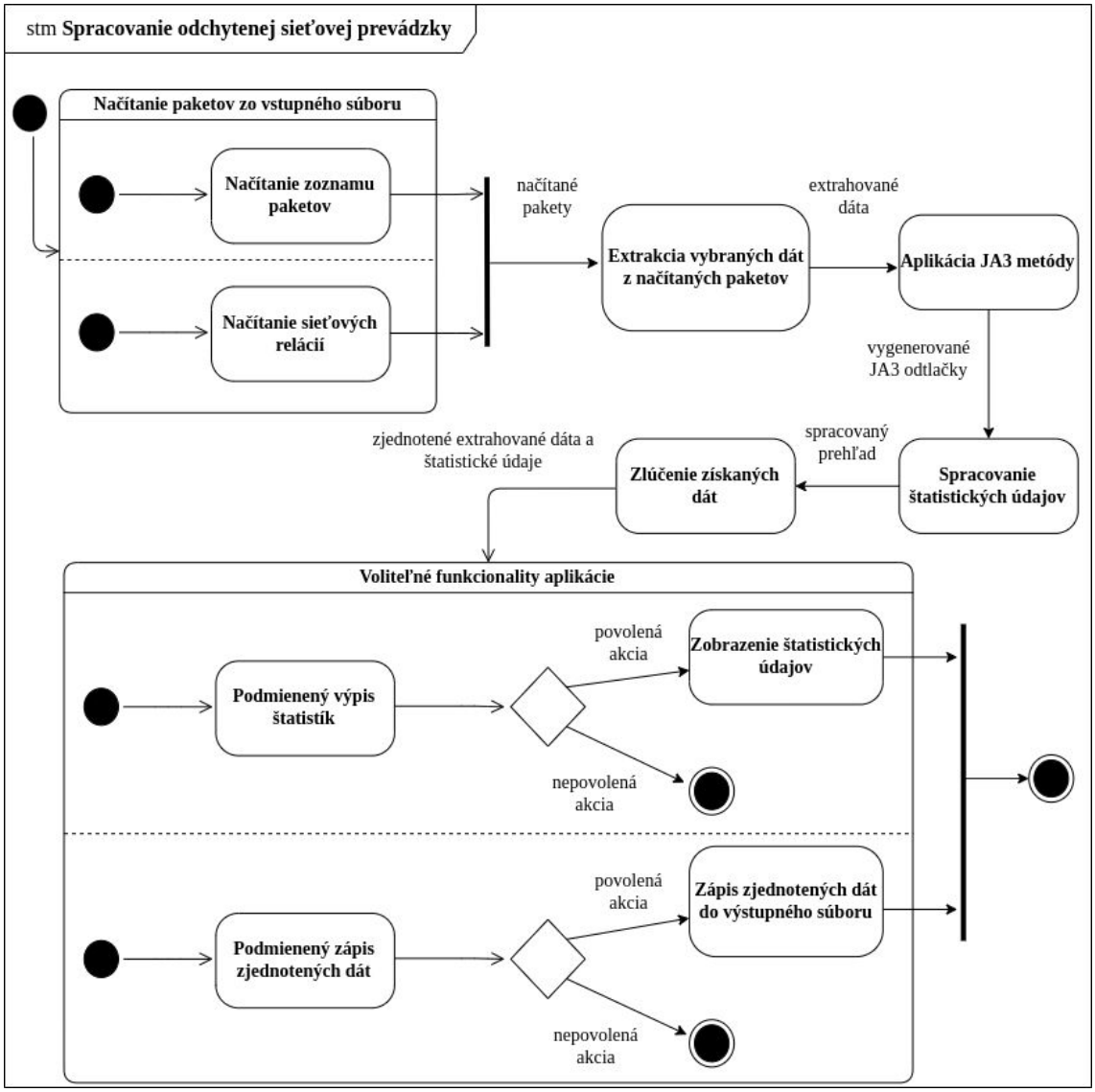
Na extrakciu vybraných polí z TLS certifikátov používame voľne dostupný nástroj **Tshark** [45], ktorý je vo všeobecnosti možné použiť na odchytávanie a analýzu paketov [46, 47]. Pomocou prepínača *-Y* a hodnoty *„tls.handshake.certificate“* si dokážeme odfiltrovať pakety, ktoré súvisia s výmenou TLS certifikátov. Spomenutý nástroj a prepínač spúšťame prostredníctom modulu **subprocess**, ktorým vytvoríme nový proces a výstup z neho, resp. z programu Tshark si presmerovávame do lokálnej premennej. Tento výstup ďalej spracovávame a z odfiltrovaných dát extrahujeme údaje z vybraných polí TLS certifikátov.

V rámci tohto komponentu taktiež používame nástroj **JA3** [48] na vytvorenie všetkých možných JA3 odtlačkov zo vstupného PCAP súboru, pričom tento nástroj spúšťame rovnakým spôsobom ako spomenutý nástroj Tshark. V tomto prípade sú už vygenerované JA3 odtlačky vo vyhovujúcej štruktúre.

V prípade, že používateľ zvolil dostupnú funkciu zobrazenia štatistických údajov o spracovaných dátach do rozhrania príkazového riadku sú zobrazené nasledovné informácie

- SHA-256 odtlačok vstupného PCAP súboru,
- časový odtlačok začiatku a konca odchytávania sieťovej prevádzky,
- počet extrahovaných TCP sieťových spojení,
- počet jedinečných extrahovaných doménových mien,
- počet extrahovaných verejných IP adries,

- prehľad najčastejšie sa vyskytujúcich verejných zdrojových a cieľových IP adries a ich príslušné počty,
- počet HTTP spojení,
- počet jedinečných extrahovaných URL adries,
- počet TLS certifikátov, z ktorých boli extrahované údaje.



Obr. 6: Stavy komponentov spracovania odchytenej sieťovej prevádzky

V neposlednom rade sú všetky uvedené extrahované dáta skombinované do jedného objektu, pričom jednotlivé údaje zostávajú vo svojich príslušných štruktúrach a takto skombinované dáta sú taktiež zapísané do výstupného súboru formátu JSON.

Diagram, ktorý môžeme vidieť na obrázku 6 nám poskytuje prehľad popísaného komponentu spracovania odchytenej sieťovej prevádzky a jednotlivých funkcií, ktoré ho tvoria.

Detekcia C2 komunikácie

Detekčné techniky v triede **DetectionEngine** (*detection_engine.py*) pracujú nad extrahovanými dátami z poskytnutého PCAP súboru, čím sa predchádza opakovanému prehľadávaniu všetkých odchytených paketov. Na začiatku detekčnej fázy sa vytvára prázdny objekt, reprezentujúci indikátory C2 komunikácie. Tento objekt je postupne počas jednotlivých techník, v prípade detekcie škodlivej aktivity rozširovaný o údaje, ktoré predstavujú *potenciálne indikátory C2 komunikácie*. Z údajov, ktoré tvoria detegované indikátory C2 komunikácie sú ďalej extrahované verejné IP adresy, doménové mená a URL adresy, ktoré ma následne používateľ možnosť obohatiť o ďalšie informácie, prostredníctvom rôznych implementovaných služieb.

Výsledky o detegovaní, resp. nedetegovaní potenciálnych indikátorov C2 komunikácie sú priebežne počas vykonávania tejto fázy zobrazované v rozhraní príkazového riadku, kde používateľ spustil našu aplikáciu. Metódy detekcie potenciálnych indikátorov C2 komunikácie triedy DetectionEngine je možné doplniť o ďalšie funkcie prostredníctvom údajov, ktoré sú obsiahnuté v predspracovanej databáze rozšírenia **C2Hunter**.

Integráciu spomenutého rozšírenia do našej aplikácie a jednotlivé implementované funkcionality, ktoré tvoria komponent detekcia C2 komunikácie, si bližšie popíšeme v ďalej časti tejto práce.

Obohacovanie detegovaných indikátorov kompromitácie

Ako bolo uvedené pri predošlom komponente, v prípade detekcie jednotlivých indikátorov C2 komunikácie a ich spracovania, sú dodatočne extrahované do samostatných objektov verejné IP adresy, doménové mená a URL adresy. Tieto údaje má používateľ možnosť obohatiť o ďalšie informácie prostredníctvom API rozhrania služieb **AbuseIPDB** [49], **AlienVault** [50], **Shodan** [51], **ThreatFox** [52], **URLhaus** [53] a **VirusTotal** [54]. Na tomto mieste je potrebné uviesť, že v prípade využitia služieb Shodan a VirusTotal je odporúčané použiť *platený* API prístup z dôvodu, že neplatené verzie sú vo viacerých smeroch limitované. Koncový používateľ má možnosť voľby, ktoré služby budú použité na obohacovanie detegovaných potenciálnych indikátorov C2 komunikácie.

Obohacovanie detegovaných indikátorov C2 komunikácie je obsiahnuté v triede **EnrichmentEngine** (*enrichment_engine.py*). Na základe zoznamov údajov, teda či sa jedná o zoznam IP adries, doménových mien, alebo URL adries, naša aplikácia vie, ktoré služby

je možné použiť pre ich obohacovanie. Dopytovania API rozhraní uvedených služieb boli implementované na základe ich príslušných dokumentácii. Obohatené extrahované indikátory C2 komunikácie sú zapísané do výstupného súboru formátu JSON. Tento súbor je tvorený tromi hlavnými kľúčmi, ktoré obsahujú príslušné obohatené údaje, teda IP adresy, doménové mená a URL adresy. Pre prehľad štruktúry tohto súboru uvádzame nižšie jeho príklad, ktorý by v praxi obsahoval obohatené údaje.


```
{
    "ip_addresses": {
        "78.128.112.139": {
            "abuseipdb": {},
            "threatfox": {},
            "virustotal": {},
            "shodan": {},
            "alienvault": {},
            "urlhaus": {}
        }
    },
    "domain_names": {
        "considerf.info": {
            "abuseipdb": {},
            "threatfox": {},
            "virustotal": {},
            "shodan": {},
            "alienvault": {},
            "urlhaus": {}
        }
    },
    "urls": {
        "https://construtic.com.br/eiml/": {
            "abuseipdb": {},
            "threatfox": {},
            "virustotal": {},
            "shodan": {},
            "alienvault": {},
            "urlhaus": {}
```

```
        }
    }
}
```

Reportovanie výstupov analýzy

Posledný komponent našej aplikácie, ktorým sú metódy reportovania výstupov analýzy, je reprezentovaný triedou **DetectionReporter** (*detection_reporter.py*). Na záver behu aplikácie sú detegované potenciálne indikátory C2 komunikácie zapísané do výstupného súboru JSON. Správa, ktorá obsahuje výsledky analýzy je vytvorená pomocou modulu **jinja2**, ktorý nám umožňuje vytvárať dynamické webové stránky alebo dokumenty. Predpripravená šablóna výstupnej správy sa nachádza v súbore *report_template.html*, v priečinku templates. Vygenerovanú správu o výsledkoch analýzy formátu HTML využijeme taktiež na vygenerovanie tejto správy vo formáte PDF pomocou modulu **pdfkit**. Uvedené výstupné správy o výsledkoch analýzy poskytujú detailný prehľad detegovaných indikátorov C2 komunikácie pre koncového používateľa, ktorý ma možnosť podľa potreby zdieľať tieto správy s ďalšími ľuďmi.

Obohatené indikátory kompromitácie nie sú obsiahnuté v správe o výsledkoch analýzy formátu HTML ani PDF z dôvodu, že ich rozsah môže byť príliš veľký, čo by mohlo mať v konečnom dôsledku negatívny dopad na prehľadnosť výstupnej správy. Zobrazenie obohatených údajov, prípadne ich ďalšia analýza je vhodná napríklad prostredníctvom nástroja na vizualizáciu a analýzu dát s názvom **Kibana** [55].

7 Implementované detekčné metódy

Pri výbere techník, pomocou ktorých sa snažíme detegovať indikátory C2 komunikácie prostredníctvom našej aplikácie **C2Detective**, sme sa snažili nájsť rovnováhu medzi možnosťou výberu funkcionalít a dodatočnej konfigurácie detekčných techník koncovým používateľom. Aplikácia C2Detective taktiež poskytuje modularitu detekčných schopností pomocou rozšírení. V čase písania tejto práce, naša aplikácia umožňuje integráciu s rozšírením **C2Hunter**, ktorý sme ako samostatný nástroj pôvodne vyvíjali nezávisle od aplikácie C2Detective.

V nasledujúcom výpise uvádzame prehľad *prahových hodnôt*, ktoré sú používateľom definované v konfiguračnom súbore aplikácie C2Detective a ktoré sú následne využívané v detekčnej fáze tohto programu.

- **MAX_FREQUENCY**: Podiel TCP sieťových spojení zo všetkých TCP spojení v percentách.
- **MAX_DURATION**: Dĺžka sieťové spojenia v sekundách.
- **MAX_HTTP_SIZE**: Veľkosť tela HTTP správy v bajtoch, ktorá je odoslaná príjemcovi.
- **MAX_SUBDOMAIN_LENGTH**: Počet znakov subdomény.

Ďalej si v tejto časti priblížime jednotlivé implementované funkcionality, ktoré vyhľadávajú vybrané indikátory C2 komunikácie. Celkovo dokážeme pomocou aplikácie C2Detective detegovať štrnásť potenciálnych indikátorov C2 komunikácie a to konkrétne

- DGA doménové mená,
- techniku DNS Tunneling,
- známe škodlivé JA3 odtlačky,
- sieťové spojenia s nadmernou frekvenciou,
- dlhé sieťové spojenia,
- neobvykle veľké HTTP odpovede,
- Tor sieťovú prevádzu,
- sieťovú prevázku ku výstupným uzlom Tor siete,

- doménové mená spájané s ťažbou kryptomien,
- známe škodlivé TLS certifikáty vybraných C2 frameworkov,
- sieťovú komunikáciu so známymi IP adresami C2 riadiacich serverov,
- sieťovú komunikáciu so známymi doménovými menami C2 riadiacich serverov,
- dopytované URL adresy spájané s C2 riadiacimi servermi,
- sieťovú komunikáciu s potenciálnymi C2 riadiacimi servermi.

Uvedené indikátory, resp. ich vedľajšie produkty dokážeme priradiť príslušným kategóriám, ktoré tvoria spomínaný diagram **Pyramid of Pain**, ako aj technikám, resp. podtechnikám matice **MITRE ATT&CK**. Taktiež si v tejto časti predstavíme rozšírenie C2Hunter, jeho integráciu s aplikáciou C2Detective a taktiež funkcionality, ktoré nám tento nástroj ponúka.

7.1 Detekcia DGA doménových mien

Ako sme už uviedli v predošlej časti, **DGA** je využívané škodlivými aktérmi na dynamické generovanie veľkého počtu pseudonáhodných doménových mien. Následne sa vyberá podmnožina doménových mien, ktoré slúžia pre C2 komunikačný kanál medzi riadiacim serverom útočníka a kompromitovanými systémami. Tento indikátor C2 komunikácie je možné pozorovať vo fáze *Command and Control* a taktiež vo fáze *Actions on Objectives* modelu Cyber Kill Chain.

Na detekciu DGA doménových mien v aplikácii C2Detective využívame nástroj s názvom **DGA Detective**, ktorý bol vyvinutý v rámci celoeurópskeho inovačného projektu *SOCCRATES* [56]. Uvedený nástroj pomocou binárnej klasifikácie určuje, či dané doménové meno bolo vytvorené pomocou DGA, alebo nie. Klasifikácia doménových mien bola trénovaná pomocou konvolučnej neurónovej siete, pričom na cvičenie modelu bol použitý súbor doménových mien, ktorý poskytla nadácia *Shadowserver Foundation*.

Popisovaný nástroj je taktiež distribuovaný ako modul pre programovací jazyk Python [57], čo nám umožnilo jeho jednoduchú integráciu. V detekčnej fáze využívame funkcie tohto modulu na klasifikáciu doménových mien, kde ako vstup poskytujeme extrahované doménové mená z *DNS* požiadaviek, ako aj z hlavičiek *Host* z HTTP spojení. V prípade detekcie DGA doménových mien pomocou nástroja DGA Detective, predmetné doménové mená pridávame do príslušného objektu, ktorý je súčasťou potenciálnych indikátorov

C2 komunikácie. Následne naša aplikácia pridá do celkového hodnotenia analýzy jeden potenciálny indikátor C2 komunikácie.

Detekcia sieťovej komunikácie s DGA doménovými menami predstavuje indikátor C2 komunikácie s veľkou váhou. Z tohto dôvodu je po identifikácii kompromitovaného systému, bezprostredne potrebné vykonať hĺbkovú kontrolu daného systému na výskyt škodlivého kódu.

7.2 Detekcia techniky DNS Tunneling

Techniku **DNS Tunneling**, ktorú škodliví aktéri používajú na ukrytie komunikácie s kompromitovanými systémami pomocou protokolu DNS, sme si už taktiež popísali v predošlej časti tejto práce. Na vytvorenie C2 komunikačného kanálu, útočník vytvorí vlastný DNS server, ktorý má pod kontrolou. Kompromitovaný systém následne posiela DNS dopyty, resp. prijíma DNS odpovede z útočníkom spravovaného DNS servera. Prenášané údaje sú dekódované a ďalej použité na škodlivé účely [58]. Indikátory techniky DNS Tunneling môžeme detegovať vo fázach *Command and Control*, a *Actions on Objectives* modelu Cyber Kill Chain.

V detekčnej fáze nástroja C2Detective porovnávame jednotlivé dĺžky subdomén, ktoré sú súčasťou extrahovaných doménových mien a môžu obsahovať zakódované údaje, s používateľom definovanou hodnotou **MAX_SUBDOMAIN_LENGTH** z konfiguračného súboru. V prípade, že daná subdoména presahuje používateľom definovanú maximálnu dĺžku, je toto doménové meno spolu s príslušnou subdoménou pridané do predmetného objektu, ktorý je súčasťou potenciálnych indikátorov C2 komunikácie. V neposlednom rade si aplikácia zapamätá, že bol detegovaný indikátor a do celkového hodnotenia pridáva jeden potenciálny indikátor C2 komunikácie.

Táto metóda detekcie môže vykazovať viacero falošných pozitív, keďže aj legitímne systémy môžu používať nezvyčajne dlhé doménové, resp. subdoménové mená. Z tohto dôvodu, je po detekcii uvedenej techniky potrebné prešetriť validitu dopytovaných doménových mien predmetným systémom, prípadne zmeniť hodnotu **MAX_SUBDOMAIN_LENGTH** v konfiguračnom súbore a opakovať analýzu PCAP súboru pomocou našej aplikácie C2Detective.

Pri detekcii tejto konkrétnej techniky vieme čiastočne eliminovať falošné pozitíva odfiltrovaním validných doménových mien. Aplikácia C2Detective používa vlastný ***whitelist doménových mien***, do ktorého má používateľ možnosť pridávať také doménové mená, ktorých subdomény nemajú byť v detekčnej fáze pri odhaľovaní indikátorov techniky DNS Tunneling posudzované.

7.3 Detekcia známych škodlivých JA3 odtlačkov

Metóda **JA3**, ktorú sme si predstavili spolu C2 frameworkom Merlin, vytvára MD5 hash odtlačok vybraných polí paketu Client Hello. Tieto **JA3 odtlačky** vieme využiť na detekciu známej škodlivej aktivity v počítačovej sieti. Popisovaný indikátor môžeme detegovať vo fázach *Delivery*, *Exploitation*, *Command and Control* a taktiež vo fáze *Actions on Objectives* modelu Cyber Kill Chain.

Spoločnosť *Salesforce*, ktorá predstavila JA3 metódu, taktiež uverejnila rovnomenný voľne dostupný nástroj s otvoreným zdrojovým kódom [48]. Pred spustením detekčnej fázy, pomocou spomínaného nástroja vytvoríme všetky možné JA3 odtlačky zo vstupného PCAP súboru, ktoré si pre ďalšie analytické úlohy ukladáme do samostatného objektu. Aplikácia C2Detective spracováva pomocou vlastného skriptu voľné dostupné JA3 detekčné pravidlá, ktoré poskytuje spoločnosť *Proofpoint* [59]. Tieto detekčné pravidlá obsahujú pre nás dôležitý typ škodlivého systému, resp. škodlivej aktivity a príslušné MD5 odtlačky, resp. JA3 odtlačky.

V detekčnej fáze vyhľadávame vo vygenerovaných JA3 odtlačkoch zo vstupného PCAP súboru zhodu so škodlivými JA3 odtlačkami zo spracovaného externého zdroju. V prípade zhody, predmetné sieťové spojenie spolu s jeho JA3 odtlačkom pridávame do príslušného objektu, ktorý je súčasťou potenciálnych indikátorov C2 komunikácie. Následne naša aplikácia pridáva do celkového hodnotenia jeden potenciálny indikátor C2 komunikácie.

Aj tento prístup detekcie môže vykazovať viacero falošných pozitív z dôvodu, že rôzne systémy a aplikácie môžu používať rovnakú konfiguráciu, ako známe škodlivé systémy. resp. aplikácie. V praxi je vhodné po detekcii škodlivej aktivity na základe JA3 odtlačku korelovať zistenia s **JA3S** odtlačkami, ktoré sú vytvárané z vybraných polí paketu Server Hello. Popisovaný prístup umožňuje analytikovi overiť škodlivú aktivitu na strane klienta a rovnako na strane servera, čo pri manuálnom overovaní taktiež dokáže odfiltrovať potenciálne falošné pozitíva.

7.4 Detekcia sieťových spojení s nadmernou frekvenciou

Sieťové spojenia s nadmernou frekvenciou predstavujú jav, kedy hostiteľský systém komunikuje s potenciálnym C2 riadiacim serverom s neobvykle vysokou frekvenciou. Tieto požiadavky môžu byť vo forme HTTP, HTTPS, DNS, alebo akéhokoľvek iného protokolu používaného škodlivým softvérom, alebo častou škodlivého kódu. Takéto spojenia predstavujú jeden z prvých potenciálnych indikátorov infekcie malvérom, ktorý zabezpečuje C2 komunikáciu. Pri návrhu a implementácii popisovanej techniky sme sa inšpirovali

podobnou metódou, ktorú poskytuje spomínaný nástroj RITA [60].

Nadmerná frekvencia sieťových spojení môže byť zapríčinená komunikačným vzorom **beaconing**. Táto technika, ktorú malvér používa na vytvorenie pravidelného komunikačného kanála s riadiacim C2 serverom, môže mať podobu jednoduchej správy, ktorá sa v pravidelných intervaloch odosiela na príslušný riadiaci C2 server. Účelom takejto signalizácie je preniesť informáciu riadiacemu C2 serveru, že infikovaný systém je stále aktívny a pripravený prijímať príkazy. Vysokú frekvenciu týchto sieťových spojení môžeme použiť na identifikáciu infikovaného hostiteľa. Na základe kontextu konkrétneho kybernetického útoku dokážeme pozorovať popisovaný indikátor vo fázach *Reconnaissance*, *Delivery*, *Exploitation*, *Command and Control* a v neposlednom rade vo fáze *Actions on Objectives* modelu Cyber Kill Chain.

Pri detekcii sieťových spojení s nadmernou frekvenciou v aplikácii C2Detecive využívame spracované TCP spojenia z poskytnutého PCAP súboru s ich príslušnými frekvenciami. Vo fáze detekcie indikátorov C2 komunikácie, na základe používateľom definovanej hodnoty premennej **MAX_FREQUENCY** v konfiguračnom súbore, vyhľadávame spojenia, ktorých počet predstavuje väčší percentuálny podiel zo všetkých TCP sieťových spojení, ako definovaná hodnota. V prípade, že dané spojenie spĺňa uvedenú podmienku, je pridané do príslušného objektu, ktorý obsahuje takéto spojenia a zároveň je súčasťou potenciálnych indikátorov C2 komunikácie. V neposlednom rade, do celkového hodnotenia detekcie naša aplikácie pridá jeden potenciálny indikátor C2 komunikácie.

Po identifikácii sieťových spojení s nadmernou frekvenciou dokážeme zamerať ďalšie analytické procesy na podozrivé systémy v našej počítačovej sieti a následne prijať relevantné opatrenia na zmiernenie hrozby. Takéto opatrenia môžu zahŕňať blokovanie sieťovej prevádzky k identifikovanému riadiacemu C2 serveru, umiestnenie infikovaných systémov do karantény a vykonanie ďalšej analýzy s cieľom identifikovať škodlivý softvér a jeho schopnosti.

7.5 Detekcia dlhých sieťových spojení

Dlhé sieťové spojenia predstavujú neprerušené pripojenia, ktoré sú otvorené medzi infikovaným hostiteľom a riadiacim C2 serverom po dlhšiu dobu ako je bežné sieťové spojenie. Tento typ spojenia môže indikovať infekciu malvérom, ktorý vyžaduje nepretržitú komunikáciu s riadiacim C2 serverom [61]. Dlhé sieťové spojenie je zvyčajne vytvárané prostredníctvom protokolov, ako je HTTP [62], HTTPS, alebo pomocou vlastného protokolu, ktorý využíva daný malvér.

Po nadviazaní sieťového spojenia infikovaný systém nepretržite odosiela údaje ria-

diacemu C2 serveru, pričom tento server odpovedá príkazmi, alebo aktualizáciami, ktoré má malvér vykonať na infikovanom systéme. Jednou z hlavných výhod dlhých sieťových spojení pre C2 komunikáciu je, že poskytujú trvalý komunikačný kanál s riadiacim C2 serverom, ktorý umožňuje malvéru zachovať si svoje operačné schopnosti. Takéto spojenia sťažujú odhalenie C2 komunikácie, pretože môžu byť súčasťou legitímnej sieťovej prevádzky. Popisovaný indikátor môže byť detegovaný vo fázach *Command and Control* a *Actions on Objectives* modelu Cyber Kill Chain.

Využitý modul **scapy** v aplikácii C2Detective nám poskytuje funkciu s názvom *sessions*, ktorá zoskupuje pakety do sieťových relácií. Pri prehľadávaní relácii, resp. zoznamu paketov, ktoré prislúchajú danej relácii, vypočítavame celkový čas daného spojenia na základe rozdielu časového odtlačku prvého a posledného paketu v aktuálnom zozname. Vypočítanú dĺžku sieťového spojenia následne porovnávame s používateľom definovanou premennou **MAX_DURATION** z konfiguračného súboru. V prípade, že dané sieťové spojenie presahuje maximálnu definovanú dĺžku trvania spojenia, je táto relácia pridaná do príslušného objektu pre takéto relácie, ktorý je taktiež súčasťou potenciálnych indikátorov C2 komunikácie. Aplikácia si následne zapamätá, že bol detegovaný takýto indikátor a do celkového hodnotenia detekcie pridá jeden potenciálny indikátor C2 komunikácie.

Po odhalení dlhých sieťových spojení s riadiacim C2 serverom postupujeme podobne ako pri identifikácii sieťových spojení s nadmernou frekvenciou. Okrem už popísaných krokov, v prípade, že nie je zavedená, je vhodné využiť segmentáciu počítačovej siete, čo môže obmedziť schopnosti malvéru na infikovaných systémoch.

7.6 Detekcia neobvykle veľkých HTTP odpovedí

Neobvykle veľké HTTP odpovede môžu naznačovať škodlivú aktivitu, konkrétne exfiltráciu údajov z kompromitovaného systému. Komunikácia infikovaného hostiteľa s riadiacim C2 serverom môže zahŕňať okrem odosielania a prijímania príkazov aj zakódované, zašifrované, alebo inak ukryté exfiltrované dáta. Z kompromitovaného systému môžu byť dáta exfiltrované pomocou bežne používaných protokolov, zatiaľ čo v našej práci sa zameriavame na protokol HTTP, ktorý sa vo všeobecnosti používa na prenos údajov cez internet. Pri C2 komunikácii môže byť spomínaný protokol HTTP použitý na prenos údajov medzi napadnutým systémom a riadiacim serverom útočníka [63]. Z dôvodu, že neobvykle veľké HTTP odpovede môžu byť prejavom aj inej aktivity ako len výhradne exfiltrácii dát, tento indikátor prislúcha viacerým fázam modelu Cyber Kill Chain v závislosti od kontextu daného kybernetického útoku. Vo všeobecnosti môže byť popisovaný indikátor pozorovaný vo fázach *Reconnaissance*, *Delivery*, *Exploitation*, *Command and Control* a v

neposlednom rade vo fáze *Actions on Objectives*.

HTTP hlavička s názvom *Content-Length* definuje veľkosť tela správy v bajtoch. Táto veľkosť zahŕňa aj kódovania obsahu, čo znamená, že ak je obsah správy kódovaný, uvedená veľkosť v hlavičke Content-Length je veľkosťou komprimovaného obsahu a nie jeho pôvodnej veľkosti. Ak uvedená veľkosť v hlavičke Content-Length nezodpovedá veľkosti správy, jej obsah bude skrátený, resp. sa doplní nulovými hodnotami na danú dĺžku [62].

Pri extrakcii dát z poskytnutého PCAP súboru v aplikácii C2Detective sa mimo iné zameriavame aj na HTTP požiadavky a odpovede, z ktorých si vybrané hodnoty ukladáme do samostatného objektu. Následne vo fáze detekcie indikátorov C2 komunikácie prehľadávame spracované HTTP relácie, pričom sa zameriavame na hodnoty v hlavičkách *Content-Length*. Extrahované hodnoty porovnávame s používateľom definovanou hodnotou **MAX_HTTP_SIZE** z konfiguračného súboru. V prípade, že veľkosť tela HTTP správy presahuje definovanú prahovú hodnota, dané spojenie pridávame do príslušného objektu, ktorý je súčasťou potenciálnych indikátorov C2 komunikácie. V neposlednom rade si aplikácia zapamätá, že bol detegovaný tento indikátor a do celkového hodnotenia pridá potenciálny indikátor C2 komunikácie.

V prípade detekcie neobvykle veľkých HTTP odpovedí v praxi, zameriavame ďalšie analytické procesy na predmetné spojenia a pokúšame sa získať obsah daných odpovedí. Tak ako pri predošlých prahových hodnotách, aj v tomto prípade je možné, že sú detegované falošné pozitíva a je potrebné upraviť prahovú hodnotu, alebo ak je to možné, vykonať kontrolu konfigurácie daného systému.

7.7 Detekcia Tor sieťovej prevádzky

Anonymizačnú sieť **Tor** môžu škodliví aktéri využiť na ukrytie svojej identity, čo pri obrane systémov sťažuje identifikáciu a blokovanie škodlivej sieťovej prevádzky. Túto službu môžu útočníci použiť na vytvorenie C2 komunikačného kanála medzi svojím riadiacim serverom a kompromitovanými systémami. Exfiltrácia dát pomocou siete Tor má niekoľko obmedzení. Jedným z hlavných obmedzení je, že škodlivý aktér musí mať na kompromitovanom systéme prístup k softvéru, ktorý umožňuje komunikáciu cez Tor protokol. V prípade, že takýto softvér nie je nainštalovaný na danom systém, nie je možné použiť túto sieť na exfiltráciu dát. Ďalším obmedzením je, že výkonnosť protokolu Tor je obmedzovaná napríklad sieťovou rýchlosťou jednotlivých uzlov siete, ktoré tvoria komunikačnú cestu, alebo celkovou záťažou Tor siete. Tieto faktory môžu spôsobiť oneskorenie, alebo úplné prerušenie komunikácie s infikovaným systémom, čo môže v konečnom dôsledku znefunkčniť exfiltráciu dát. Používanie anonymizačnej siete Tor škodlivými aktérmi môžeme priradiť

prvej a šiestej fáze modelu Cyber Kill Chain, teda fázam *Reconnaissance*, resp. *Command and Control*. Pre účely C2 komunikácie využívali, alebo využívajú sieť Tor napríklad skupina **APT28** a malvér **Industroyer**, **Ursnif** a **WannaCry** [64].

Aplikácia C2Detective spracováva pomocou vlastného skriptu voľne dostupné zoznamy IP adries systémov Tor siete [65]. V detekčnej fáze vyhľadávame v externých TCP spojeniach, také sieťové spojenia, ktoré komunikovali s týmito spracovanými IP adresami. V prípade, že nachádzame TCP spojenie s predmetnými IP adresami, tieto spojenia pridávame do príslušného objektu, ktorý je súčasťou potenciálnych indikátorov C2 komunikácie. Následne naša aplikácia pridáva do celkového hodnotenia jeden potenciálny indikátor C2 komunikácie.

Na tomto mieste je potrebné uviesť, že pri detekcii Tor sieťovej prevádzky v našej aplikácii rozlišujeme, či sa jedná o komunikáciu so *vstupnými*, alebo *výstupnými* uzlami Tor siete. Uvedené prípady sú v aplikácii C2Detective reprezentované ako samostatné potenciálne indikátory C2 komunikácie. Je to z dôvodu, že v prípade detekcie Tor sieťovej prevádzky ku vstupným uzlom Tor siete detegujeme sieťovú komunikáciu, ktorej zdrojom je systém v našej počítačovej sieti. Na druhej strane, v prípade detekcie Tor sieťovej prevádzky z výstupných uzlov Tor siete, detegujeme sieťovú komunikáciu, ktorej zdrojom je systém mimo našej počítačovej sieti. Teda jedná sa o sieťovú prevádzku, ktorá prichádza do našej infraštruktúry z internetu.

Z dôvodu, že sieť Tor je možné používať aj na legitímne účely, akými sú ochrana súkromia, alebo obchádzanie internetovej cenzúry, jej úplné blokovanie môže mať negatívny vplyv na používateľov v danej sieťovej infraštruktúre. Po identifikácii Tor sieťovej prevádzky je potrebné prešetriť, či bolo jej použitie úmyselné, v opačnom prípade je nutné vykonať hĺbkovú kontrolu daného systému na výskyt škodlivého kódu.

7.8 Detekcia dopytov doménových mien spájaných s ťažbou kryptomien

Cryptojacking [66] je technika, pri ktorej škodliví aktéri využívajú malvér, alebo iný škodlivý softvér, na prevzatie výpočtového výkonu systému s cieľom ťažby kryptomien bez vedomia, alebo súhlasu jeho správcu, resp. jeho majiteľa. Odchádzajúca sieťová prevádzka k doménovým menám, ktoré sú spájané s ťažbou kryptomien a technikou cryptojacking, môže byť indikátorom kompromitácie systému. Ťažbu kryptomien na neoprávnených systémoch môžeme priradiť piatej fáze modelu Cyber Kill Chain, ktorá sa nazýva *Installation*.

V aplikácii C2Detecive využívame voľne dostupný zoznam doménových mien spájaných s ťažbou kryptomien a technikou cryptojacking, ktorý udržiava a poskytuje *The Block List*

Project [67]. V detekčnej fáze našej aplikácie vyhľadávame v tomto zozname extrahované dopytované doménové mená z poskytnutého PCAP súboru, pričom analyzované doménové mená získavame z paketov, ktoré obsahujú DNS požiadavky, pri počiatočnom spracovaní vstupného súboru. V prípade, že je potvrdený výskyt, potenciálne škodlivé doménové meno pridávame do príslušného objektu, ktorý je súčasťou potenciálnych indikátorov C2 komunikácie. Aplikácia následne pridáva do celkového hodnotenia jeden potenciálny indikátor C2 komunikácie.

Po detekcii dopytov doménových mien, ktoré sú spájané s ťažbou kryptomien je nevyhnutné začať dôkladne monitorovať činnosť predmetného systému a identifikovať akékoľvek podozrivé správanie. Takéto správanie môže napríklad zahŕňať náhle zvýšenie využitia procesora, alebo grafickej karty. Vo všeobecnosti môžeme nakonfigurovať bezpečnostné riešenia tak, aby blokovali prístup na vybrané škodlivé webové stránky, čo nám poskytuje mechanizmus **DNS sinkhole**. Týmto spôsobom dokážeme teoreticky zabrániť používateľom v neúmyselnom prístupe na tieto stránky.

7.9 Detekcia známych hodnôt TLS certifikátov vybraných C2 frameworkov

Detekcia predvolených nastavení vybraných C2 frameworkov nám umožňuje detegovať potenciálnu C2 komunikáciu s menším výpočtovým zaťažením. Takéto nastavenia C2 frameworkov môžu využívať napríklad menej pokročilí škodliví aktéri.

Z otvorených zdrojov sme spracovali techniku, ktorá využíva vyhľadávanie v službe **Shodan** v spojení s technikou ***fingerprinting*** [68]. Túto techniku si bližšie priblížime pri popise rozšírenia C2Hunter. Vybrané vyhľadávacie reťazce pre službu Shodan [69], ktoré obsahujú hodnoty z polí TLS certifikátov, sme predspracovali do súboru formátu JSON. V tomto súboru, kľúče reprezentujú názvy C2 frameworkov a príslušné hodnoty sú extrahované z predmetných vyhľadávacích reťazcov. Takto spracované údaje môžeme považovať za *taktické informácie*, ktoré sú produktom procesov CTI. Hodnoty predvolených nastavení vybraných C2 frameworkov uvádzame pre prehľad v tabuľke nižšie, pričom znak plus označuje, že v TLS certifikáte musia byť obsiahnuté oba reťazce.

C2 framework	Pole TLS certifikátu	Škodlivá hodnota
Cobalt Strike	Serial Number	146473198
Metasploit	Common Name (CN) - Issuer	MetasploitSelfSignedCA
Covenant	Common Name (CN) - Issuer / Subject	Covenant
Mythic	Organization (0) - Issuer / Subject	Mythic
Sliver	Common Name (CN) - Issuer / Subject	multiplayer + operators
PoshC2	Common Name (CN) - Issuer / Subject	P18055077

Tabuľka 6: Známe hodnoty TLS certifikátov vybraných C2 frameworkov

V detekčnej fáze aplikácie C2Detective vyhľadávame v extrahovaných dátach z polí TLS certifikátorov z odchytených paketov hodnoty, ktoré sa môžu vyskytovať v poliach TLS certifikátov C2 frameworkov, ktoré používajú predvolené nastavenia. V prípade zhody vo vyhľadávaní, predmetné extrahované polia TLS certifikátu spolu s názvom C2 frameworku a detegovanou hodnotou pridávame do príslušného objektu, ktorý je súčasťou potenciálnych indikátorov C2 komunikácie. Následne naša aplikácia pridá do celkového hodnotenia analýzy jeden potenciálny indikátor C2 komunikácie.

7.10 Využitie rozšírenia C2Hunter

Nástroj **C2Hunter** [70] bol spočiatku vyvíjaný nezávisle od aplikácie C2Detective. Medzi jeho hlavné funkcionality patrí vyhľadanie v službe Shodan s cieľom detegovať predvolené nastavania známych C2 frameworkov, agregácia threat feedov **Feodo Tracker** [71], **ThreatFox** [52] a **URLhaus** [53] a v neposlednom rade vyhľadávanie aktívnych systémov v spracovaných dátach na základe geografickej lokácie, ktorú používateľ definuje v konfiguračnom súbore.

Naša aplikácia C2Detective využíva tento nástroj ako rozšírenie na detekciu C2 komunikácie so známymi škodlivými systémami. Konkrétne využíva dáta spracovávané z uvedených threat feedov, ktoré nám poskytujú *taktické informácie* a údaje zo služby Shodan. Tieto údaje sú ukladané do príslušných tabuliek lokálnej SQLite databázy. To znamená, že až po správnom nakonfigurovaní a spracovaní spomínaných údajov je možné nástroj C2Hunter použiť ako rozšírenie aplikácie C2Detective.

Použitie popisovaného rozšírenia predstavuje voliteľnú funkcionalitu pre našu aplikáciu, teda jeho použitie, resp. nepoužitie, je na koncovom používateľovi. Konfiguráciu samotného nástroja C2Hunter, ako aj jeho využitie s aplikáciou C2Detective bližšie popisujeme v používateľskej príručke, ktorá je súčasťou príloh tejto práce. V tejto časti si ďalej popíšeme

detekčné funkcionality, ktoré poskytuje rozšírenie C2Hunter pre našu aplikáciu.

7.10.1 Detekcia sieťovej komunikácie so známymi C2 riadiacimi servermi

V detekčnej fáze našej aplikácie je možné vyhľadávať extrahované verejné IP adresy, doménové mená a URL adresy v lokálnej databáze nástroja C2Hunter. To znamená, že dokážeme detegovať **známe škodlivé IP adresy** a **známe škodlivé doménové mená** C2 riadiacich serverov, ktoré prijali, alebo iniciovali spojenie, ako aj **známe škodlivé URL adresy**, ktoré sú spájané s C2 komunikáciou. Uvedené indikátory je možné pozorovať v ľubovoľnej fáze modelu Cyber Kill Chain na základe kontextu daného kybernetického útoku.

C2 IP adresy sú obsiahnuté vo všetkých spracovaných threat feedov nástrojom C2Hunter, teda Feodo Tracker, ThreatFox a URLhaus. Doménové mená je možné vyhľadávať v dátach threat feedu ThreatFox a v neposlednom rade, URL adresy sú súčasťou threat feedu ThreatFox a URLhaus. Jednotlivé extrahované údaje, verejné IP adresy, doménové mená a URL adresy, z poskytnutého PCAP súboru, vyhľadávame v tabuľkách spomenutej databázy, pričom v prípade nálezu je daný indikátor C2 komunikácie priradený do príslušného objektu, ktorý je súčasťou potenciálnych indikátorov C2 komunikácie. Rovnako, ako pri predošlých detekčných funkcionalitách si naša aplikácia pridá do celkového hodnotenia jeden, resp. dva alebo tri potenciálne indikátory C2 komunikácie.

Pred dotazovaním databázy vytvárame príslušné *dynamické SQL príkazy*, ktoré obsahujú hodnoty z podzoznamov extrahovaných IP adries, doménových mien a URL adries. Počet hodnôt podzoznamov, teda IP adries, doménových mien a URL adries v jednom dotaze na databázu má používateľ možnosť definovať v konfiguračnom súbore aplikácie, konkrétne v objekte ***settings*** a premennej *chunk_size*.

7.10.2 Detekcia sieťovej komunikácie s potenciálnymi C2 riadiacimi servermi

Touto technikou sme sa prvotne inšpirovali pri implementácii už popísanej funkcionality detekcie známych hodnôt TLS certifikátov vybraných C2 frameworkov z extrahovaných dát z poskytnutého PCAP súboru. Výskumník vo svojom článku [68] uvádza rôzne *vyhľadávacie reťazce* [69], pomocou ktorých deteguje služby z kategórie **MaaS**. Vyhľadávania v službe Shodan tento výskumník rozšíril o také, pomocou ktorých vieme detegovať **predvolené nastavenia známych C2 frameworkov**. Detekcia je vykonávaná na základe zhodujúcich sa polí v odpovediach zo systémov, napríklad HTTP hlavičky, hodnoty polí TLS certifikátu, ale aj hash odtlačok faviconu a iné. Tento prístup umožňuje budovať lokálnu databázu

potenciálnych C2 riadiacich serverov pre nástroj C2Hunter, čo môžeme následne využiť na *proaktívne* vyhľadávanie C2 komunikácie v sieťovej infraštruktúre.

Počas detekčnej fázy našej aplikácie je možné vyhľadávať extrahované verejné IP adresy v predmetnej tabuľke predpripravenej lokálnej databázy nástroja C2Hunter, ktorá obsahuje aj informácie o potenciálnych C2 riadiacich serverov. V prípade výskytu sieťovej komunikácie s potenciálnym C2 serverom, sú informácie o takomto spojení pridané do príslušného objektu, ktorý je súčasťou potenciálnych indikátorov C2 komunikácie. Tak ako pri predošlých funkcionalitách, aj v tomto prípade si naša aplikácia pridá do celkového hodnotenia jeden potenciálny indikátor C2 komunikácie.

8 Testovanie detekčných funkcionalít

Implementované detekčné metódy aplikácie **C2Detective** sme testovali pomocou PCAP súborov, ktoré obsahujú C2 komunikáciu a s ňou spojené škodlivé aktivity. Všetky popísané testovania boli vykonané s nakonfigurovaným rozšírením **C2Hunter**, ktorý mal v čase vykonávania týchto testov spracované dostupné threat feedy ku dňu *02. 05. 2023*. Všetky testovacie PCAP súbory, ktoré boli použité na testovacie účely sú priložené v prílohe tejto práce.

Prvý testovací súbor s názvom *„2023-01-16-IcedID-infection-with-Backonnect-and-VNC-and-Cobalt-Strike.pcap“* sme prevzali z voľne dostupného zdroju s názvom **Malware Traffic Analysis**, ktorý spravuje výskumník Brad Duncan [72]. Prehľad prahových hodnôt pre dané testovanie, s ktorými sme spustili aplikáciu C2Detective môžeme vidieť v tabuľke 7.

Premenná	**Hodnota**
MAX_FREQUENCY	10
MAX_DURATION	3600
MAX_HTTP_SIZE	1000000
MAX_SUBDOMAIN_LENGTH	30

Tabuľka 7: Konfigurácia detekčných prahových hodnôt

Z celkových možných trinástich potenciálnych indikátorov pri danej konfigurácii boli detegované štyri indikátory C2 komunikácie, pričom detekcia DGA doménových mien nebola povolená. Konkrétne boli detegované indikátory C2 komunikácie

- sieťové spojenia s nadmernou frekvenciou,
- dlhé sieťové spojenia,
- sieťová komunikácia so známymi IP adresami C2 riadiacich serverov,
- sieťová komunikácia so známymi doménovými menami C2 riadiacich serverov.

Nasledujúce obrázky zobrazujú detegované indikátory spracované vo výstupnej správe o výsledkoch analýzy. Na obrázku 7 vidíme detegované sieťové spojenia s nadmernou frekvenciou medzi IP adresami 10.1.11.101 a 51.195.169.87. Tieto TCP sieťové spojenia predstavovali väčší podiel zo všetkých TCP spojení v percentách na základe definovanej

prahovej hodnoty **MAX_FREQUENCY**. Na obrázku 8 vidíme detegované dlhé sieťové spojenia medzi IP adresami 10.1.11.101 a 23.254.202.234, ktorých dĺžka presahovala nakonfigurovanú prahovú hodnotu **MAX_DURATION**. Ako môžeme vidieť, celková dĺžka sieťového spojenia medzi týmito systémami bola 5178 sekúnd, resp. 1 hodina 26 minút a 18 sekúnd. Ďalším detegovaným indikátorom bola sieťová komunikácia so známymi IP adresami C2 riadiacich serverov, ktoré môžeme vidieť na obrázku 9. Uvedený indikátor vo výstupnej správe o výsledkoch analýzy obsahuje taktiež detailný prehľad spojení zobrazený v samostatnej tabuľke, ktorý tu z dôvodu veľkého rozsahu nezobrazujeme. Na obrázku 10 vidíme detegované známe doménové mená C2 riadiacich serverov zo spracovanej sieťovej komunikácie. Tretí a štvrtý indikátor, teda sieťová komunikácie so známymi IP adresami a známymi doménovými menami C2 riadiacich serverov, boli detegované na základe údajov, ktoré nám poskytlo rozšírenie C2Hunter.

Source IP	Source Port	Destination IP	Destination Port	Frequency
10.1.11.101	64823	51.195.169.87	8080	7557
51.195.169.87	8080	10.1.11.101	64823	8050

Obr. 7: Sieťové spojenia s nadmernou frekvenciou v prvej testovacej vzorke

Source IP	Source Port	Destination IP	Destination Port	Duration
10.1.11.101	64638	23.254.202.234	443	5178
23.254.202.234	443	10.1.11.101	64638	5178

Obr. 8: Dlhé sieťové spojenia v prvej testovacej vzorke

C2 IP address
45.12.109.195
23.254.202.234
23.227.202.188
51.195.169.87
89.44.9.157
5.230.74.203

Obr. 9: Známe IP adresy C2 riadiacich serverov v prvej testovacej vzorke

C2 Domain Name
siantdarik.lol
dgormiugatox.com
dgormiugatox.com
felzater.lol
ijoyzymama.com
nindaxloart.com
clarkitservices.com

Obr. 10: Známe doménové mená C2 riadiacich serverov v prvej testovacej vzorke

Druhý testovací súbor s názvom *„zeus-dga.pcap“* sme prevzali z verejného repozitára, ktorý obsahuje rôzne testovacie PCAP súbory pre IPS riešenia [73]. V tomto prípade sme sa zamerali na testovanie detekcie **DGA doménových mien**. Toto testovanie sme povolili dostupným prepínačom aplikácie C2Detective. Z dôvodu, že tento súbor bol pôvodným autorom vygenerovaný s cieľom testovania detekcie DGA doménových mien, neobsahuje iné indikátory C2 komunikácie ako uvedené DGA doménové mená. Na obrázku 11 môžeme vidieť informáciu o detekcii DGA doménových mien v rozhraní príkazového riadka. Predmetné doménové mená sú taktiež obsiahnuté aj v správe o výsledkoch analýzy druhej testovacej vzorky, ktorá je priložená vo formáte HTML a PDF v prílohe tejto práce.

```
[11:07:18] [INFO] Detected domain names generated by Domain Generation Algorithms (DGA)
[11:07:18] [INFO] Listing detected domain names generated by Domain Generation Algorithms (DGA)
>> zbgrqobsiqrpv.com
>> xxcnirvbqivbucfsbliu.com
>> eoooghrkxhm.com
>> hqhjoeiiydul.com
>> csmanuivsrlx.com
>> lukhhthjxul.com
>> jgjvgfetpammdrxwn.com
>> yqpfpfwcmnaoptgibg.com
>> tpqxtpaxlslldu adgpl.com
```

Obr. 11: DGA doménové mená v druhej testovacej vzorke

Pri treťom testovaní sme sa zamerali na detekciu techniky **DNS Tunneling**, pri ktorej sme použili súbor s názvom *„dnscat2_dns_tunneling_1hr.pcap“*. Tento voľne dostupný PCAP súbor je súčasťou internetového článku na tému detekcie techniky DNS Tunneling [74]. Z dôvodu, že aj tento súbor bol vygenerovaný so špecifickým zámerom, neobsahuje iné indikátory C2 komunikácie ako testovaná technika. Aplikácia C2detective s nastavenými prahovými hodnotami z prvého testovania, pre uvedený vstupný súbor detegovala techniku DNS Tunneling a to konkrétne 6980 DNS dopytov domény *„cisco-update.com“*, pričom dĺžky subdomén predmetnej domény presahovali nakonfigurovanú prahovú hodnotu **MAX_SUBDOMAIN_LENGTH**. Informáciu o detekcií predmetnej techniky v rozhraní príkazového riadku vidíme na obrázku 12. Úplný prehľad dopytov je obsiahnutý vo výstupnej správe o výsledkoch analýzy, ktorá je taktiež priložená vo formáte HTML a PDF v prílohe tejto práce.

```
[11:50:36] [INFO] Detected DNS Tunneling technique
[11:50:36] [INFO] Listing information about detected DNS Tunneling technique
>> Queried 6890 unique subdomains for 'cisco-update.com'
```

Obr. 12: Informácie o technike DNS Tunneling v tretej testovacej vzorke

Pre testovanie schopností detekcie ďalších indikátorov C2 komunikácie sme si v simulovanom prostredí odchytili sieťovú prevádzku, ktorá obsahovala škodlivú aktivitu. Predmetný PCAP súbor s názvom *„test_4.pcap“*, sme tak ako aj predošlé spomenuté PCAP súbory priložili medzi prílohy tejto práce. Z celkových možných štrnástich, resp. trinástich potenciálnych indikátorov pri nezmenenej konfigurácii bolo detegovaných šesť indikátorov C2 komunikácie, pričom detekcia DGA doménových mien nebola povolená. Konkrétne boli detegované indikátory C2 komunikácie

- známe hodnoty TLS certifikátov vybraných C2 frameworkov,
- Tor sieťová prevádzka,

- dopyty doménových mien spájaných s ťažbou kryptomien,
- sieťová komunikácia so známymi doménovými menami C2 riadiacich serverov,
- dopytovanie URL adries, ktoré sú spájané s C2 riadiacimi servermi,
- sieťová komunikácia s potenciálnymi C2 riadiacimi servermi.

Obrázky umiestnené nižšie zobrazujú detegované indikátory spracované vo výstupnej správe o výsledkoch analýzy. Na obrázku 13 vidíme detegované IP adresy uzlov, ktoré sú súčasťou Tor siete a taktiež vidíme detailný prehľad spojení s týmito detegovanými IP adresami zobrazený v samostatnej tabuľke. Obrázok 14 zobrazuje detegovanú známu hodnotu TLS certifikátu, konkrétne reťazec *„Covenant“* rovnomenného C2 frameworku Covenant. Ako môžeme vidieť vo výstupnej správe o výsledkoch analýzy, v príslušnej tabuľke, označenia *None* symbolizujú, že dané polia v predmetnom TLS certifikáte neobsahovali žiadne údaje. Tretím detegovaným indikátorom C2 komunikácie boli dopyty doménových mien spájaných s ťažbou kryptomien. Jednotlivé dopytované doménové mená vidíme na obrázku 15, pričom hodnotu *„www.google-analytics.com“* považujeme za falošné pozitívum z dôvodu, že sa v skutočnosti nejedná o doménové meno, ktoré je spájané s ťažbou kryptomien. Detekcia tejto konkrétnej hodnoty bola zapríčinená jej výskytom v spracovanom externom zdroji, ktorý je využívaný na detekciu takýchto doménových mien. Na obrázku 16 vidíme detegovanú URL adresu, ktorá je spájaná s C2 aktivitami, pričom detegovaná sieťová komunikácia so známymi doménovými menami C2 riadiacich serverov obsahovala doménové meno predmetnej URL adresy. C2 URL adresa a jej príslušné doménové meno boli detegované na základe IoCs, ktoré nám poskytlo rozšírenie C2Hunter. Obrázok 17 zobrazuje detegovanú IP adresu a sieťové spojenia s ňou, ktorá bola taktiež detegovaná na základe údajov, ktoré nám poskytlo rozšírenie C2Hunter. Konkrétne sa jedná o systém, ktorý bol označený ako C2 riadiaci server na základe výstupov techniky ***fingerprinting***. Kompletná správa o výsledkoch analýzy štvrtej testovacej vzorky je priložená vo formáte HTML a PDF v prílohe tejto práce.

Tor Node IP Address
188.68.56.181
212.147.124.158
202.61.197.87

Timestamp	Source IP	Source Port	Destination IP	Destination Port
2023-05-03 21:35:53	212.147.124.158	9001	10.0.2.15	45196
2023-05-03 21:35:53	10.0.2.15	45196	212.147.124.158	9001
2023-05-03 21:35:57	10.0.2.15	43014	188.68.56.181	9001
2023-05-03 21:35:57	188.68.56.181	9001	10.0.2.15	43014
2023-05-03 21:35:59	10.0.2.15	52602	202.61.197.87	9001
2023-05-03 21:35:59	202.61.197.87	9001	10.0.2.15	52602

Obr. 13: Tor sieťová prevádzka vo štvrtej testovacej vzorke

General Information		Connection			
C2 Framework	Malicious Value	Source IP	Source Port	Destination IP	Destination Port
Covenant	Covenant	109.123.231.70	7443	10.0.2.15	39634

Obr. 14: Známa hodnota TLS certifikátu C2 frameworku vo štvrtej testovacej vzorke

Crypto Domain Name
xmr.nanopool.org
zec.nanopool.org
www.google-analytics.com
etc.nanopool.org
nanopool.org

Obr. 15: Doménové mená spájané s ťažbou kryptomien vo štvrtej testovacej vzorke

C2 URL
http://voiceinfosys.net/af

Obr. 16: Dopytovaná C2 URL adresa vo štvrtej testovacej vzorke

Potential C2 IP address
109.123.231.70

Timestamp	Source IP	Source Port	Destination IP	Destination Port
2023-05-03 21:35:59	10.0.2.15	33284	109.123.231.70	7443
2023-05-03 21:35:59	109.123.231.70	7443	10.0.2.15	33284
2023-05-03 21:36:00	10.0.2.15	39634	109.123.231.70	7443
2023-05-03 21:36:00	109.123.231.70	7443	10.0.2.15	39634

Obr. 17: Potenciálna C2 sieťová komunikácia vo štvrtej testovacej vzorke

Pri poslednom testovaní sme použili PCAP súbor s názvom *„2022-09-23-IcedID-infection-with-Cobalt-Strike.pcap“* [75], ktorý sme prevzali z už spomínaného externého zdroju s názvom Malware Traffic Analysis. Pre testovacie účely bola znížená prahová hodnota **MAX_HTTP_SIZE**. Prehľad nakonfigurovaných prahových hodnôt pre posledné testovanie, s ktorými sme spustili aplikáciu C2Detective môžeme vidieť v tabuľke 8.

Premenná	**Hodnota**
MAX_FREQUENCY	10
MAX_DURATION	3600
MAX_HTTP_SIZE	500000
MAX_SUBDOMAIN_LENGTH	30

Tabuľka 8: Zmenená konfigurácia detekčných prahových hodnôt

Z celkových možných štrnástich, resp. trinástich potenciálnych indikátorov pri danej konfigurácii bolo detegovaných šesť indikátorov C2 komunikácie, pričom detekcia DGA doménových mien nebola pri tomto testovaní povolená. Konkrétne boli detegované indikátory C2 komunikácie

- sieťové spojenia s nadmernou frekvenciou,
- neobvykle veľké HTTP odpovede,
- známe škodlivé JA3 odtlačky,
- sieťová komunikácia so známymi IP adresami C2 riadiacich serverov,
- sieťová komunikácia so známymi doménovými menami C2 riadiacich serverov,
- dopytovanie URL adries, ktoré sú spájané s C2 riadiacimi servermi.

Príklady výsledkov detekcie väčšiny z uvedených indikátorov C2 komunikácie sme si už predstavili v predchádzajúcich testoch. Ako môžeme vidieť, pre danú vzorku sme detegovali dva nové indikátory C2 komunikácie. Na obrázku 18 môžeme vidieť časť výpisu z rozhrania príkazového riadku, ktorý nás informuje o tom, že bolo detegovaných celkovo tridsaťdva potenciálnych škodlivých JA3 odtlačkov. Predmetné škodlivé JA3 odtlačky spolu s príslušnými informáciami sú obsiahnuté v správe o výsledkoch analýzy piatej testovacej vzorky, ktorá je priložená vo formáte HTML a PDF v prílohe tejto práce. Ďalším novým detegovaným indikátorom C2 komunikácie boli neobvykle veľké HTTP odpovede. Ako sme uviedli, bolo potrebné znížiť hodnotu premennej **MAX_HTTP_SIZE** aby bol daný indikátor zachytený. Táto skutočnosť nemení nič na tom, že daná detekčná metóda funguje správne a teda úspešne deteguje HTTP odpovede, ktorých veľkosť presahuje definovanú prahovú hodnotu **MAX_HTTP_SIZE**. Predmetné HTTP spojenie, ktorého veľkosť obsahu bola vyhodnotená ako neobvykle veľká, môžeme vidieť na obrázku 19.

```
[01:06:07] [INFO] Looking for known malicious JA3 fingerprints ...
[01:06:07] [INFO] Detected known malicious JA3 fingerprints
[01:06:07] [INFO] Listing information about detected JA3 fingerprints
>> Found 32 potentially malicious JA3 fingerprint matches
```

Obr. 18: Známe škodlivé JA3 odtlačky v piatej testovacej vzorke

```
[01:06:07] [INFO] Looking for unusual big HTTP message body size ...
[01:06:07] [INFO] Detected unusual big HTTP message body size
[01:06:07] [INFO] Listing connections with unusual big HTTP message body size
>> 137.184.114.20:80 -> 10.9.23.101:58592 = 678971 bytes
```

Obr. 19: Neobvykle veľká HTTP odpoveď v piatej testovacej vzorke

8.1 Porovnanie detekčných schopností s nástrojom RITA

Pre porovnanie detekčných schopností C2 komunikácie našej aplikácie s iným riešením sme zvolili už spomínaný nástroj **RITA**. Ako testovaciu vzorku sme použili PCAP súbor z prvého testovania. Správcovia projektu RITA poskytujú inštalačný skript, ktorý ale ako sme zistili až pri samotnej inštalácii, podporuje iba vybrané Linuxové distribúcie, konkrétne *Ubuntu*, *Debian*, *CentOS* a *Red Hat Enterprise Linux* a ich vybrané verzie, ktoré nikde neuvádzajú.

Po úspešnej inštalácii bolo potrebné transformovať testovací PCAP súbor pomocou nástroja **Zeek** z dôvodu, že aplikácia RITA nepodporuje na vstupe PCAP súbory. Až po transformovaní PCAP súboru na požadovaný formát sme mohli daný dataset importovať do aplikácie. Testovaná aplikácia po načítaní vstupných dát neupozorní automatizovaným spôsobom na detegované potenciálne indikátory C2 komunikácie ako naša aplikácia a je teda potrebné manuálne dohľadávať výsledky analýzy.

Ako môžeme vidieť na obrázkoch nižšie, pomocou dostupných prepínačov nástroja RITA sme postupne zobrazovali výstupy analýzy nášho vstupného súboru. Na obrázku 20 môžeme vidieť detegované dve sieťové spojenia, ktoré boli vyhodnotené ako potenciálna C2 komunikácia. Pri spustení príkazu *show-strobes* ale neboli nájdené žiadne informácie o komunikačnom vzore **beaconing**. Následne sme si taktiež vypísali spracované HTTP hlavičky *User-Agent*. Na obrázku 21 vidíme doménové mená, ktoré nástroj RITA získal z analýzy DNS. Pomocou príkazov *show-bl* sme si chceli vypísať detegované známe škodlivé IP adresy a doménové mená, ktoré prijali, alebo iniciovali sieťové spojenia, ale ako môžeme vidieť žiadne takéto indikátory neboli detegované.

```
martin@test:~$ rita show-beacons -H test1
+-------+-------------+----------------+-------------+------------+-------------+----------+----------+-----------+
| SCORE |  SOURCE IP  | DESTINATION IP | CONNECTIONS | AVG  BYTES | TOTAL BYTES | TS SCORE | DS SCORE | DUR SCORE |
+-------+-------------+----------------+-------------+------------+-------------+----------+----------+-----------+
| 0.994 | 10.1.11.101 | 45.12.109.195  |          30 |       3109 |       93278 |    0.999 |    0.996 |      0.98 |
| 0.508 | 10.1.11.101 | 23.227.202.188 |         168 |      16755 |     2814911 |    0.934 |    0.997 |       0.1 |
+-------+-------------+----------------+-------------+------------+-------------+----------+----------+-----------+
martin@test:~$ rita show-strobes -H test1
No results were found for test1
martin@test:~$ rita show-useragents -H test1
+-------------------------------------------------------------------------------------------------------------+------------+
|                                                USER AGENT                                                   | TIMES USED |
+-------------------------------------------------------------------------------------------------------------+------------+
| Mozilla/5.0 (iPhone; CPU iPhone OS 12_0 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko)             |        168 |
| Version/12.0                                                                                                |            |
| No JA3 hash generated                                                                                       |         44 |
| Empty user agent string                                                                                     |          1 |
+-------------------------------------------------------------------------------------------------------------+------------+
```

Obr. 20: Analýza komunikačného vzoru beaconing pomocou nástroja RITA

```
martin@test:~$ rita show-exploded-dns -H test1
+---------------------------------+--------------------+-----------------+
|             DOMAIN              | UNIQUE SUBDOMAINS | TIMES LOOKED UP |
+---------------------------------+--------------------+-----------------+
| microsoft.com                   |                  3 |              10 |
+---------------------------------+--------------------+-----------------+
| data.microsoft.com              |                  2 |               8 |
+---------------------------------+--------------------+-----------------+
| siantdarik.lol                  |                  1 |              25 |
+---------------------------------+--------------------+-----------------+
| events.data.microsoft.com       |                  1 |               5 |
+---------------------------------+--------------------+-----------------+
| v10.events.data.microsoft.com   |                  1 |               5 |
+---------------------------------+--------------------+-----------------+
| settings-win.data.microsoft.com |                  1 |               3 |
+---------------------------------+--------------------+-----------------+
| checkappexec.microsoft.com      |                  1 |               2 |
+---------------------------------+--------------------+-----------------+
| ijoyzymama.com                  |                  1 |               1 |
+---------------------------------+--------------------+-----------------+
| clarkitservices.com             |                  1 |               1 |
+---------------------------------+--------------------+-----------------+
| felzater.lol                    |                  1 |               1 |
+---------------------------------+--------------------+-----------------+
| nindaxloart.com                 |                  1 |               1 |
+---------------------------------+--------------------+-----------------+
martin@test:~$ rita show-bl-hostnames -H test1
No results were found for test1
martin@test:~$ rita show-bl-source-ips -H test1
No results were found for test1
martin@test:~$ rita show-bl-dest-ips -H test1
No results were found for test1
martin@test:~$ rita html-report test1
[-] Writing: /home/martin/test11/test1
[-] Wrote outputs, check /home/martin/test11 for files
```

Obr. 21: Detekcia komunikácie so známymi škodlivými systémami pomocou nástroja RITA

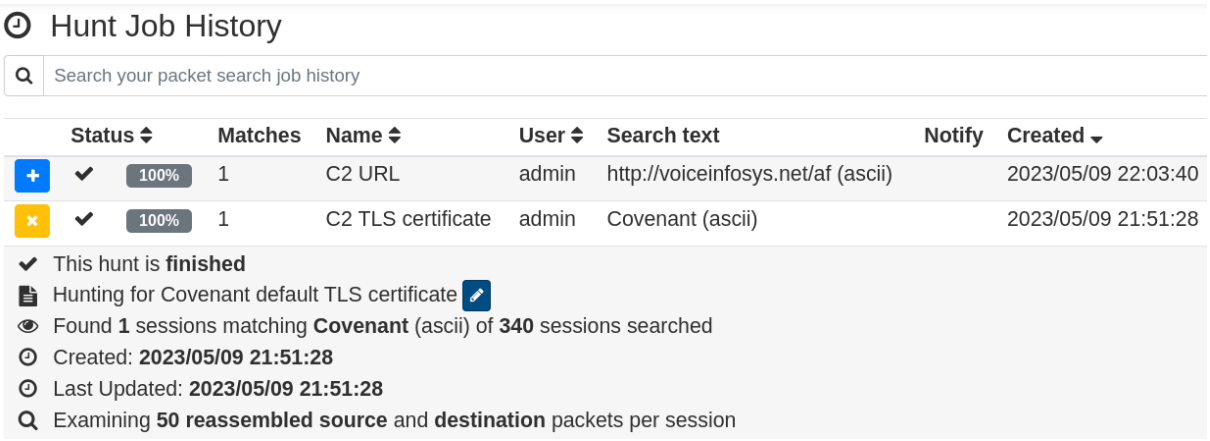
Hoci aplikácia RITA poskytuje rozsiahlejšie štatistické údaje o odchytenej sieťovej prevádzke ako naša aplikácia a zároveň dokáže prostredníctvom výstupov z aplikácie Zeek spracovávať vysokorýchlostné a veľkoobjemové počítačové siete, nepodarilo sa nám pomocou tohto nástroja detegovať rovnaký počet potenciálnych indikátorov C2 komunikácie. Pri použití tej istej testovacej vzorky s našou aplikáciou v prvom popísanom teste sme automatizovaným spôsobom detegovali sieťové spojenia s nadmernou frekvenciou, dlhé sieťové spojenia, sieťovú komunikáciu so známymi IP adresami C2 riadiacich serverov a v neposlednom rade aj sieťovú komunikáciu so známymi doménovými menami C2 riadiacich serverov. Výstupy analýzy rovnakej vzorky pomocou nástroja RITA, ako sme mohli vidieť vyššie, detegovali iba potenciálnu komunikáciu s dvomi rôznymi IP adresami C2 IP riadiacich serverov, pričom náš nástroj detegoval celkovo šesť rôznych IP adries v spojení s C2 komunikáciou.

8.2 Porovnanie detekčných schopností s nástrojom Arkime

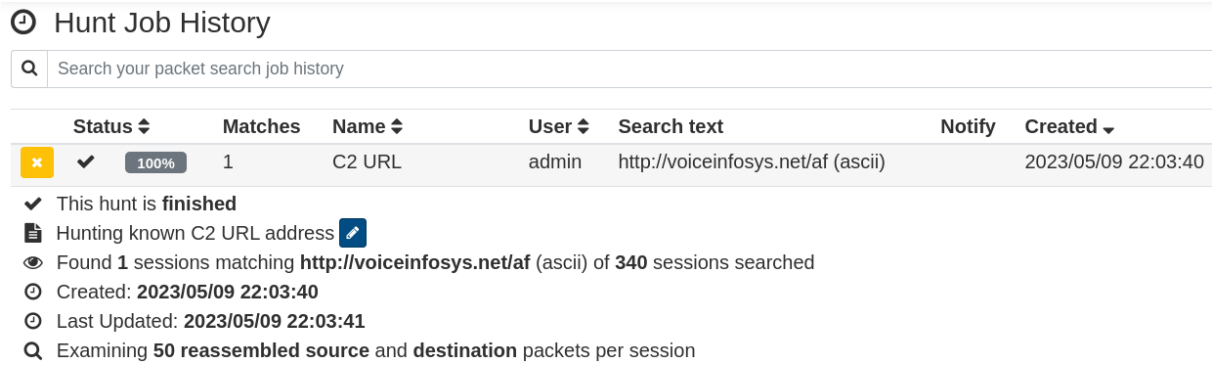
Pre ďalšie porovnanie detekčných schopností C2 komunikácie našej aplikácie s iným riešením sme si vybrali popísaný nástroj s názvom **Arkime**. Po úspešnej inštalácii a konfigurácii sme mohli do tohto nástroja načítať vybraný PCAP súbor obsahujúci odchytenú sieťovú prevádzku. Ako testovaciu vzorku sme použili PCAP súbor *„test_4.pcap“* zo štvrtého testovania.

Nástroj Arkime poskytuje vyhľadávanie v odchytenej sieťovej prevádzke prostredníctvom funkcie s názvom **Hunt**, ktorej hlavnou úlohou je identifikovať a upozorniť používateľa na konkrétnu sieťovú aktivitu, alebo správanie, ktoré môže byť potenciálne škodlivého charakteru. Používateľ má možnosť vytvárať takzvaný ***Hunt Job***, v ktorom zadáva konkrétnu hodnotu, ktorá bude vyhľadávaná v spracovanej odchytenej sieťovej prevádzke, resp. v jej podmnožine, ktorú je možné definovať. Takýto prístup je vhodný v prípadoch, kedy v sieťovej infraštruktúre chceme vyhľadávať konkrétne IoCs.

V našej modelovej situácii sme vytvorili dve vyhľadávania, pričom prvé obsahovalo reťazec *„Covenant“* a druhé vyhľadávanie obsahovalo taktiež reťazec, konkrétne URL adresu *„http://voiceinfosys.net/af“*. Ako môžeme vidieť na obrázku 22 Arkime úspešne našiel hodnotu „Covenant“ v jednej zo sieťových relácií. Detail predmetného spojenia neuvádzame z dôvodu veľkého rozsahu a exportovanie zistení do súboru formátu CSV neobsahovalo úplné údaje o detegovanej aktivite. Na obrázku 23 vidíme, že Arkime v druhom vyhľadávaní detegoval predmetnú URL adresu. Aj v tomto prípade, z dôvodu veľkého rozsahu a neúplného exportovaného CSV súboru, neuvádzame detail príslušného sieťového spojenia.



Obr. 22: Výsledok vyhľadávania C2 TLS certifikátu pomocou nástroja Arkime



Obr. 23: Výsledok vyhľadávania C2 URL adresy pomocou nástroja Arkime

Napriek tomu, že nástroj Arkime poskytuje plnohodnotné webové rozhranie s množstvom analytických funkcionalít, ako sme zistili pri testovaní, tento nástroj má veľmi limitované možnosti pre *automatizovanú* detekciu C2 komunikácie. V neposlednom rade treba uviesť, že aj Arkime je podporovaný iba na vybraných Linuxových distribúciach, konkrétne *Ubuntu*, *CentOS*, *Arch Linux*, *Red Hat Enterprise Linux* a *Amazon Linux*.

8.3 Vyhodnotenie použitia aplikácie C2Detective

Jednou z hlavných výhod našej aplikácie voči testovaným nástrojom je jej jednoduchá konfigurácia, intuitívne používanie poskytovaných funkcionalít prostredníctvom dostupných prepínačov, využitie voľne dostupných threat feedov pre detekčné účely pomocou rozšírenia C2Hunter a v neposlednom rade ***automatizovaný proces detekcie*** potenciálnych indikátorov C2 komunikácie, ktoré sú prehľadne zhrnuté v každej výstupnej správe o výsledkoch príslušnej analýzy. Napriek tomu, že aplikácia C2Detective v čase písania tejto práce nie je prispôsobená na spracovávanie vysokorýchlostných a veľkoobjemových počítačových sietí, jej použitie môže byť aplikované na vybrané systémy, o ktorých máme podozrenie, že boli kompromitované.

Záver

Cieľom tejto práce bolo oboznámiť sa s **Command and Control** komunikáciou v kontexte kybernetickej bezpečnosti a vytvoriť aplikáciu, ktorá implementuje vybrané existujúce techniky a pravidlá ako aj vlastné pravidlá detekcie potenciálnych indikátorov takejto škodlivej sieťovej komunikácie.

V úvode práce sme si pre lepšie porozumenie rozsahu popisovanej problematiky, ktorú rieši naša práca, predstavili **indikátory kompromitácie** a ich kategorizáciu pomocou diagramu s názvom **Pyramid of Pain**. Ďalej sme si priblížili procesy **Cyber Threat Intelligence**, ktorých výstupom sú informácie o kybernetických hrozbách, ktoré môžeme okrem iného použiť aj na detekciu Command and Control komunikácie. Životný cyklus kybernetického útoku, ktorého súčasťou je fáza Command and Control bol prezentovaný prostredníctvom rámcov kybernetickej bezpečnosti s názvom **Cyber Kill Chain** a **MITRE ATT&CK**.

Ďalšia časť našej práce sa venovala analýze vybraných troch voľne dostupných Command and Control frameworkov s otvoreným zdrojovým kódom, **Sliver**, **Merlin** a **Mythic**. Predstavili sme a porovnali techniky, ktoré nám tieto frameworky poskytujú pre účely komunikácie s kompromitovanými systémami a taktiež sme uviedli teoretický prístup k detekcii Command and Control komunikácie.

Vychádzajúc z analytickej časti a definovaných funkcionálnych a nefunkcionálnych požiadaviek sme navrhli a popísali systémovú špecifikáciu našej aplikácie, ktorú sme nazvali **C2Detective**. Celkovo sme implementovali *jedenásť detekčných metód*, ktoré v konečnom dôsledku dokážu zachytiť *štrnásť potenciálnych indikátorov Command and Control komunikácie*. Aplikácia C2Detective bola navrhnutá tak, aby zabezpečovala jednoduchú integráciu nových detekčných techník, napríklad pomocou rozšírení. Túto vlastnosť sme demonštrovali využitím lokálnej databázy Command and Control riadiacich serverov, ktorej údaje agreguje a spracováva nástroj **C2Hunter**. Tento nástroj, ktorý sme vyvíjali nezávisle od našej práce, využíva mimo iné techniku ***fingerprinting*** v spojení s dátami, ktoré poskytuje služba **Shodan**. Prostredníctvom API rozhraní vybraných služieb aplikácia C2Detective umožňuje používateľovi obohatiť detegované indikátory Command and Control komunikácie o informácie, ktoré sú výsledkom procesov Cyber Threat Intelligence. Výstupom našej aplikácie je správa o výsledkoch analýzy vo formáte HTML a PDF. Takéto výstupné správy poskytujú používateľovi detailný prehľad detegovaných indikátorov Command and Control komunikácie a ďalších relevantných informácii.

Implementované techniky detekcie Command and Control komunikácie sme testovali

na vzorkách odchytenej sieťovej komunikácie, ktoré obsahujú škodlivé aktivity spájané s komunikáciou s kompromitovanými systémami. Pri vhodnej konfigurácii aplikácie C2Detective sa nám podarilo demonštrovať detekciu všetkých možných potenciálnych indikátorov Command and Control komunikácie, ktoré dokáže táto aplikácia detegovať. V neposlednom rade sme na testovacích vzorkách porovnali detekčné schopnosti voľne dostupných nástrojov s názvom **RITA** a **Arkime** s našou aplikáciou.

Aplikácia **C2Detective** predstavuje prínos, resp. je alternatívou k malému množstvu bezplatných nástrojov, ktoré majú otvorený zdrojový kód a ich poskytované funkcionality vieme využiť pri *automatizovanej detekcii* Command and Control komunikácie. Medzi hlavné výhody nášho riešenia patrí jednoduchá konfigurácia, intuitívne používanie detekčných metód prostredníctvom prepínačov aplikácie, využitie voľne dostupných threat feedov pre detekčné účely pomocou rozšírenia **C2Hunter** a v neposlednom rade ***automatizovaný proces detekcie*** potenciálnych indikátorov C2 komunikácie, ktoré sú prehľadne zhrnuté v každej výstupnej správe o výsledkoch príslušnej analýzy. Taktiež naša aplikácia poskytuje používateľovi možnosť *obohatiť* detegované potenciálne indikátory C2 komunikácie o Cyber Threat Intelligence informácie z viacerých zdrojov, ktoré môžu byť využité pre ďalšie analytické procesy.

Napriek tomu, že aplikácia C2Detective v čase písania tejto práce nie je prispôsobená na spracovávanie vysokorýchlostných a veľkoobjemových počítačových sietí, jej použitie môže byť efektívne pri zameraní sa na vybrané systémy, o ktorých máme podozrenie, že boli kompromitované.

Zoznam použitej literatúry

1. SAMONAS, Spyridon a COSS, David. The CIA strikes back: Redefining confidentiality, integrity and availability in security. *Journal of Information System Security*. 2014, roč. 10, č. 3.
2. MAVROEIDIS, Vasileios, HOHIMER, Ryan, CASEY, Tim a JESANG, Audun. Threat actor type inference and characterization within cyber threat intelligence. In: *2021 13th International Conference on Cyber Conflict (CyCon)*. IEEE, 2021, s. 327–352.
3. FORTINET. *Command and Control Attacks* [online]. [cit. 2023-05-12]. Dostupné z : `https://www.fortinet.com/resources/cyberglossary/command-and-control-attacks`.
4. MITRE CORPORATION. *TA0011: Command and Control* [online]. [cit. 2023-05-12]. Dostupné z : `https://attack.mitre.org/tactics/TA0011/`.
5. CROWDSTRIKE. *Indicators of Compromise (IOC)* [online]. 2022. [cit. 2023-05-12]. Dostupné z : `https://www.crowdstrike.com/cybersecurity-101/indicators-of-compromise/`.
6. NATIONAL CYBER SECURITY CENTRE (NCSC), Netherlands. *Factsheet Indicators of Compromise* [online]. 2017. [cit. 2023-05-12]. Dostupné z : `https://english.ncsc.nl/publications/factsheets/2019/juni/01/factsheet-indicators-of-compromise`.
7. BIANCO, David J. The Pyramid of Pain [online]. 2013 [cit. 2023-05-12]. Dostupné z : `https://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html`.
8. CROWDSTRIKE. *What is Threat Intelligence?* [online]. 2023. [cit. 2023-05-12]. Dostupné z : `https://www.crowdstrike.com/cybersecurity-101/threat-intelligence/`.
9. CHISMON, David a RUKS, Martyn. Threat intelligence: Collecting, analysing, evaluating. *MWR InfoSecurity Ltd.* 2015.
10. REBEKAH BROWN, Robert M. Lee. *2021 SANS Cyber Threat Intelligence (CTI) Survey* [online]. 2021. [cit. 2023-05-12]. Dostupné z : `https://www.threatq.com/documentation/Survey_CTI-2021_ThreatQuotient.pdf`.

11. CREST. What is Cyber Threat Intelligence and how is it used? [online]. 2015 [cit. 2023-05-12]. Dostupné z : https://www.crest-approved.org/wp-content/uploads/2022/04/CREST-Cyber-Threat-Intelligence.pdf.

12. NATIONAL CYBER SECURITY CENTRE (NCSC), United Kingdom. *Cyber Threat Intelligence in Government: A Guide for Decision Makers & Analysts* [online]. 2019. [cit. 2023-05-12]. Dostupné z : https://hodigital.blog.gov.uk/wp-content/uploads/sites/161/2020/03/Cyber-Threat-Intelligence-A-Guide-For-Decision-Makers-and-Analysts-v2.0.pdf.

13. CROWDSTRIKE. *What Is The Cyber Kill Chain? Process & Model* [online]. 2022. [cit. 2023-05-12]. Dostupné z : https://www.crowdstrike.com/cybersecurity-101/cyber-kill-chain/.

14. HUTCHINS, Eric M, CLOPPERT, Michael J, AMIN, Rohan M, et al. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains [online]. 2011 [cit. 2023-05-12]. Dostupné z : https://www.lockheedmartin.com/content/dam/lockheed-martin/rms/documents/cyber/LM-White-Paper-Intel-Driven-Defense.pdf.

15. STROM, Blake E, APPLEBAUM, Andy, MILLER, Doug P, NICKELS, Kathryn C, PENNINGTON, Adam G a THOMAS, Cody B. Mitre ATT&CK: Design and philosophy. In: *Technical report*. The MITRE Corporation, 2018.

16. CROWDSTRIKE. *MITRE ATT&CK Framework* [online]. [cit. 2023-05-12]. Dostupné z : https://www.crowdstrike.com/cybersecurity-101/mitre-attack-framework/.

17. TRELLIX. *What Is the MITRE ATT&CK Framework?* [online]. [cit. 2023-05-12]. Dostupné z : https://www.trellix.com/en-us/security-awareness/cybersecurity/what-is-mitre-attack-framework.html.

18. MITRE CORPORATION. *MITRE ATT&CK Enterprise tactics* [online]. [cit. 2023-05-12]. Dostupné z : https://attack.mitre.org/tactics/enterprise/.

19. PALO ALTO NETWORK. *What is the MITRE ATT&CK Framework?* [online]. [cit. 2023-05-12]. Dostupné z : https://www.paloaltonetworks.com/cyberpedia/what-is-mitre-attack-framework.

20. PALO ALTO NETWORKS. *What is a Command and Control Attack?* [online]. [cit. 2023-05-12]. Dostupné z : https://www.paloaltonetworks.com/cyberpedia/command-and-control-explained.

21. GRIMMICK, Robert. *What is C2? Command and Control Infrastructure Explained* [online]. 2022. [cit. 2023-05-12]. Dostupné z : https://www.varonis.com/blog/what-is-c2.

22. FORTRA, LLC. *Cobalt Strike* [online]. [cit. 2023-05-12]. Dostupné z : https://www.cobaltstrike.com/. [Softvér].

23. RAPID7. *Metasploit Framework* [online]. [cit. 2023-05-12]. Dostupné z : https://github.com/rapid7/metasploit-framework. [Softvér].

24. BC SECURITY. *Empire* [online]. [cit. 2023-05-12]. Dostupné z : https://github.com/BC-SECURITY/Empire. [Softvér].

25. BISHOP FOX. *The Red Team Beginner's Guide to Sliver* [online]. [cit. 2023-05-12]. Dostupné z : https://github.com/BishopFox/sliver/wiki/Beginner's-Guide.

26. BISHOP FOX. *Sliver* [online]. [cit. 2023-05-12]. Dostupné z : https://github.com/BishopFox/sliver. [Softvér].

27. LAKSHMANAN, Ravie. Threat Actors Turn to Sliver as Open Source Alternative to Popular C2 Frameworks [online]. 2023 [cit. 2023-05-12]. Dostupné z : https://thehackernews.com/2023/01/threat-actors-turn-to-sliver-as-open.html.

28. CYBEREASON GLOBAL SOC AND INCIDENT RESPONSE TEAM. Sliver C2 Leveraged by Many Threat Actors [online]. 2023 [cit. 2023-05-12]. Dostupné z : https://www.cybereason.com/blog/sliver-c2-leveraged-by-many-threat-actors.

29. TUYL, Russel Van. *Merlin* [online]. [cit. 2023-05-12]. Dostupné z : https://github.com/Ne0nd0g/merlin. [Softvér].

30. TUYL, Russel Van. *Merlin Command and Control framework* [online]. [cit. 2023-05-12]. Dostupné z : https://merlin-c2.readthedocs.io/en/latest/index.html.

31. ALTHOUSE, John. *TLS Fingerprinting with JA3 and JA3S* [online]. [cit. 2023-05-12]. Dostupné z : https://engineering.salesforce.com/tls-fingerprinting-with-ja3-and-ja3s-247362855967/.

32. COBB, Ryan. *Covenant* [online]. [cit. 2023-05-12]. Dostupné z : https://github.com/cobbr/Covenant. [Softvér].

33. COBB, Ryan. *Covenant Wiki* [online]. [cit. 2023-05-12]. Dostupné z : https://github.com/cobbr/Covenant/wiki.

34. ASHER-DOTAN, Lital. *What is Domain Generation Algorithm: 8 Real World DGA Variants* [online]. 2016. [cit. 2023-05-12]. Dostupné z : https://www.cybereason.com/blog/what-are-domain-generation-algorithms-dga.

35. ORCHILLES, Jorge. *C2 Matrix* [online]. [cit. 2023-05-12]. Dostupné z : https://howto.thec2matrix.com/.

36. MITRE CORPORATION. *MITRE D3FEND Framework* [online]. [cit. 2023-05-12]. Dostupné z : https://d3fend.mitre.org/.

37. *Arkime* [online]. [cit. 2023-05-12]. Dostupné z : https://github.com/arkime/arkime. [Softvér].

38. *Zeek* [online]. [cit. 2023-05-12]. Dostupné z : https://github.com/zeek/zeek. [Softvér].

39. ACTIVE COUNTERMEASURES. *Real Intelligence Threat Analytics (RITA)* [online]. [cit. 2023-05-12]. Dostupné z : https://github.com/activecm/rita. [Softvér].

40. OPEN INFORMATION SECURITY FOUNDATION. *Suricata* [online]. [cit. 2023-05-12]. Dostupné z : https://github.com/OISF/suricata. [Softvér].

41. ROESCH, Marty. *Snort* [online]. [cit. 2023-05-12]. Dostupné z : https://www.snort.org/. [Softvér].

42. *The Python Package Index* [online]. [cit. 2023-05-12]. Dostupné z : https://pypi.org/. [Softvér].

43. KUBEČKA, Martin. *C2Detective* [online]. [cit. 2023-05-12]. Dostupné z : https://github.com/martinkubecka/C2Detective. [Softvér].

44. SECDEV. *Scapy* [online]. [cit. 2023-05-12]. Dostupné z : https://github.com/secdev/scapy. [Softvér].

45. COMBS, Gerald. *Tshark Manual Page* [online]. [cit. 2023-05-12]. Dostupné z : https://www.wireshark.org/docs/man-pages/tshark.html. [Softvér].

46. SANDERS, Chris a SMITH, Jason. *Applied network security monitoring: collection, detection, and analysis*. Elsevier, 2013.

47. BEJTLICH, Richard. *The practice of network security monitoring: understanding incident detection and response*. No Starch Press, 2013.

48. SALESFORCE. *JA3* [online]. [cit. 2023-05-12]. Dostupné z : https://github.com/salesforce/ja3. [Softvér].

49. ABUSEIPDB, LLC. *AbuseIPDB* [online]. [cit. 2023-05-12]. Dostupné z : https://www.abuseipdb.com/.

50. ALIENVAULT, INC. *AlienVault* [online]. [cit. 2023-05-12]. Dostupné z : https://otx.alienvault.com/.

51. SHODAN, LLC. *Shodan* [online]. [cit. 2023-05-12]. Dostupné z : https://www.shodan.io/.

52. ABUSE.CH. *ThreatFox* [online]. [cit. 2023-05-12]. Dostupné z : https://threatfox.abuse.ch/.

53. ABUSE.CH. *URLhaus* [online]. [cit. 2023-05-12]. Dostupné z : https://urlhaus.abuse.ch/.

54. CHRONICLE SECURITY IRELAND LIMITED. *VirusTotal* [online]. [cit. 2023-05-12]. Dostupné z : https://www.virustotal.com/.

55. ELASTIC NV. *Kibana* [online]. [cit. 2023-05-12]. Dostupné z : https://www.elastic.co/kibana/. [Softvér].

56. COMMUNITY FOR OPEN SOURCE SECURITY AUTOMATION SOFTWARE. *DGA Detective* [online]. 2021. [cit. 2023-05-12]. Dostupné z : https://cossas-project.org/portfolio/dgad/.

57. COMMUNITY FOR OPEN SOURCE SECURITY AUTOMATION SOFTWARE. *DGA Detective* [online]. [cit. 2023-05-12]. Dostupné z : https://github.com/COSSAS/dgad. [Softvér].

58. FARNHAM, Greg a ATLASIS, Antonios. Detecting DNS tunneling. *SANS Institute InfoSec Reading Room*. 2013, roč. 9, s. 1–32.

59. EMERGING THREATS RESEARCH TEAM. *Emerging Threats JA3 Ruleset* [online]. [cit. 2023-05-12]. Dostupné z : https://rules.emergingthreats.net/open/suricata-5.0/rules/emerging-ja3.rules.

60. STRAND, John. Detecting Malware Beacons With Zeek and RITA [online]. 2020 [cit. 2023-05-12]. Dostupné z : https://www.blackhillsinfosec.com/detecting-malware-beacons-with-zeek-and-rita/.

61. GARCIA, Sebastian. Modelling the network behaviour of malware to block malicious patterns. the stratosphere project: a behavioural ips. *Virus Bulletin*. 2015, s. 1–8.

62. GOURLEY, David, TOTTY, Brian, SAYER, Marjorie, AGGARWAL, Anshu a REDDY, Sailu. *HTTP: The Definitive Guide*. O'Reilly Media, Inc., 2002.

63. PAL, Debashis. How to: Detect and prevent common data exfiltration attacks [online]. 2022 [cit. 2023-05-12]. Dostupné z : https://blog.apnic.net/2022/03/31/how-to-detect-and-prevent-common-data-exfiltration-attacks/.

64. MITRE CORPORATION. *T1090.003: Proxy - Multi-hop Proxy* [online]. [cit. 2023-05-12]. Dostupné z : https://attack.mitre.org/techniques/T1090/003/.

65. AUSTIN, Daniel. *Tor Node List* [online]. [cit. 2023-05-12]. Dostupné z : https://www.dan.me.uk/tornodes.

66. MALWAREBYTES. *Cryptojacking – What is it?* [online]. [cit. 2023-05-12]. Dostupné z : https://www.malwarebytes.com/cryptojacking.

67. THE BLOCK LIST PROJECT. *The Block List Project - Crypto List* [online]. [cit. 2023-05-12]. Dostupné z : https://blocklistproject.github.io/Lists/alt-version/crypto-nl.txt.

68. THOMAS, Will. Detecting and Fingerprinting Infostealer Malware-as-a-Service platforms [online]. 2022 [cit. 2023-05-12]. Dostupné z : https://blog.bushidotoken.net/2022/11/detecting-and-fingerprinting.html.

69. THOMAS, Will. *Adversary Infrastructure on Shodan* [online]. [cit. 2023-05-12]. Dostupné z : https://github.com/BushidoUK/OSINT-SearchOperators/blob/main/ShodanAdversaryInfa.md.

70. KUBEČKA, Martin. *C2Hunter* [online]. [cit. 2023-05-12]. Dostupné z : https://github.com/martinkubecka/C2Hunter. [Softvér].

71. ABUSE.CH. *Feodo Tracker* [online]. [cit. 2023-05-12]. Dostupné z : https://feodotracker.abuse.ch/.

72. DUNCAN, Brad. *IcedID (Bokbot) with backconnect and VNC and Cobalt Strike* [online]. 2023. [cit. 2023-05-12]. Dostupné z : https://www.malware-traffic-analysis.net/2023/01/16/index2.html.

73. *Zeus Malware DGA Domain Names* [online]. 2017. [cit. 2023-05-12]. Dostupné z : https://github.com/thesraid/capper/blob/master/pcaps/zeus-dga.pcap.

74. CARTIER, Hannah. *Malware of the Day – dnscat2 DNS Tunneling* [online]. 2021. [cit. 2023-05-12]. Dostupné z : https://www.activecountermeasures.com/malware-of-the-day-dnscat2-dns-tunneling/.

75. DUNCAN, Brad. *IcedID (BokBot) infection with Cobalt Strike* [online]. 2022. [cit. 2023-05-12]. Dostupné z : https://www.malware-traffic-analysis.net/2022/09/23/index.html.

Prílohy

A Štruktúra projektu

- Zdrojový kód aplikácie **C2Detective** v rovnomennom priečinku.
- Zdrojový kód aplikácie **C2Hunter** v rovnomennom priečinku.
- Testovacie PCAP súbory v priečinku s názvom ***test_samples***.
- Správy o výsledkoch analýzy z testovania v priečinku s názvom ***test_reports***.

B Používateľská príručka

Táto používateľská príručka poskytuje prehľad o konfigurácii a správnom a efektívnom používaní aplikácie **C2Detective**. Okrem toho uvádza informácie o nástroji **C2Hunter**, ktorý je možné použiť ako rozšírenie aplikácie C2Detective.

B.1 Konfigurácia a použitie aplikácie C2Detective

Pred spustením aplikácie **C2Detective** musí byť na danej pracovnej stanici nainštalovaný **Python** interpreter, pričom najnižšia podporovaná verzia je **3.8** a aktuálne najvyššia podporovaná verzia, ktorá je zároveň aj odporúčaná je verzia **3.10**. Taktiež je potrebné mať nainštalovaný nástroj **Tshark** [45]. Repozitár, ktorý obsahuje zdrojový kód aplikácie je verejne dostupný na platforme Github [43]. Na úvod odporúčame vytvoriť izolované virtuálne Python prostredie, ktoré nám zabezpečí, aby nevznikali konflikty vo verziách využívaných modulov, najmä pri práci na viacerých projektoch, ktoré využívajú rovnaké moduly s rôznymi verziami. Nasledujúcimi príkazmi nainštalujeme modul s názvom ***virtualenv***, následne v priečinku kde sa nachádza zdrojový kód aplikácie C2Detective vytvoríme izolované virtuálne prostredie pre Python verziu 3.10, ktoré pomenujeme *venv*. V neposlednom rade aktivujeme novo vytvorené venv prostredie, v ktorom vykonávame všetky ďalšie kroky.

```
$ pip install virtualenv
$ virtualenv --python=python3.10 venv
$ source venv/bin/activate
```

Pred prvým použitím je taktiež potrebné nainštalovať požadované moduly, ktorých zoznam sa nachádza v súbore ***requirements.txt***. Pre automatizovanú inštaláciu týchto modulov môžeme použiť nasledujúci príkaz.

```
$ pip install -r requirements.txt
```

V priečinku, kde sa nachádza zdrojový kód aplikácie C2Detective, resp. v podpriečinku **config** je potrebné vytvoriť konfiguračný súbor s názvom ***config.yml*** podľa vzorového súboru *example.yml*, ktorý sa už nachádza v uvedenom podpriečinku. V prípade, že používateľ chce použiť funkcionalitu obohacovanie detegovaných indikátorov kompromitácie, do konfiguračného súboru ***config/config.yml*** je potrebné doplniť API kľúče pre uvedené služby. V prípade, že sa v priečinku config nenachádza súbor s názvom ***domain_whitelist.txt***, je

potrebné v uvedenom priečinku tento súbor vytvoriť. Následne je možné pridávať doménové mená, jeden vstup pre jeden riadok súboru, ktoré nebudú posudzované pri detekcii techniky DNS Tunneling.

Použitý modul **Scapy** vyžaduje pri odchytávaní paketov privilegované oprávnenia. Ak používateľ chce využiť implementovanú funkcionalitu odchytávania paketov, je potrebné pridať **CAP_NET_RAW** a **CAP_NET_ADMIN** atribúty zvolenému Python binárnemu súboru (napríklad verzii 3.10). Nasledujúcimi príkazmi v prvom kroku overíme cestu ku spomenutému binárnemu súboru, následne priradíme vyžadované atribúty a v neposlednom rade overíme, či boli tieto atribúty správne priradené.

```
$ which python3.10
$ sudo setcap 'CAP_NET_RAW+eip CAP_NET_ADMIN+eip' /usr/bin/python3.10
$ getcap /usr/bin/python3.10
```

Pred prvým použitím aplikácie je vhodné sa zoznámiť s dostupnými prepínačmi. Úplný prehľad týchto prepínačov si môžeme zobraziť pomocou nasledovného príkazu.

```
$ python3.10 c2detective.py --help
```

Aplikácia C2Detective vyžaduje na vstupe PCAP súbor, ktorého cestu špecifikujeme pomocou prepínača *-i*, resp. *--input*, alebo je možné pred spustením analytickej časti programu odchytiť sieťovú prevádzku pomocou prepínača *-p*, resp. *--packet-capture*. Špecifické nastavenia odchytávania paketov, ako sieťové rozhranie, filter, doba odchytávania a názov výstupného PCAP súboru, je potrebné špecifikovať v konfiguračnom súboru ***config/config.yml***, v objekte s názvom ***sniffing***.

Pri prvom spustení apikácie je potrebné stiahnuť požadované súbory pomocou dostupných skriptov, ktoré pripravia do preddefinovanej štruktúry zoznam Tor uzlov, doménových mien, ktoré sú spájané s kryptomenami a voľne dostupné JA3 pravidlá. Pri opätovných spusteniach program deteguje, či nie je potrebné dané zoznamy aktualizovať, o čom je používateľ informovaný po spustení programu v jeho výpisoch. Nasledujúci príkaz aktualizuje všetky spomenuté zoznamy pred procesom spracovania vybraného vstupného súboru (samples/test.pcap).

```
$ python3.10 c2detective.py -input samples/test.pcap -utn -ucd -ujr
```

Detekciu DGA doménových mien je potrebné povoliť prepínačom *-d*, resp. *--dga* a obohacovanie detegovaných indikátorov kompromitácie sa povoľuje prepínačom *-e*,

resp. *--enrich*. Ako bolo spomenuté v úvode tejto príručky, do konfiguračného súboru ***config/config.yml*** je potrebné doplniť API kľúče pre jednotlivé služby, ktoré sú využívané pri obohacovaní dát. Pomocou prepínaču *-g*, resp. *--plugins* povoľujeme využitie rozšírení, ktoré je taktiež potrebné definovať v konfiguračnom súbore ***config/config.yml***, konkrétne v objekte ***plugins***. Pri použití rozšírenia C2Hunter definujeme úplnú cestu ku jeho databázovému súboru. Tu je potrebné uviesť, že použitie uvedeného rozšírenia si vyžaduje jeho spustenie pred využitím v aplikácii C2Detective. Konfiguráciu a použitie dostupného rozšírenia C2Hunter popisujeme v ďalšej časti tejto príručky.

Aplikácia C2Detective taktiež poskytuje zobrazenie rôznych informačných a štatistických údajov v rozhraní príkazového riadku pomocou prepínača *-s*, resp. *--statistics*. V prípade, že používateľ chce zapísať všetky vybrané extrahované údaje do výstupného súboru formátu JSON, je potrebné použiť prepínač *-w*, resp. *--write-extracted*.

Využívanie viacerých konfiguračných súborov umožňuje pokročilým používateľom zefektívňovať ich prácu pri automatizovaní analytických procesov pomocou nástroja C2Detective. Cestu ku vlastnému konfiguračnému súboru špecifikujeme pomocou prepínača *-c*, resp. *--config*. V prípade, že si používateľ neželá, aby boli výstupné súbory zapisované do preddefinovaného priečinku s názvom **reports**, pomocou prepínača *-o*, resp. *--output* a definovanej cesty ku zvolenému výstupnému priečinku, zmení ich výslednú lokáciu.

Nižšie uvádzame príklad spustenia nástroja C2Detective, ktorý spracuje súbor definovaný prepínačom *-i*, na základe prepínača *-w* zapíše vybrané extrahované dáta do samostatného súboru a prepínačom *-d* sa aktivuje detekcia DGA doménových mien.

```
$ python3.10 c2detective.py -w -d -i samples/zeus-dga.pcap
```

Po ukončení analytickej časti programu sú výsledky tohto procesu zapísané do výstupnej správy formátu HTML a PDF. Tento súbor sa v základnej konfigurácii nachádza v priečinku s názvom **reports** a obsahuje detailný prehľad detegovaných indikátorov C2 komunikácie, ako aj všeobecné a štatistické údaje o analyzovanom PCAP súbore, resp. o odchytených dátach v počítačovej sieti. V tomto priečinku taktiež nájdeme spracované detegované indikátory C2 komunikácie v súbore formátu JSON a v prípade, že bol použitý prepínač *-w*, resp. *--write-extracted*, v predmetnom priečinku nájdeme aj extrahované údaje zo vstupu v súbore formátu JSON.

B.2 Využitie rozšírenia C2Hunter

Konfigurácia nástroja **C2Hunter** pred prvým spustením je podobná ako tá pre aplikáciu C2Detective. Najnižšia podporovaná verzia Python interpretera je **3.6**, pričom v

čase písania tejto príručky najvyššia podporovaná verzia nie je nijako limitovaná. Repozitár, ktorý obsahuje zdrojový kód tohto nástroja je taktiež verejne dostupný na platforme Github [70]. Aj tu sa odporúča využiť izolované virtuálne Python prostredie pomocou príkazov z predchádzajúcej časti tejto príručky.

Pred prvým použitím je taktiež potrebné nainštalovať požadované moduly, ktorých zoznam sa nachádza v súbore ***requirements.txt***. Nasledujúci príkaz môžeme použiť pre automatizovanú inštaláciu týchto modulov.

```
$ pip install -r requirements.txt
```

V priečinku, kde sa nachádza zdrojový kód aplikácie C2Hunter, resp. v podpriečinku **config** je potrebné vytvoriť konfiguračný súbor s názvom ***config.yml*** podľa vzorového súboru *example.yml*, ktorý sa už nachádza v uvedenom podpriečinku. V prípade, že používateľ chce použiť funkcionalitu, ktorá na základe predvolenej konfigurácie vybraných C2 frameworkov a ich vyhľadávania v dátach, ktoré poskytuje služba **Shodan** agreguje potencionálne C2 servery, je potrebné do uvedeného konfiguračného súboru doplniť API kľúč pre spomínanú službu Shodan. Na tomto mieste je potrebné uviesť, že API plán pre službu Shodan s názvom *Freelancer* nemusí byť postačujúci.

Pri spúšťaní nástroja C2Hunter je možné zakázať agregáciu dát z jednotlivých zdrojov, a to konkrétne pre službu **Shodan** prepínačom *-ds*, reps. *--disable-shodan*, pre službu **Feodo Tracker** prepínačom *-df*, resp. *--disable-feodotracker*, pre službu **URLhaus** prepínačom *-du*, resp. *--disable-urlhaus* a v neposlednom rade pre službu **ThreatFox** prepínačom *-dt*, resp. *--disable-threatfox*. Používateľ má taktiež možnosť zakázať zálohovanie výstupných súborov, ktoré sa vykonáva na záver behu programu, prostredníctvom prepínača *-db*, resp. *--disable-backup*.

V prípade, že to agregované dáta umožňujú, používateľ si dokáže odfiltrovať aktívne systémy pomocou prepínača *-s*, resp. *--search-country-code* na základe dvojpísmenového kódového označenia krajiny (napríklad „SK“), ktoré definuje v konfiguračnom súbore. V prípade, že sú detegované nejaké aktívne systémy v danej geolokácii, tieto systémy sú zapísané do príslušných výstupných súborov. Aktívne systémy je taktiež možné zobraziť v rozhraní príkazového riadku počas behu programu prostredníctvom prepínača *-p*, resp. *--print-active*.

Aj nástroj C2Hunter poskytuje možnosť načítať vlastný konfiguračný súbor špecifikáciou jeho cesty pri použití prepínača *-c*, resp. *--config*. Zmeniť priečinok, do ktorého sa zapisujú výstupné súbory môže používateľ pomocou prepínača *-o*, resp. *--output* a definíciou cesty ku novému výstupnému priečinku.

Nižšie uvádzame príklad spustenia nástroja C2Hunter, ktorý spracuje dáta zo všetkých dostupných zdrojov, predmetné dáta uloží do lokálnej SQLite databázy a na záver zobrazí aktívne systémy, ktoré majú rovnakú geolokáciu ako kód krajiny, ktorý sa nachádza v konfiguračnom súbore.

```
$ python c2hunter.py --search-country-code --print-active
```

Po zozbieraní dostupných zdrojov je možné využiť nástroj C2Hunter ako rozšírenie aplikácie C2Detective. Nemôžeme zabudnúť, že je potrebné do konfiguračného súboru našej aplikácie C2Detective doplniť systémovú cestu k databázovému súboru nástroja C2Hunter.

Uvedený príklad nižšie spracuje súbor definovaný prepínačom *-i* a na základe prepínača *-g* aktivuje nakonfigurované rozšírenie C2Hunter, čím rozšírime detekčné schopnosti aplikácie C2Detective.

```
$ python3.10 c2detective.py -i samples/icedid.pcap -g
```