UNIVERZITA PAVLA JOZEFA ŠAFÁRIKA
PRÍRODOVEDECKÁ FAKULTA

IDENTIFIKÁCIA TYPOV ÚTOČNÍKOV POMOCOU ÚDAJOV Z HONEYPOTOV

2017 **Bc. Lenka KLEINOVÁ**

UNIVERZITA PAVLA JOZEFA ŠAFÁRIKA
PRÍRODOVEDECKÁ FAKULTA

IDENTIFIKÁCIA TYPOV ÚTOČNÍKOV POMOCOU ÚDAJOV Z HONEYPOTOV

DIPLOMOVÁ PRÁCA

Študijný program:	Informatika
Pracovisko (katedra/ústav):	Ústav informatiky
Vedúci diplomovej práce:	RNDr. JUDr. Pavol Sokol, PhD.

Košice 2017

Bc. Lenka KLEINOVÁ

Univerzita P. J. Šafárika v Košiciach
Prírodovedecká fakulta

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta:	Bc. Lenka Kleinová
Študijný program:	Informatika (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor:	9.2.1. informatika
Typ záverečnej práce:	Diplomová práca
Jazyk záverečnej práce:	slovenský
Sekundárny jazyk:	anglický

Názov: Identifikácia typov útočníkov pomocou údajov z honeypotov

Názov EN: Identifying the types of attackers using data from honeypots

Cieľ:
(1) Analyzovať možnosti identifikácie útočníkov pomocou modelov útočníkov
(2) Porovnať aktuálne prístupy k identifikácii typov útočníkov
(3) Navrhnúť modely útočníkov zohľadňujúc údaje získané honeypotmi
(4) Navrhnúť a implementovať rozhodovací algoritmus pre určenie typu útočníka v údajoch získaných pomocou honeypotov

Literatúra:
[1] SHOSTAK, Adam. Threat modeling: Designing for security. John Wiley & Sons, 2014.
[2] JOSHI, R. C., and ANJALI Sardana, eds. Honeypots: A New Paradigm to Information Security. CRC Press, 2011.
[3] EKELHART, Andreas, et al. Integrating attacker behavior in IT security analysis: a discrete-event simulation approach. Information Technology and Management, 2015, 16.3: 221-233.
[4] LENIN, Aleksandr; WILLEMSON, Jan; SARI, Dyan Permata. Attacker profiling in quantitative security assessment based on attack trees. In: Secure IT Systems. Springer International Publishing, 2014. p. 199-212.
[5] KRAUTSEVICH, Leanid; MARTINELLI, Fabio; YAUTSIUKHIN, Artsiom. Towards modelling adaptive attacker's behaviour. In: Foundations and Practice of Security. Springer Berlin Heidelberg, 2013. p. 357-364.

Vedúci:	RNDr. JUDr. Pavol Sokol, PhD.
Oponent:	RNDr. Rastislav Krivoš-Belluš, PhD.
Ústav :	ÚINF - Ústav informatiky
Riaditeľ ústavu:	prof. RNDr. Viliam Geffert, DrSc.

Dátum schválenia: 10.04.2017

prof. RNDr. Viliam Geffert, DrSc.
riaditeľ ústavu

Univerzita Pavla Jozefa Šafárika v Košiciach
Prírodovedecká fakulta
Ústav informatiky

Poďakovanie

Týmto by som sa rada poďakovala vedúcemu svojej práce RNDr. JUDr. Pavlovi Sokolovi, PhD. za pomoc, trpezlivosť a cenné rady pri vypracovaní tejto práce. Takisto moja vďaka patrí RNDr. Tomášovi Horváthovi, PhD., ktorý prispel svojimi skúsenosťami a vedomosťami najmä z oblasti dolovania v údajoch. Rovnako sa chcem poďakovať za hodnotné vstupy od organizácie CESNET, najmä Ing. Pavlovi Káchovi.

Vyhlásenie

Vyhlasujem, že som túto diplomovú prácu vypracovala samostatne na základe vedomostí získaných štúdiom a s pomocou uvedenej literatúry.

Bc. Lenka Kleinová

Abstrakt v štátnom jazyku

Cieľom tejto záverečnej práce je identifikácia aktérov kybernetických útokov - útočníkov. Keďže čím viac organizácii je závislých na rôznych počítačových systémoch, je potrebné zabezpečiť ich pred neautorizovaným prístupom a následne pred ich zneužitím. Na dosiahnutie tohto cieľa je v prvom rade potrebné poznať postupy útočníkov pri útokoch, aké nástroje využívajú, aké sú ich motivácie či znalosti o systéme, na ktorý útočia. Nástroj, ktorý je možné použiť na získavanie informácií o útokoch, resp. útočníkoch je honeypot. Ide o systém, ktorý sa navonok javí ako skutočný produkčný systém, ale jeho úlohou je sledovať a zaznamenať postup útočníka pre následnú analýzu. V práci zaraďujeme útočníkov do skupín s využitím zhlukovacieho algoritmu. Každá takáto skupina je reprezentovaná „ideálnym“ útočníkom, resp. centroidom zhluku, podľa ktorého následne analyzujeme získané skupiny. Výsledkom našej práce je rozbor použitia metódy k-means zhlukovania pre určenie typov útočníkov.

Kľúčové slová: útočník, útok, honeypot, zhlukovanie, k-means.

Abstract

In our diploma thesis, we focus on identification of actors of cyber-attacks - attackers. Currently, more and more organizations are dependent on different computer systems and that's why it's necessary to prevent unauthorized access and misuse of these systems. To achieve this goal, we need to know attacker's steps, tools he uses, his motivations or knowledge about the system. Tool for gaining this information is a honeypot. It lures attackers, monitors their activities and stores this information for further analysis. In the thesis, we categorize attackers into groups using clustering algorithm. Every group is represented by an „ideal" attacker – centroid of a cluster. According to this centroid, we analyze the groups. The result of our thesis is the analysis of the use of k-means clustering for identifying the attackers.

Key words: attacker, attack, honeypot, clustering, k-means.

Obsah

Zoznam skratiek a značiek

DDoS Distributed Denial of Service, distribuované odopretie služby

DNS Domain name service, systém doménových mien

HTTP Hypertext transfer protocol, hypertextový prenosový protokol

ICMP Internet Control Message Protocol, informačný sieťový protokol

IDS Intrusion Detection System, systém pre odhalenie prieniku

IP Internet Protocol, základný protokol pracujúci na sieťovej vrstve

SSH Secure shell, zabezpečený komunikačný protokol

TCP Transmission Control Protocol, protokol riadenia prenosu, jeden zo základných sieťových protokolov

UDP User Datagram Protocol, datagramový protokol, nespoľahlivý sieťový protokol

UML Unified Modeling Lanfuage, jednotný jazyk pre modelovanie

TLS Transport Layer Security, protokol na zabezpečenie transportnej vrstvy

FTP File Transfer Protocol, protokol na prenos súborov

SCADA Supervisory Control And Data Acquisition, systémy pre dispečerské riadenie a zber údajov

Úvod

Kybernetické útoky sa vyskytujú čoraz častejšie a sú čím ďalej sofistikovanejšie. S postupom času pribúdajú nové typy útokov, čím sa práve táto oblasť výskumu stáva veľmi dôležitou. Poznatky o kybernetických útočníkoch a ich kategorizácia výrazne pomáha pri ochrane zdrojov organizácií. Štúdium kybernetických útočníkov začalo už okolo roku 1980, kedy sa začali rozširovať osobné počítače. Slovo „hacker“ bolo v tom čase spájané s človekom schopným programovať a narábať s operačným systémom. Až niekoľko rokov neskôr sa začalo spájať so škodlivou činnosťou a v súčasnosti sa preferuje pojem „cracker“ na označenie kybernetického útočníka.

Realitou v súčasnosti je fakt, že služby kybernetickej kriminality sú lacné. Ceny za krádež dát a za iné služby podobného charakteru stále klesajú. Práve útočníci sú v pozícii, v ktorej majú na nízke ceny najväčší vplyv, čo zvyšuje riziko a počet útokov. Jednotlivci majú motiváciu vykonávať útoky najmä kvôli nedostatočnému zabezpečeniu systémov, vďaka ktorému môžu aj nováčikovia vykonať úspešný SQL injection útok alebo phishing útok. Dark web a dark net je využívaný ako akási skrýša pre útočníkov, ktorí ho vedia zneužiť vo svoj prospech. Prístup k nim vyžaduje určité technické znalosti a na jeho zabezpečenie pred neželanými „návštevníkmi“ boli implementované rôzne technické prekážky.

Útočníkov v kyberpriestore je veľmi náročné identifikovať a nájsť. Čiže len veľmi málo z nich si za svoje činy odpykáva trest. To je ďalší dôvod rozmáhania sa práve takéhoto typu útokov. Zaradenie útočníka do skupiny na základe zvoleného modelu útočníka so sebou prináša prídavnú informáciu o danom útočníkovi a môže pomôcť pri včasnej detekcii útoku a zabráneniu ďalšieho postupu útočníka. Vďaka informácii o zaradení útočníka do skupiny je možné predpovedať jeho ďalšiu aktivitu a podniknúť tak kroky zaisťujúce čo najmenšie dôsledky tejto škodlivej činnosti.

V našej práci sa na otázku kybernetickej bezpečnosti pozeráme práve z hľadiska útočníka. Za každým útokom stojí útočník, ktorého charakterizuje jeho správanie v napadnutom systéme. Jeho správanie je do určitej miery ovplyvnené jeho motiváciou, schopnosťami, resp. skúsenosťami, ktoré má s vykonávaním útokov, a znalosťami o systéme, na ktorý útočí. Väčšina existujúcich modelov kybernetických útočníkov berie do úvahy práve tieto atribúty pri kategorizácii útočníkov.

Cieľom našej práce je v prvom rade porovnať existujúce prístupy ku kategorizácii kybernetických útočníkov a analyzovať možnosti využitia rôznych modelov útočníkov pri identifikácii typov útočníkov. Následne je naším cieľom navrhnúť vlastný model útočníka, navrhnúť spôsob rozhodovania, do akej kategórie bude útočník zaradený, a tento postup implementovať. V práci sme pre dosiahnutie tohto cieľa zvolili metódu zhlukovania, pomocou ktorej útočníkov zaraďujeme do jednotlivých skupín.

Keďže ako nástroj na získavanie údajov o útočníkoch využívame honeypoty, tak prvá kapitola definuje pojem honeypot, honeynet a rozoberá výhody, ale aj riziká, ktoré so sebou honeypoty prinášajú. Súčasťou je podkapitola, ktorá popisuje jednotlivé existujúce typy honeypotov použité pri získavaní údajov o útokoch, ktoré máme k dispozícii od CESNETu.

V druhej kapitole sa zameriavame na klasifikáciu útočníkov. Do úvahy berieme najmä tri vlastnosti každého útočníka: jeho motiváciu, jeho schopnosti, resp. skúsenosti s vykonávaním útokov, a jeho znalosti o systéme, na ktorý útočí. Práve tieto tri atribúty dokážu dobre rozhodovať o tom, do akej skupiny bude útočník zaradený.

V tretej kapitole sú popísané aktuálne prístupy k identifikácii útočníkov. Rôzni autori rozoznávajú rôzne typy útočníkov. Delia ich na základe ich motivácie, škody, ktorú dokážu spôsobiť alebo na základe iných kritérií. V kapitole sú popísané metódy, ktoré autori využívajú na odlíšenie kybernetických útočníkov.

Štvrtá kapitola rozoberá existujúce prístupy, ktoré je možné využiť na určovanie typov útočníkov a bližšie sa venuje jednému z nich- zhlukovaniu, konkrétne k-means algoritmu.

V piatej kapitole popisujeme a analyzujeme údaje, s ktorými pracujeme. Údaje sú vo formáte IDEA a v tejto kapitole rozoberáme, ktoré hodnoty sú podstatné pre identifikáciu útočníka. Zároveň sme v tejto kapitole navrhli vlastné riešenie pre zaradenie útočníka do skupiny. Toto riešenie je založené na využití zhlukovacieho algoritmu aplikovaného na množinu vopred pripravených vektorov, ktoré predstavujú útočníkov. Nájdené zhluky reprezentované centroidmi následne analyzujeme a popisujeme.

1 Honeypoty a honeynety

Na to, aby bolo možné zdokonaľovať bezpečnostné systémy a lepšie tak chrániť produkčné systémy pred škodlivou činnosťou prichádzajúcou či už zvonka alebo zvnútra, je v prvom rade potrebné poznať postupy útočníkov pri neautorizovanom vniknutí do takýchto systémov. Otázkou teda je, ako efektívne získavať tieto informácie. Jedným z riešení je využitie honeypotov, ktoré sú založené na relatívne jednoduchej myšlienke: nalákať útočníka a následne zaznamenávať jeho činnosť. V tejto kapitole bližšie definujeme pojem honeypot a sieť honeypotov – honeynet a poukážeme na výhody a nevýhody vyplývajúce z používania týchto systémov.

Na to, aby bolo možné zdokonaľovať bezpečnostné systémy a lepšie tak chrániť produkčné systémy pred škodlivou činnosťou prichádzajúcou či už zvonku alebo zvnútra, je potrebné poznať postupy útočníkov pri neautorizovanom vniknutí do takýchto systémov. Otázkou je, ako efektívne získavať tieto informácie. Jedným z riešení je využitie honeypotov. V tejto kapitole bližšie definujeme pojem honeypot a sieť honeypotov – honeynet a poukážeme na výhody a nevýhody vyplývajúce z používania týchto systémov.

1.1 Definícia honeypotu

Existuje niekoľko možností, ako definovať pojem honeypot. Zakladateľ Honeynet Project-u zadefinoval **honeypot** ako „informačný systém, ktorého význam spočíva v jeho neautorizovanom využití“ [10]. Honeypotom môže byť napríklad program, ktorý sa navonok javí ako služba atraktívna pre útočníka, no v skutočnosti je jedinou úlohou tohto systému útočníka nalákať a následne monitorovať jeho kroky pri vykonávaní útoku [9]. Honeypot dokáže monitorovať napríklad kombinácie mena a hesla skúšané útočníkom pri pokusoch o získanie prístupu do systému. Súčasne honeypot je schopný zaznamenávať všetky príkazy, ktoré útočník použil, pamätá si súbory, ku ktorým útočník pristupoval a ktoré modifikoval a monitoruje aj všetky spustené procesy. Je možné konštatovať, že honeypot je akási „pasca“ na útočníka, ktorej cieľom je, aby útočník zneužil ponúkanú službu, resp. operačný systém. Týmto honeypot poskytuje informácie o tom, aké kroky vykonáva útočník pri útoku a aké prostriedky na to využíva. Takéto informácie sú následne veľmi hodnotné pri ďalšej analýze a zdokonaľovaní bezpečnostných nástrojov.

1.1.1 Generický model honeypotu

Honeypot je zámerne navrhnutý tak, aby ho bolo možné vo verejnej sieti ohroziť. Súčasne musí byť navrhnutý tak, aby útočník netušil o tom, že nejde o produkčný systém, ale len o honeypot. Honeypot teda nemá žiadnu produkčnú hodnotu (až na to, že je používaný ako pasca na útočníkov) a zároveň každý pokus o pripojenie na takýto systém, resp. samotné pripojenie je vo väčšine prípadov považované za nelegitímne. Honeypot vo väčšine prípadov neposkytuje služby pre legitímnych používateľov. Táto skutočnosť so sebou prináša isté výhody pri analýze a spracovávaní údajov. Honeypot (ak neuvažujeme o sieti honeypotov) pozostáva z nasledujúcich častí [9]:

- produkčný system honeypotu,
- firewall,
- monitorovacia jednotka.
- zaznamenávacia jednotka a
- výstražná jednotka.

Produkčný systém honeypotu v skutočnosti nepredstavuje produkčný systém v pravom slova zmysle, ale len systém, ktorý poskytuje rôzne súbory a falošné prostriedky, ktoré má útočník v domnienke, že útočí na skutočný systém, zneužiť. Na všetky kroky útočníka je nastavená automatická odpoveď tak, aby útočník neodhalil, že útočí na honeypot.

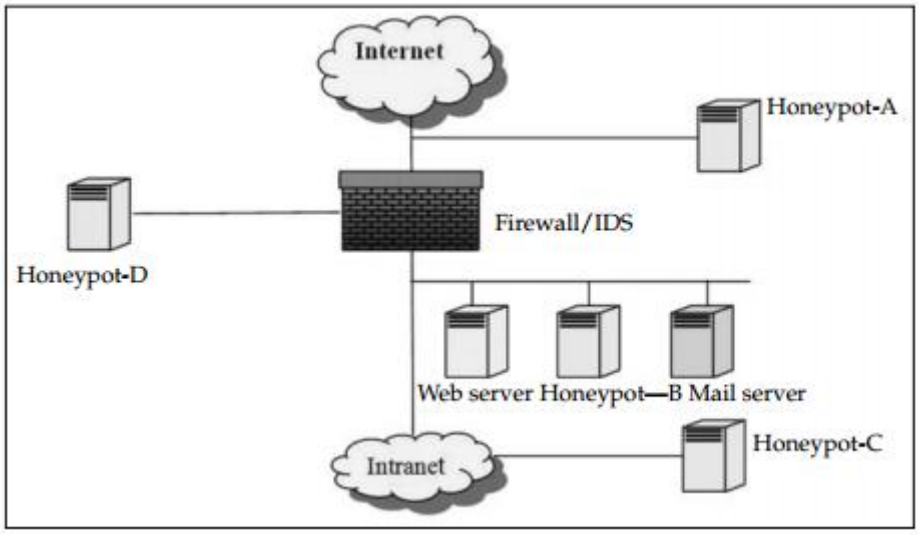
Firewall zaznamenáva a poskytuje informácie o tom, akým spôsobom sa útočník snažil získať prístup do systému. Zachytáva všetky pakety, ktoré prichádzajú do systému, keďže každá aktivita na honeypote je považovaná za nelegitímnu.

Monitorovacia jednotka vyhodnocuje riziká, monitoruje všetky aktivity v počítačovej sieti, a týmto odhaľuje nelegitímnu činnosť v systéme. V systéme sa ďalej nachádza riadiaca stanica, ktorá prijíma informácie o škodlivej činnosti od monitorovacej jednotky. Riadiaca stanica má za úlohu odhaliť útočníkove zámery, použitú metodiku a prostriedky tým, že analyzuje časové pečiatky, pakety, či súbory, ktoré boli zmenené. Ako monitorovacia jednotka môže slúžiť napríklad systém na detekciu útoku (Intrusion Detection System, IDS).

Zaznamenávacia jednotka slúži na uchovávanie záznamov systému, firewallu a tiež informácií o toku údajov medzi firewallom a honeypotom.

Výstražná jednotka slúži na generovanie upozornení a ich posielanie administrátorovi s informáciou o toku dát do alebo z honeypotu. Jej veľkou výhodou je, že administrátor môže sledovať útočníkovu aktivitu ešte v čase, keď prebieha.

Na obr. 1 je znázornené jedno z možných zapojení honeypotov pre ochranné účely. Honeypot A simuluje systém bez kontroly prístupu firewallom. Honeypot B simuluje zraniteľnú službu ako FTP (File Transfer Protocol, protokol na prenos súborov) alebo TELNET, ktoré majú za úlohu zaujať útočníka. Napokon honeypoty C a D simulujú iné systémy v počítačovej sieti organizácie s cieľom odlákať útočníka od reálnych produkčných systémov.



Obrázok 1 Príklad nasadenia honeypotov [9]

1.1.2 Výhody a nevýhody honeypotov

Honeypoty sú popri firewalloch a systémoch na detekciu útokov (IDS) veľmi úspešným bezpečnostným nástrojom už od ich vzniku. Avšak ako nástroj pre včasnú detekciu neautorizovanej činnosti prináša nie len výhody, ale aj riziká. Hlavné prínosy honeypotov sú [9]:

1. Údaje, ktoré sú zberané honeypotmi, sú vhodné na analýzu najmä kvôli tomu, že zaznamenávajú informácie len vtedy, keď s nimi niekto interaguje. Teda získavame tak údaje, ktoré sú jednoduchšie na spracovanie a ďalšiu analýzu.

2. Nezaznamenávajú falošné správy (tzv. „false positives“). Dôvodom je to, že každá interakcia s honeypotom je považovaná za neautorizovanú, a teda v zachytených údajoch sa nevyskytujú žiadne záznamy, ktoré by predstavovali reálnu prevádzku. S honeypotom by mal interagovať len administrátor a každá iná aktivita je považovaná za škodlivú. To robí analýzu získaných údajov o to jednoduchšiu napr. v porovnaní s IDS alebo firewallmi. To výrazne prispieva k rýchlej detekcii hrozieb a následnému včasnému generovaniu upozornenia.

3. Honeypoty vo veľkej miere prispievajú k odhaľovaniu tzv. „zero-day“ útokov, teda útokov, ktoré sa objavili po prvýkrát a doposiaľ neboli známe. Takéto útoky honeypot odhalí vďaka jeho vlastnosti, že každé pripojenie naň je škodlivé, rovnako rýchlo ako aj tie útoky, ktoré sú už známe. Tieto systémy dokážu zachytávať rôzne informácie o útočníkovi, ako napríklad všetky príkazy, ktoré zadal, aký malware uploadoval na systém, všetku komunikáciu cez chat, teda všetky sieťové pakety. Tak má administrátor prehľad o tom, čo útočník vykonáva a ako.

4. Medzi výhody honeypotov jednoznačne patrí aj ich jednoduchosť z dôvodu, že nevyžadujú žiadne špeciálne prostriedky na prevádzku, žiadne zložité algoritmy, nie je potrebné uchovávať žiadne stavové tabuľky a ani obnovovať elektronické podpisy za účelom šifrovania.

5. Honeypoty dokážu svoju funkciu plniť aj v šifrovanom prostredí alebo v prostredí s využitím IPv6. To je výhoda oproti mnohým iným bezpečnostným technológiám ako je napríklad IDS, lebo nezáleží na tom, akými prostriedkami útočník zaútočí, honeypot to vždy zachytí.

Okrem vyššie uvedených výhod honeypoty so sebou prinášajú aj isté riziká. Medzi tieto riziká môžeme zaradiť [9]:

1. V prípade, že na honeypot nikto neútočí, je jeho prevádzka zbytočná.

2. Ak útočník nezaútočí priamo na honeypot, ale na iný systém v rámci siete, kde sa nachádza aj honeypot, tak tento honeypot útočníkovu aktivitu nezachytí, pretože útok nie je namierený priamo naň. Navyše, ak útočník odhalí, že sa v sieti nachádza honeypot, tak sa mu môže vyhnúť a uskutočniť útok na systémy organizácie bez toho, aby o tom honeypot vedel.

3. Honeypot je možné aj zneužiť na vykonávanie ďalších útokov.

4. Je dôležité dbať na správne nastavenie honeypotu, pretože v opačnom prípade hrozí riziko tzv. „fingerprintingu“, teda riziko odhalenia honeypotu.

5. Fingerprinting predstavuje hrozbu najmä pre výskumné honeypoty, pretože ak útočník takýto honeypot odhalí, môže mu následne poskytovať falošné informácie, vykonávať kroky, ktoré by za normálnych okolností nevykonával, a tým znehodnotiť údaje, ktoré sa ďalej využívajú pri analýze.

Z dôvodu vyššie uvedených rizík, ktoré so sebou honeypoty prinášajú, nemôžu honeypoty úplne nahradiť iné bezpečnostné systémy. Avšak aj napriek tomu majú obrovský prínos pre výskum v oblasti kybernetickej bezpečnosti.

1.2 Definícia honeynetu

Pojem **honeynet** označuje sieť dvoch alebo viacerých honeypotov zapojených v sieti. Honeynety sa začali vyvíjať od roku 1999, keď už spomínaný Lance Spitzner navrhol ich koncept. Honeynety nachádzajú svoje uplatnenie pri monitorovaní rozsiahlejších sietí, kde by nasadenie samostatného honeypotu nebolo postačujúce [9].

Honeynet môže byť definovaný aj ako vysoko-interaktívny honeypot, teda honeypot poskytujúci skutočný, nie emulovaný, operačný systém [9]. Takýto druh honeypotu má vysokú úspešnosť v nalákaní útočníka, ale na druhej strane je kvôli použitiu skutočného operačného systému vystavený vysokému riziku.

Honeynet je vybudovaný zo štandardných produkčných systémov, ktoré sú umiestnené za zariadenie, ktoré kontroluje prístup do a zo siete. Takýmto zariadením môže byť napríklad firewall. Útočník môže útočiť na ktorýkoľvek z týchto systémov v honeynete a každá zachytená činnosť v tomto monitorovanom prostredí vypovedá o tom, aké metódy a aké nástroje útočníci využívajú, ale aj o tom, čo ich k danému útoku vedie. Vďaka honeynetu je možné získať údaje o každom kroku útočníka. Architektúra honeynetu pozostáva zo štyroch prvkov:

- kontrola údajov,
- zachytávanie údajov,
- zber údajov a
- analýza údajov.

Prvé dva prvky sú najpodstatnejšie a sú súčasťou každého honeynetu. Zber údajov sa týka len tých organizácií, ktoré prevádzkujú viacero honeynetov v distribuovanom prostredí. Dôležitou súčasťou je aj analýza údajov, bez ktorej by honeynet strácal svoj význam.

1.3 Prehľad existujúcich honeypotov

Rôzne typy honeypotov umožňujú zberať rôzne údaje o útoku, resp. o útočníkovi, ktorý za konkrétnym útokom stojí. Vyše 60% všetkých útokov sú práve útoky namierené proti webovým aplikáciám [11]. Cieľom týchto útokov sú väčšinou stránky organizácií prostredníctvom ktorých je potom zákazníkom sprostredkovaný škodlivý obsah alebo je spôsobený únik citlivých informácií. Na lepšie pochopenie týchto ale aj iných typov útokov sú k dispozícii honeypoty.

Cieľom bezpečnosti je ochrana počítačovej siete, informácií a zdrojov pred krádežou, znehodnotením, modifikáciou a zároveň zabezpečenie prístupu používateľov k týmto zdrojom. V minulosti bolo predstavených množstvo bezpečnostných riešení na obranu pred neautorizovaným prístupom k informáciám a iným internetovým zdrojom. Boli navrhnuté rôzne efektívne mechanizmy na detekciu útokov už v čase ich priebehu a na ich analýzu po ich ukončení. Avšak napriek mnohým výhodám týchto mechanizmov bolo veľmi zložité získať detailné informácie o útočníkovi. Honeypot je práve tou technológiou, ktorá dokáže zberať detailné údaje o útokoch tým, že útočníka naláka na vykonanie nelegálnej činnosti.

Pojem honeypot bol po prvýkrát použitý ešte počas studenej vojny ako nástroj na špehovanie. Až v roku 1990 sa tento pojem začal spájať s oblasťou informačnej bezpečnosti. Najväčšou výzvou pre organizáciu je poznať svojho „nepriateľa“, akým spôsobom môže zaútočiť, čo vykonáva po tom, ako sa mu podarí kompromitovať systém, ale aj to, prečo útočí. Honeypoty sú schopné práve tieto informácie organizácii poskytnúť.

V nasledujúcich podkapitolách sa bližšie zameriavame na niekoľko existujúcich honeypotov, najmä však na údaje, ktoré zberajú a ktoré sú využiteľné v rámci analýzy útokov a útočníkov.

1.3.1 Glastopf

Glastopf [32] je minimalistický nízko-interaktívny webový honeypot napísaný v Pythone, ktorý je navrhnutý na zachytávanie informácií o aktuálnych útokoch na webové aplikácie, ako sú napríklad SQL Injection, remote file inclusion a local file inclusion útoky. Emuluje tisícky zraniteľností. Glastopf skenuje prichádzajúce požiadavky a hľadá najmä reťazce ako „=http://“ alebo „CAST(0x“. V prípade, že sa nájde zhoda, stiahne sa súbor, analyzuje sa a Glastopf odpovie útočníkovi tak, aby to bolo čo najbližšie jeho očakávaniam. Ak splní útočníkove požiadavky, útočník poskytne ďalšie pre nás užitočné údaje.

Údaje, ktoré vieme získať o útoku s využitím honeypotu Glastopf sú: TIME (časová pečiatka), SOURCE (zdroj), REQUEST_URL (požadovaná URL adresa), REQUEST_RAW, PATTERN, FILENAME (názov súboru), COUNT, FIRSTTIME (časová pečiatka prvého prístupu), LASTTIME (časová pečiatka posledného prístupu), CONTENT (obsah).

1.3.2 Kippo

Kippo [36] je stredne-interaktívny SSH (Secure shell, zabezpečený komunikačný protokol) honeypot, ktorý je navrhnutý na zaznamenávanie brute force útokov a celej SHELL interakcie útočníka so systémom. Rovnako ako Glastopf je napísaný v Pythone. Umožňuje pridávať falošný obsah do súborov, čiže útočník môže prečítať súbory ako napríklad /etc/passwd s falošným obsahom. Poskytuje plný, avšak falošný súborový systém s možnosťou pridávať a odstraňovať súbory. Kippo počúva SSH spojenia na porte 2222, pričom port je možné zmeniť.

Záznamy relácií sú uchovávané v UML (Unified Modeling Lanfuage, jednotný jazyk pre modelovanie) kompatibilnom formáte pre jednoduchú rekonštrukciu priebehu útoku s pôvodným časovaním. Kippo ukladá všetky stiahnuté súbory pomocou príkazu WGET za účelom neskoršej analýzy.

Údaje, ktoré vieme získať o útoku pomocou honeypotu Kippo sú: SESSION (relácia), SUCCESS (informácia o úspešnosti útoku), USERNAME (používateľské meno), PASSWORD (heslo), TIMESTAMP (časová pečiatka), IP (IP adresa), STARTTIME (začiatok spojenia), ENDTIME (koniec spojenia), CLIENT, SENSOR, URL, OUTFILE.

1.3.3 Dionaea

Dionaea [37] je nízko-interaktívny honeypot zameraný na zachytávanie malvéru a na získavanie kópií malvéru. Je rovnako ako predchádzajúce dva honeypoty napísaný v Pythone a jeho veľkou výhodou je modulárna architektúra. Pôvodne bol tento honeypot vyvíjaný v rámci Honeynet Project-u. Je nasledovníkom honeypotu Nepenthes, oproti ktorému jeho výhodou je detekcia SHELL kódov s využitím LibEmu a podpora protokolov IPv6 a TLS (Transport Layer Security, protokol na zabezpečenie transportnej vrstvy). LibEmu je nástroj, ktorý Dionaea využíva na detekciu a evaluáciu údajov prichádzajúcich od útočníka s cieľom získať kópiu malvéru.Dionaea obsahuje napadnuteľné chyby v kóde. Na minimalizovanie dôsledkov Dionaea beží v obmedzenom prostredí bez akýchkoľvek administratívnych práv.

Tento honeypot využíva niekoľko protokolov, ktoré sú veľmi populárne najmä ako cieľ pre rôzne typy malvéru. Medzi tieto protokoly zaraďujeme [37]:

- **Server Message Block (SMB)** – populárny cieľ pre počítačové červy
- **Hypertext Transfer Protocol (HTTP)** – Dionaea podporuje HTTP na porte 80 a tiež podporuje aj HTTPS.
- **File Transfer Protocol (FTP)** – Dionaea poskytuje základný FTP server na porte 21. Umožňuje vytváranie priečinkov, nahrávanie a sťahovanie súborov.
- **Trivial File Transfer Protocol (TFTP)** – Dionaea poskytuje TFTP server na porte 60 a môže byť používaný na prácu so súbormi.
- **Microsoft SQL Server (MSSQL)** – počúva na TCP/1433, umožňuje prihlásenie používateľov a dokáže dekódovať dopyty do databázy.
- **Voice over IP (VoIP)** – čaká na prichádzajúce SIP správy, loguje všetky dáta ako incidenty a vhodne na ne reaguje.

Tak ako každý honeypot, aj Dionaea poskytuje logovanie do textových súborov, avšak takéto riešenie nie je dobre škálovateľné. Preto Dionaea poskytuje aj efektívnejšie riešenie, ktorým sú tzv. incidenty. Incidenty obsahujúce informácie o pôvode a vlastnostiach útoku sú prenášané pomocou iHandle. Zaujímavé incidenty sú následne zapisované pomocou Python skriptu do SQLite databázy. Výhodou takéhoto uchovávania údajov je schopnosť zhlukovania informácií pri ich získavaní z databázy.

Akonáhle sa získa kópia malvéru, je možné ju buď uložiť lokálne alebo preposlať tento súbor externému nástroju alebo službe pre ďalšiu analýzu.

Údaje, ktoré môžeme získať o útoku pomocou tohto honeypotu sú: CONNECTION (spojenie), CONNECTION_TYPE (typ spojenia), CONNECTION_TRANSPORT, CONNECTION_PROTOCOL (protokol), CONNECTION_TIMESTAMP (časová pečiatka nadviazania spojenia), CONNECTION_ROOT, CONNECTION_PARENT, LOCAL_HOST (lokálne zariadenie), LOCAL_PORT (lokálny port), REMOTE_HOST (vzdialené zariadenie), REMOTE_HOSTNAME (meno vzdialeného zariadenia), REMOTE_PORT (vzdialený port).

2 Útok a útočník

Každý útok je špecifický, vykonávaný inými metódami, resp. využívajúc rozličné nástroje. Za každým útokom stojí útočník, ktorý do veľkej miery ovplyvňuje jeho priebeh, určuje cieľ útoku a je zodpovedný za dopad tejto škodlivej činnosti na systém, na ktorý útočí. Poznať vlastnosti útočníka je preto dôležitým aspektom pri ochrane pred kybernetickými útokmi a pri návrhu rôznych bezpečnostných systémov. V tejto kapitole si definujeme základné pojmy týkajúce sa kybernetických útokov a útočníkov, bližšie sa pozrieme na konkrétne vlastnosti útočníkov a popíšeme klasifikáciu útočníkov.

2.1 Definícia útoku

Pojem **kybernetický útok** označuje akýkoľvek typ škodlivej činnosti vykonávanej jednotlivcom, skupinou, organizáciou alebo štátom, ktorá je namierená na počítačové informačné systémy, infraštruktúru, počítačové siete alebo osobné počítače. Jeho cieľom je touto činnosťou odcudziť, modifikovať alebo znehodnotiť špecifický cieľ využívajúc neautorizovaný prístup do systému [22].

2.1.1 Vektor útoku

Každý útok pozostáva z konkrétnych krokov, ktoré útočník vykonáva. Táto postupnosť krokov sa nazýva **vektor útoku**. Niektoré vektory môžu pozostávať aj z viacerých hrozieb a naopak niektoré kroky, ktoré útočník vykonal nemusia byť škodlivé. Každý krok môže zahŕňať objekt, na ktorý útočník útočí, jeho zraniteľnosti, nástroj, pomocou ktorého je možné túto zraniteľnosť využiť a presunúť sa na ďalší objekt, čo napokon vytvára úspešný útok. Znalosť vektora útoku je veľmi dôležitá kvôli správnemu pochopeniu detailov útoku a lepšej obrane proti ďalším útokom [12].

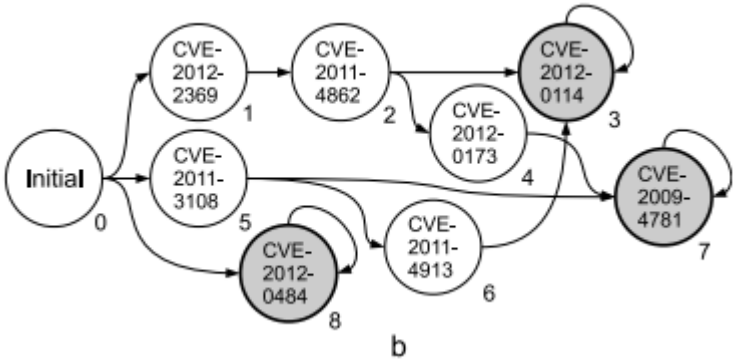
2.1.2 Graf útoku

Graf útoku je stručná reprezentácia všetkých možných „ciest“ v systéme, ktoré vedú do stavu, kedy útočník úspešne dosiahol svoj cieľ [23]. Moderné techniky na prácu s grafmi útokov dokážu automaticky odhaliť všetky možné spôsoby, akými môže

útočník kompromitovať počítačovú sieť, a to analýzou konfiguračných údajov serverov a siete [24].

Graf útoku priamo zobrazuje logické závislosti medzi konfiguráciami systémov a právami, ktoré má útočník k dispozícii. Niektoré typy týchto grafov znázorňujú „prečo k útoku môže dôjsť“, iné zas to, „ako môže k útoku dôjsť“. Grafy útokov slúžia ako základ pre detekciu, obranu pred útokmi a forenznú analýzu [24].

Na nasledujúcom obrázku máme možnosť vidieť typ grafu útoku, ktorý zobrazuje, aké zraniteľnosti v systéme môže útočník využiť na dosiahnutie svojho cieľa. Uzly zafarbené na sivo sú cieľové uzly [1].



Obrázok 2 Graf útoku [1]

Na to, aby sme lepšie pochopili taktiky útokov vo všeobecnosti a vedeli, ako sa proti nim brániť, sa pozrieme na 3 typy útokov:

- útoky proti smart systémom,
- APT útoky a
- cielené útoky.

2.1.3 Útoky proti smart systémom (CPS)

Smart systém (alebo inak cyber-physical system, CPS) je systém spolupracujúcich výpočtových jednotiek kontrolujúcich fyzické entity. Väčšinou je navrhnutý ako sieť interagujúcich prvkov. Patria sem najmä systémy pre priemyselnú výrobu, na ktoré majú útoky veľmi škodlivý dopad. Tento typ útoku musí najprv odhaliť zraniteľnosti všetkých komponentov systému, ktoré môžu byť softvérové, hardvérové ale aj také, ktoré súvisia s ľudským faktorom. V závislosti od zraniteľnosti

väčšinou nasleduje cielený phishing. Ako náhle je systém napadnutý a škodlivý kód nainštalovaný, útočník vytvorí komunikačný kanál a môže riadiť a kontrolovať systém. Charakteristickou črtou týchto útokov je priamy dosah na zariadenia fyzického sveta. Na ich vykonanie je však často potrebná znalosť konkrétneho systému najmä pri SCADA (Supervisory Control And Data Acquisition) systémoch, teda systémoch pre dispečerské riadenie a zber údajov. Za týmito útokmi sú väčšinou útočníci zo skupiny cyber teroristov, cyber kriminálnikov alebo hacktivistov [12].

2.1.4 Cielené útoky

Ide o útoky namierené proti konkrétnemu jednotlivcovi, spoločnosti, systému alebo softvéru s využitím konkrétnych poznatkov o cieli [12]. Tieto útoky nie sú až tak rozšírené. Na základe poznatkov o cieli, útočník pošle špecifické správy s cieľom nalákať obeť, s tým, že tieto falošné správy sa nedajú odlíšiť od skutočných. V inicializačnej fáze, keď útočník zbiera informácie o systéme, sa zameriava na IT prostredie, infraštruktúru systému ale aj na informácie o jednotlivých osobách. Po získaní prístupu sa využije zraniteľnosť na vykonanie škodlivého kódu. Väčšinou ide o stiahnutie malvéru do systému koncového používateľa, ktorý vytvorí komunikačný kanál s útočníkom, ktorý následne môže vykonávať rôzne akcie (napríklad získavať informácie o spoločnosti). Časté metódy využívané pri tomto typu útoku sú phishing email, zero-day útoky, ale útočníci tiež využívajú sociálne siete k tomu, aby čo najviac používateľov kliklo na nejaký odkaz a stiahlo malvér. Súčasťou cielených útokov môžu byť všetky skupiny útočníkov [12].

2.1.5 Pokročilé dlhotrvajúce hrozby

Pokročilé dlhotrvajúce hrozby (Advanced persitent threats, APT) predstavujú rozmanitú množinu utajených procesov namierených proti špecifickým objektom. Sú väčšinou vykonávané v rámci kampaní proti konkrétnym organizáciám. Hlavným cieľom je získanie dát a nie poškodenie siete. Úspešný útok tohto typu vyžaduje utajenosť počas celej dĺžky trvania útoku. Tí, ktorí stoja za týmto typom útoku majú väčšinou pokročilé znalosti, všetko dopredu plánujú, využívajú špeciálne vytvorené malvéry a detailné poznatky o obeti. Útočníci väčšinou disponujú rozsiahlymi zdrojmi a finančnými prostriedkami. Cieľom je umiestniť vytvorený

škodlivý kód na jeden alebo viac počítačov a vykonávať ho pre dosiahnutie konkrétneho cieľa, čo najdlhšiu dobu. Ide o útoky, ktoré sú cielené, útočníci majú splniť nejakú misiu, ktorá je organizovaná, dobre finančne zabezpečená a silno motivovaná. Hlavným znakom je dlhé trvanie útoku. Sú tiež podporované externe kyberkriminálnikmi, hacktivistami či dokonca špecializovanými spoločnosťami [12].

2.2 Definícia útočníka

Kybernetický útočník je definovaný ako akákoľvek osoba, ktorá zámerne využíva zraniteľnosti v technických, ale aj netechnických kontrolných prvkoch s cieľom odcudziť alebo kompromitovať informačné systémy a siete alebo zamedziť dostupnosť k informačným systémom a sieťovým zdrojom pre legitímnych používateľov [26].

Ak hovoríme o skupinách útočníkov, tak sa zameriavame na ich motiváciu a schopnosti, pričom keď rozprávame o útokoch, tak nás zaujímajú nástroje a metódy, ktoré útočníci využívajú. V nasledujúcom texte sa budeme zameriavať práve na motiváciu a schopnosti útočníkov.

2.3 Vlastnosti útočníka

Diskusie o útočníkoch sa zameriavajú na rozpoznávanie ich motivácie a schopností. V rámci týchto diskusií nás zaujímajú najmä nástroje a metódy, ktoré útočníci využívajú. Okrem motivácie a schopností je dôležité venovať pozornosť aj znalostiam útočníka. Podľa dostupných zdrojov [12,13,28] delíme vlastnosti útočníka na 3 skupiny:

- **Schopnosti útočníka** - pod schopnosťami útočníka budeme rozumieť jeho schopnosti v oblasti kybernetickej bezpečnosti, resp. jeho skúsenosti s útokmi na rôzne systémy.
- **Znalosti útočníka** - pod znalosťami útočníka rozumieme to, do akej miery pozná systém, resp. štruktúru siete, na ktorú útočí. Táto vlastnosť rozhoduje napríklad o tom, či ide o útočníka zo skupiny „insider“ alebo nie, a je možné ju určiť napríklad z počtu pokusu o pripojenie.
- **Motivácia útočníka** – predstavuje cieľ, ktorý chce útočník dosiahnuť.

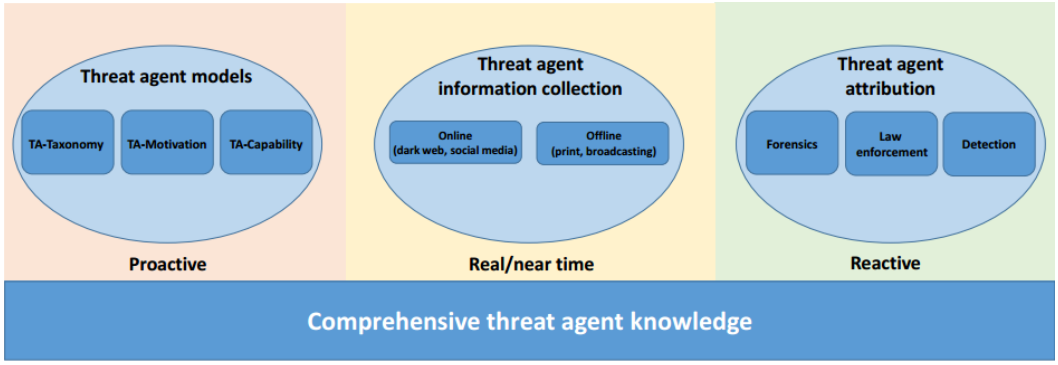
2.3.1 Znalosti útočníka

Celková znalosť o útočníkovi pozostáva z troch kategórií znalostí útočníka [12]. Prvou kategóriou sú **proaktívne znalosti**, ktoré pokrývajú všetky relevantné parametre ako rôzne skupiny útočníkov, ich schopnosti a zručnosti, ich motiváciu, interakciu v skupinách atď. Tieto znalosti sa berú do úvahy pri vyhodnocovaní hrozieb a rizík.

Druhou kategóriou sú **znalosti v reálnom čase**, ktoré pozostávajú z informácií zozbieraných z online zdrojov ako web či sociálne médiá, z tlače a masmédií. Príkladom týchto znalostí sú inštitúcie, ktorých ukradnuté karty sú ponúkané online. Tento typ znalostí je užitočný pre bezpečnostné operácie a plánovanie, ako aj pre vyhodnocovanie hrozieb a rizík.

Poslednou kategóriou sú **reaktívne znalosti**, ktoré sú výsledkom analýzy rôznych bezpečnostných incidentov. Pomáhajú lepšie chápať činnosť skupín útočníkov, ich motivácie, metódy využívané pri útokoch. Sú poskytované obchodníkmi, právnym vynútením a bezpečnostnými agentúrami po analýze bezpečnostných incidentov. Informácie o útočníkoch z vnútra organizácie (insideroch) a ich motiváciách viedli k detailným poznatkom tejto skupiny [12].

Pokrok v tejto oblasti by bol ďalším krokom k zdokonaľovaniu ochrany pred kybernetickými útokmi.



Obrázok 3 Znalosti útočníkov [12]

2.3.2 Motivácia útočníka

S každodennými operáciami v takmer každej oblasti spoločnosti, ktoré čím viac závisia od poprepájaných počítačov, stability globálnej ekonomiky či rôznych sociálnych a politických systémov, sa zvyšujú aj požiadavky na spoľahlivé fungovanie internetu a intranetových systémov. Kvôli tomu rastie aj dôležitosť kyberbezpečnosti. Napriek tomu, že vláda a rôzne korporácie sa zameriavajú aj na kybernetickú

bezpečnosť, tak globálne zostáva kybernetický priestor zraniteľný a narastá počet zneužitia internetu pre nelegálne činy.

Jednou zo stratégií ako dosiahnuť ciele v oblasti kybernetickej bezpečnosti je zdokonaliť metódy pre klasifikáciu kyberútočníkov. Užitočný prístup ku klasifikácii útočníkov je vytvorenie typológie, ktorá umožňuje bezpečnostným analytikom účinnejšie identifikovať hrozby založené na známych typoch útočníkov. Takéto typológie umožňujú lepšie pochopiť postupy útočníkov, ale sú náročné na vytvorenie, najmä ak ide o útočníkov, ktorých identity sú často anonymné. Výzvou je preto určiť, kto je páchateľom, aké schopnosti tento páchateľ má a čo ho viedlo k vykonaniu daného činu.

Nikitina [13] opisuje útočenie na počítačové systémy ako spoločenský fenomén-produkt mladých ľudí vyrastajúcich v rozvíjajúcej sa digitálnej dobe s cieľom niečo napadnúť, narušiť, zničiť. Takáto činnosť je považovaná za logicky a sociálne motivovanú kybernetickú aktivitu ako je **hacktivismus** a **crowdsourcing.**

2.3.3 Schopnosti útočníka

Metódy, ktoré útočníci využívajú pri útokoch sa neustále vyvíjajú a zdokonaľujú. Kvôli stále sofistikovanejším postupom útočníkov sa musia aj bezpečnostné systémy prispôsobovať týmto trendom. Čím sú schopnosti útočníka v tejto oblasti na vyššej úrovni, tým sa tento útočník stáva pre organizáciu väčšou hrozbou. Podľa autorov článku zaoberajúcim sa taxonómiou kyberútočníkov [28] môžeme schopnosti útočníkov odstupňovať a zaradiť do kategórií:

- veľmi nízke schopnosti,
- nízke schopnosti,
- priemerné schopnosti,
- vysoké schopnosti a
- veľmi vysoké schopnosti.

Tabuľka 1 Schopnosti útočníkov

Trieda útočníkov	Schopnosti
Script kiddies	veľmi nízke
Hacktivisti, politickí aktivisti	nízke
Cyber punks	nízke
Insideri	priemerné
Programátori	vysoké
White hat hackeri	vysoké
Black hat hackeri	veľmi vysoké
Kybernetickí teroristi	veľmi vysoké

Ako je možné vidieť v tabuľke 1, útočníci z kategórie **script kiddies** majú schopnosti, resp. skúsenosti s vykonávaním útokov na veľmi nízkej úrovni. Majú nízke programátorské zručnosti, väčšinou len stiahnu a spustia už existujúci skript, tzv. toolkit. Preto nepredstavujú pre organizácie až takú vysokú bezpečnostnú hrozbu, avšak s narastajúcou sofistikovanosťou týchto skriptov narastá aj možnosť týchto útočníkov vykonať útok spôsobujúci vyššiu škodu.

Medzi útočníkov so **schopnosťami na nízkej úrovni** patria tzv. hacktivisti, ktorí majú väčšinou politické ciele a pre svoje útoky využívajú najmä DDoS metódu. Na rovnakej úrovni schopností sú aj útočníci typu cyber punks. Tí väčšinou dokážu písať vlastné skripty prispôsobené ich cieľom. Pre organizácie predstavujú hrozbu na miernej úrovni.

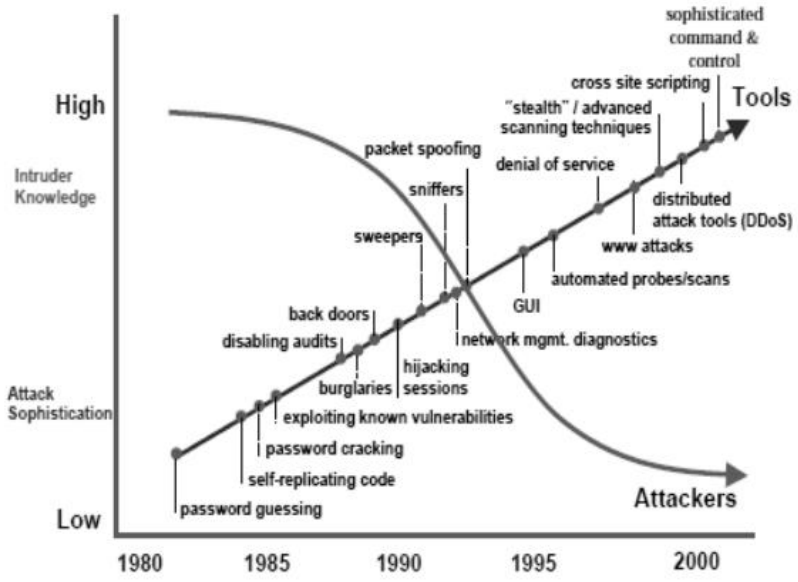
Schopnosti na priemernej úrovni majú tzv. insideri, ktorí majú k dispozícii údaje, ktoré vedia zneužiť pre napadnutie organizácie, v ktorej pracujú, alebo v ktorej pracovali. Vďaka týmto údajom síce nepotrebujú špeciálne schopnosti na vysokej úrovni, ale pre organizáciu predstavujú vysokú hrozbu.

Vysoké schopnosti majú programátori, ktorí píšu skripty, ktoré sú potom využívané script kiddies útočníkmi, a „white hat“ hackeri. Tí však nepredstavujú žiadnu hrozbu, keďže svojou činnosťou len pomáhajú napr. pri zdokonaľovaní bezpečnostných systémov.

„Black hat“ útočníci a tzv. cyber teroristi majú schopnosti na veľmi vysokej úrovni. Vykonávajú vysoko sofistikované útoky, ktoré sú namierené proti

nepriateľským národom (cyber teroristi) alebo sú súčasťou organizovaného zločinu (black hat). Predstavujú veľmi vysokú hrozbu, keďže majú veľmi silné technické znalosti a skúsenosti.

Na nasledujúcom obrázku (obr. 4) je vidno, ako sa postupom času vyvíjali nástroje používané pre útoky a zároveň schopnosti útočníkov potrebné na vykonanie útoku.

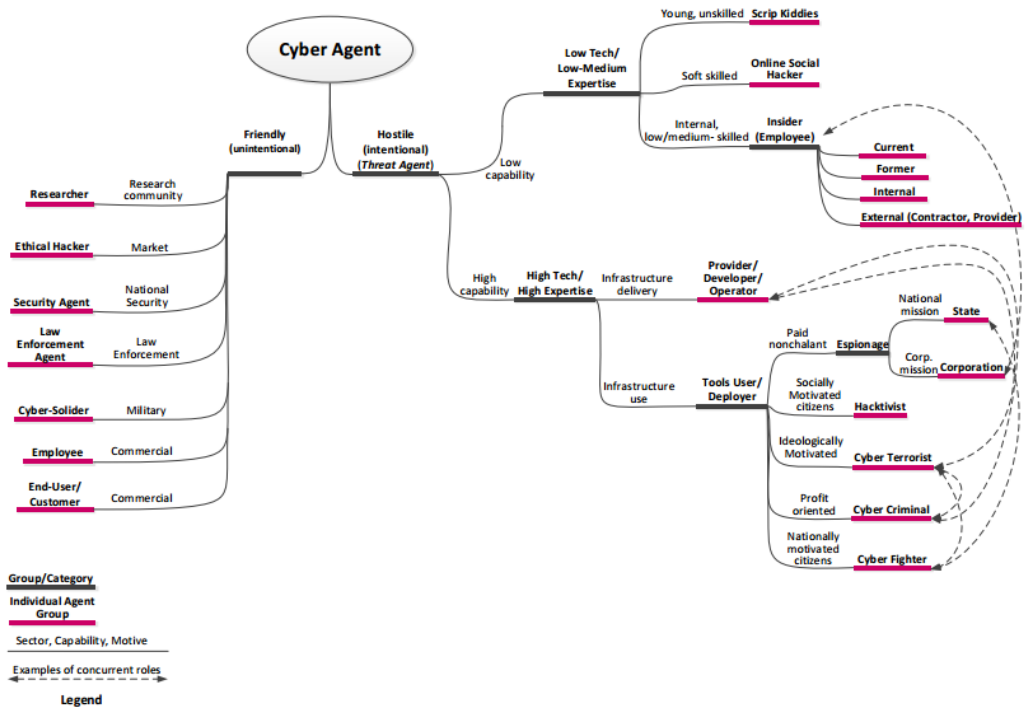


Obrázok 4 Vývoj schopností útočníkov [28]

2.4 Klasifikácia útočníkov

Útočníkov v kyberpriestore je veľmi náročné identifikovať a nájsť. Čiže len veľmi málo z nich si za svoje činy odpykáva trest. To je ďalší dôvod rozmáhania sa práve takéhoto typu útokov.

Existuje niekoľko prístupov ku klasifikácii a deleniu útočníkov. Bližšie sa týmto prístupom budeme venovať v nasledujúcej kapitole. Pre účel tejto záverečnej práce sme si vybrali klasifikácie útočníkov podľa ENISA Threat Landscape 2015 [12]. Obrázok 5 znázorňuje klasifikáciu útočníkov. Podľa tejto štúdie je najviac útočníkov zaradených do skupiny kyberkriminálnici, insideri, kyberšpióni a cyber warriors [12].



Obrázok 5 Klasifikácia útočníkov [12]

Nasledujúci obrázok zobrazuje, ktoré typy útočníkov sú pôvodcami ktorých typov útokov.

	Threat Agents								
	Cyber criminals	Insiders	Online social hackers	Nation States	Corporations	Hacktivists	Cyber Fighters	Cyber terrorists	Script kiddies
Malware	✓	✓	✓	✓	✓	✓	✓	✓	✓
Web-based attacks	✓			✓	✓	✓	✓	✓	✓
Web application attacks	✓			✓	✓	✓	✓	✓	✓
Botnets	✓			✓	✓	✓	✓	✓	✓
Denial of service	✓			✓	✓	✓	✓	✓	✓
Physical damage/ theft /loss	✓	✓		✓	✓			✓	
Insider threat	✓	✓		✓	✓			✓	
Phishing	✓	✓	✓	✓	✓	✓	✓	✓	✓
Spam	✓		✓	✓	✓	✓	✓	✓	✓
Exploit kits	✓			✓	✓	✓			✓
Data breaches	✓	✓		✓	✓	✓	✓	✓	✓

Obrázok 6 Mapa útočníkov a hrozieb [12]

2.4.1 Kybernetickí kriminálnici (cyber-criminals)

Ich hlavnou motiváciu je speňaženie poskytovaných služieb. Pracujú na zlepšovaní využívaných infraštruktúr aj na vyvíjaní stále sofistikovanejších škodlivých nástrojov [12].

2.4.2 Útočníci z vnútra organizácie (insiders)

Táto skupina útočníkov je v súčasnosti pomerne dobre rozanalyzovaná a je vytvorená detailnejšia klasifikácia tohto typu útočníkov. Ide o skupinu súčasných a bývalých zamestnancov, súčasných a bývalých poskytovateľov/dodávateľov/ konzultantov, súčasných a bývalých obchodných partnerov a zákazníkov. Okrem speňaženia a istého druhu pomsty, je u nich najväčšou motiváciou zneužitie prístupových práv. Ukázalo sa, že najčastejšie zneužívané prístupové údaje sú údaje koncových používateľov, zákazníkov, finančníkov a vedúcich pracovníkov. Systémoví administrátori sú až na deviatom mieste. Často krát majú útočníci z inej skupiny záujem využiť insiderov na dosiahnutie vlastného cieľa. Pričom najčastejší spôsob, ako ich získať je podplatenie. Súčasné informácie o tejto kategórii boli získané pomocou modelovania a tiež pomocou reakčnej analýzy [12].

2.4.3 Sociálni hackeri (online social hackers)

Počet týchto útočníkov sa zvyšuje vďaka phishingovým útokom, ktoré sú zamerané vždy na konkrétne skupiny obetí. Na získanie informácií o tejto skupine sú dôležité informácie od poskytovateľov sociálnych sietí. Nástroje pre tento typ útokov sú ľahko dostupné, takže jednotlivcom s nejakou konkrétnou motiváciou nestojí nič v ceste. Informácie o nich sú získavané na základe reakčnej analýzy, t.j. na základe analýzy konkrétnych bezpečnostných incidentov [12].

2.4.4 Hacktivisti (hacktivists)

Ich hlavným cieľom je šíriť informácie organizácií alebo vplyvných ľudí s cieľom zahanbiť ich a zvýšiť informovanosť verejnosti o tom, čo robia ilegálne. Propagujú slobodu vyjadrovania a otvorenosť internetu. Často majú tieto skupiny rovnaké ciele ako bezpečnostné spoločnosti, a preto by bolo spojenie týchto dvoch skupín zaujímavým riešením rôznych konfliktov [12].

2.4.5 Kybernetický teroristi (cyber-terrorists)

Moderné technológie internetu sa stali komunikačným kanálom a kanálom na verbovanie nových členov tejto skupiny útočníkov. Využívajú pritom útočníkov klasifikovaných ako social online hackers na udržiavanie ich infraštruktúry a na šírenie ich kampane po sociálnych sieťach. Ich záujmom je aj finančný zisk [12].

2.4.6 Script kiddies

Na internete sa nachádza množstvo informácii k tomu, aby bolo možné vykonať útok alebo vytvoriť vlastný škodlivý kód (malvér). Vďaka tomu rastie počet účasti tejto skupiny na rôznych bezpečnostných incidentoch. Niektorí z týchto útočníkov nemajú žiaden vážny zámer a vykonávajú túto činnosť len pre zábavu. Často krát ani nevedia, čo svojim útokom spôsobia. Za posledné obdobie sa ukázalo, že za týmito útokmi stoja najmä teenageri [12].

3 Aktuálne prístupy k identifikácii typov útočníkov

Jednou z úloh sietí je zabezpečiť anonymitu odosielateľa, príjemcu alebo oboch. Dosiahnutie stavu anonymity implikuje, že v sieti nie je žiaden útočník alebo útočník nie je úspešný pri svojich pokusoch o útok. To, či útočník bude alebo nebude úspešný pri svojej činnosti, sa zisťuje na základe bezpečnostného hodnotenia alebo analýzy rizík. Najkritickejšou časťou je určenie správneho modelu útočníka. Ak je tento model príliš silný, tak väčšina techník na ochranu nebude fungovať. Naopak, ak je model útočníka príliš slabý, systém nebude poskytovať dostatočnú ochranu používateľov. V rámci tejto kapitoly sa budeme venovať aktuálnym prístupom k identifikácii útočníkov, resp. konkrétnym modelom útočníkov.

3.1 Všeobecné prístupy k identifikácii útočníkov

Talianski autori [1] sa vo svojom článku venujú modelovaniu útočníkovho správania pomocou grafov a Markovových rozhodovacích procesov na predikciu možných rozhodnutí útočníka. Zároveň predpokladajú, že útočník nemá presné a detailné informácie o systéme, na ktorý útočí, čo ho zaraďuje do kategórie tzv. **adaptívnych útočníkov.** Na rozdiel od **deterministických útočníkov**, ktorí systém dobre poznajú a majú dopredu naplánované kroky útoku, adaptívny útočník môže prehodnocovať a meniť svoje rozhodnutia v rámci útoku v závislosti od vzniknutých situácií. Deterministickí útočníci sa v každom okamihu útoku dokážu rozhodnúť, aký bude nasledujúci krok s cieľom dosiahnuť optimálnu cestu v grafe útoku. Graf útoku je podľa autorov článku popísaný nasledovne: Uzol S_i reprezentuje úspešne napadnutú zraniteľnosť, hrana medzi dvoma uzlami určuje možné využitie ďalšej zraniteľnosti a takéto postupné využívanie zraniteľností vedie k novým stavom v systéme a novým možnostiam pre útočníka.

Podľa článku z roku 2005 [2] je dôležité odhaliť riziká na základe vektorov útokov. Jedným zo spôsobov, ako predchádzať bezpečnostným incidentom je zistiť, ktoré typy útočníkov sú pre systém najnebezpečnejšie a zamerať sa na ochranu aktív, ktoré s veľkou pravdepodobnosťou títo útočníci napadnú. Autori vytvorili klasifikáciu útočníkov podľa ich záujmov a cieľov. Podľa daného článku a článku z roku 2003 [3] vieme útočníkov rozdeliť na skupiny a následne ich ohodnotiť podľa toho, aké **riziko pre systém predstavujú.** Ide o skupiny:

- **neštruktúrovaný hacker,**
- **štruktúrovaný hacker,**
- **organizovaný zločinec a priemyselná e-špionáž,**
- **insider,**
- **hacktivista,**
- **financovaná teroristická skupina,**
- **štát,**
- **skript kiddie a**
- **„hobby“ hacker.**

Výsledné ohodnotenie útočníka závisí od viacerých aspektov:

- od jeho technických **zručností**,
- od **zdrojov**, ktoré ma k dispozícii,
- od jeho **zámeru** a **motivácie**, a
- od **pravdepodobnost**i, s akou je **systém napadnuteľný** daným typom útočníka.

V článku autori odvodili vzťah na číselné ohodnotenie toho, aké riziko daný útočník predstavuje a na číselné ohodnotenie jeho schopností. Podľa týchto hodnotení je potrebné nasadiť príslušné opatrenia proti danému útočníkovi.

V ďalšej práci sa autori [4] zaoberajú profiláciou útočníkov ako koncept oddelenia vlastností infraštruktúry systému od vlastností útočníkov. V článku autori ukazujú, ako je možné **poznatky o profilácii útočníka integrovať** do existujúcich bezpečnostných nástrojov bez akejkoľvek ujmy na výkonnosti. Ako príklad takejto integrácie uvádzajú analytický nástroj **AproxTree+,** ktorý je rozšírením existujúceho nástroja AproxTree. Pri analýze využívajú stromovú štruktúru na reprezentáciu útoku, ktorá predstavuje hierarchický popis možných útokov proti cieľovej infraštruktúre. V strome sú pokryté všetky možné scenáre útoku. Berúc do úvahy profiláciu útočníka, je možné prechádzaním stromu vylúčiť určité uzly, a tým celé podstromy a naopak identifikovať tie scenáre, ktoré pri danom type útočníka ohrozujú konkrétny systém. V závislosti od množstva a detailnosti informácií, ktoré sú o útočníkovi k dispozícii, je scenár vymedzený užšie a je ho tak možné ľahšie analyzovať.

Autori v nasledujúcom článku [5] priniesli ďalší pohľad na klasifikáciu útočníkov. V práci autori navrhli novú metódu charakterizácie útočníkov, ktorá je menej abstraktná ako v niektorých iných riešeniach a viac praktická a realistická. Autori predpokladajú, že útočník pozná infraštruktúru siete a jej algoritmy, keďže väčšina implementácií je open-sourcová a dobre zdokumentovaná. Vytvorili delenie útočníkov ako entít zúčastňujúcich sa nejakým spôsobom transakcií medzi dvoma stranami

využívajúc anonymnú sieť. Trieda **externých strán** je považovaná za najslabšieho útočníka, jeho vplyv je limitovaný keďže nemá žiadnu kontrolu nad počítačom medzi dvoma komunikujúcimi stranami. **Poskytovateľ služby** je trieda predstavujúca komunikačného partnera používateľa, je zviazaný s komunikujúcimi stranami, čo je možné ľahko zneužiť. Útočníci patriaci do triedy **lokálnej administrácie** môžu manipulovať so všetkým v blízkosti používateľa, čo je nebezpečné najmä vtedy, keď používateľ slepo dôveruje všetkým prenášaným údajom. Ďalším silným útočníkom je **poskytovateľ internetového pripojenia,** keďže má prístup k veľkému počtu počítačov. Čiže je možné, že väčšina dátového toku medzi dvoma komunikujúcimi stranami je zachytávaná týmto typom útočníka. **Vláda** je ďalším typom nebezpečného útočníka pretože má prístup k významnej časti sietí, má zdroje na vytváranie falošných služieb, zdroje na prelomenie jednoduchších šifrovacích schém alebo zdroje na zakázanie prístupu k špecifickým službám. **Tajné služby** tvoria najvyššiu triedu útočníkov. Môžu získať prístup k väčšine častí globálnych sietí, ak je to nevyhnutné pre ich operáciu. Okrem toho, nepodliehajú žiadnemu typu zákona, čo z nich robí vážnu bezpečnostnú hrozbu.

V práci [6] sa autori zameriavajú na motiváciu útočníka. Navrhli framework, ktorý zahŕňa motiváciu útočníka v analýze útoku pomocou stromu útoku. Rozlišujú 5 typov motivácií útočníka:

- **finančný zisk,**
- **spôsobenie škody,**
- **získanie znalosti,**
- **vyhľadávanie zábavy a**
- **získanie notoričnosti v rámci komunity.**

Ďalší krok v určovaní profilu útočníka je určenie zdrojov, ktorými útočník disponuje. Do úvahy autori berú hodnoty nasledovných parametrov: **financie**, ktoré má útočník k dispozícii; **schopnosti** útočníka; **čas,** ktorý má útočník k dispozícii na vykonanie útoku. Po nastavení stromu útoku a profilu útočníka sa tieto dva aspekty skombinujú a vykoná sa analýza, ktorá vedie k presnejšiemu typu útočníka. Existujú mnohé ďalšie klasifikácie, napríklad v [7] nájdeme 4 typy útočníkov:

- **eavesdropper,**
- **global eavesdropper,**
- **pasívny nepriateľ a**
- **aktívny nepriateľ.**

Problém s týmto delením je však ten, že v praxi je medzi týmito kategóriami veľmi malý rozdiel a je ťažké presne začleniť útočníka do niektorej z nich.

Systematické rozdelenie útočníkov pre teoretické modelovanie je popísané v [8]. Podľa toho, či je útočník v sieti alebo nie, rozlišujeme **interných a externých** útočníkov. Podľa toho, či útočník môže zmeniť stav siete alebo nie rozlišujeme **pasívnych a aktívnych** útočníkov a podľa toho, či útočník môže meniť svoje zdroje a vyvíjať svoje schopnosti počas útok alebo nie, rozlišujeme **statických a adaptívnych** útočníkov.

Z predchádzajúceho prehľadu existujúcich taxonómií útočníkov je možné zhrnúť niekoľko kritérií, ktoré autori pri tvorbe týchto taxonómií väčšinou berú do úvahy. Jedným z najčastejších kritérií je **motivácia útočníka.** Tá do veľkej miery určuje, koľko času a zdrojov útočník útoku venuje, ale aj to, aký dopad bude mať daný útok na systém. Ďalším kritériom je **miera účasti v komunikácii.** Toto kritérium uvažuje útočníkov ako entity, resp. väčšie celky, pričom každý z nich má odlišný prístup k zdrojom. Medzi dôležité kritériá patrí aj **riziko**, ktoré daný typ útočníka predstavuje pre systém. Na základe tohto poznatku je možné azda najefektívnejšie a najrýchlejšie rozhodnúť, do akej miery je potrebné nasadiť bezpečnostné opatrenia. Ako už bolo spomenuté, ďalšími kritériami sú **účasť útočníka v sieti, schopnosť útočníka zmeniť stav siete** a **vývoj schopností útočníka** počas útoku. Posledným spomínaným kritériom je **rozsah informácií**, ktoré útočník má k dispozícii o systéme, na ktorý útočí. Čím viac týchto relevantných informácií má, tým je pre systém nebezpečnejší. Tieto kritériá pre taxonómie kybernetických útočníkov zobrazuje aj nasledujúca tabuľka 2.

Tabuľka 2 Kritériá taxonómií útočníkov

Kritérium	**Skupiny útočníkov**
Motivácia	Útočník motivovaný: finančným ziskom, spôsobením škody, získaním znalostí, zábavou, notoričnosťou
Miera účasti v komunikácii	Externá strana, poskytovateľ služby, lokálny administrátor, poskytovateľ internetového pripojenia, vláda, tajné služby
Riziko	neštruktúrovaný hacker, štruktúrovaný hacker, organizovaný zločinec a priemyselná e-špionáž, insider, hacktivista, financovaná teroristická skupina, štát, skript kiddie, „hobby“ hacker
Účasť v sieti	Interný útočník, externý útočník
Schopnosť zmeniť stav siete	Pasívny útočník, aktívny útočník
Vývoj schopností	Statický útočník, adaptívny útočník
Rozsah informácií	Adaptívny útočník, deterministický útočník

3.2 Prístupy k identifikácii útočníkov podľa ich motivácie

3.2.1 Typológie útočníkov

Cieľom každého administrátora kritickej infraštruktúry by malo byť znížiť riziko cenovo čo najefektívnejšie namiesto úplného vylúčenia rizika [14]. Kategorizácia hrozieb je jednou zo stratégií manažmentu rizík. Buyens et al. [15] tvrdia, že jednou z najefektívnejších stratégií manažmentu rizík je vytvoriť **profily útočníkov**, ktoré zahŕňajú najmä úroveň ich schopností. Kategorizácia má niekoľko výhod [16]:

- umožňuje systematické štúdium bezpečnostných incidentov,

- pomáha manažérom budovať efektívne obranné systémy, ktoré sú menej zraniteľné,
- uľahčuje nahlasovanie incidentov kompetentným tímom.

Landreth [17] navrhol 5 typov útočníkov založených na ich schopnostiach a motiváciách. Medzi možné motivácie útočníka zaraďuje zámerné spôsobenie škody, intelektuálnu výzvu, vzrušenie, zvýšenie ega a kriminálny zisk. Chantler [18] kategorizuje útočníkov podľa motivácií, schopností a skúseností.

Ideálne by bolo rozlišovať útočníkov na základe viacerých parametrov ako:

- identita páchateľov a ich cieľ,
- metóda útoku a frekvencia výskytu,
- cieľ, ktorý chce útočník dosiahnuť a
- spôsobená škoda.

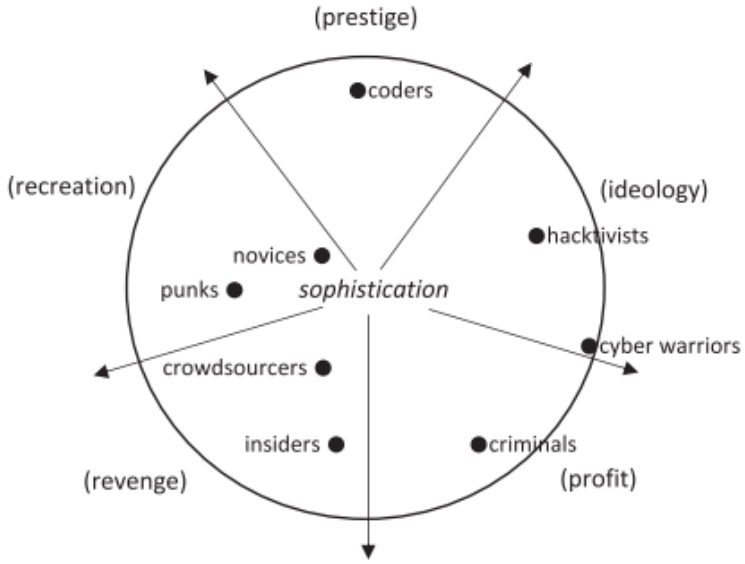
Keďže takéto údaje sú k dispozícii len zriedka, tak reálnejšie je klasifikovať útočníkov len podľa schopností a motivácií (pomsta, finančný zisk, zvedavosť, notoričnosť). V niektorých modeloch sa môžeme stretnúť aj s ideológiou ako jednou z foriem motivácie útočníka. Medzi ideologické motivácie môžeme zaradiť nacionalizmus alebo náboženstvo. Ako hekeri sú väčšinou označovaní politickí aktivisti. Rogers's [19] však upravil taxonómiu tak, že o hacktivistoch hovorí ako o útočníkoch, ktorí chcú predovšetkým získať notoričnosť a politické motivácie sú pre nich až druhoradé.

S rozvojom internetu sa postupne menilo aj zloženie jeho používateľov a s rôznorodými typmi používateľov sa začali objavovať aj rôznorodé škodlivé aktivity vďaka rozvíjajúcim sa schopnostiam útočníkov a ich motiváciám. Okrem hacktivistov sa v poslednom čase objavil aj ďalší typ útočníckej skupiny - **crowdsourcing.** Online crowdsourcing pozostáva z kolektívnej snahy vyriešiť nejaký problém väčšinou s využitím nelegálnej činnosti a dosiahnuť tým pochybné ciele. Tento typ činnosti zahŕňa takzvaný **doxing**, čo je využívanie internetových zdrojov (napríklad aj nabúranie sa do účtov na rôznych sociálnych sieťach) na získavanie osobných údajov o konkrétnych používateľoch.

Glenny [20] vo svojej práci zhrnul, že chápanie schopností a motivácií kybernetických útočníkov prispieva vo veľkej miere k vývoju bezpečnosti rôznych systémov, ktoré sú závislé na technických riešeniach.

3.2.2 Circumplex model

Kruhové modely sú používané na prezentovanie rôznych konceptov a vzťahov medzi nimi. Seebruck [21] vo svojom článku modifikoval tento model tak, že kruh je rozdelený do niekoľkých oblastí, kde každá z nich reprezentuje nejakú motiváciu. Dôležité je umiestnenie jednotlivých skupín útočníkov. Skupiny blízko vedľa seba sú si podobné, umiestnenie skupiny blízko okraja oblasti indikuje, že motivácie sa prekrývajú a pozície ďalej od centra naznačujú pokročilejšie schopnosti útočníka.



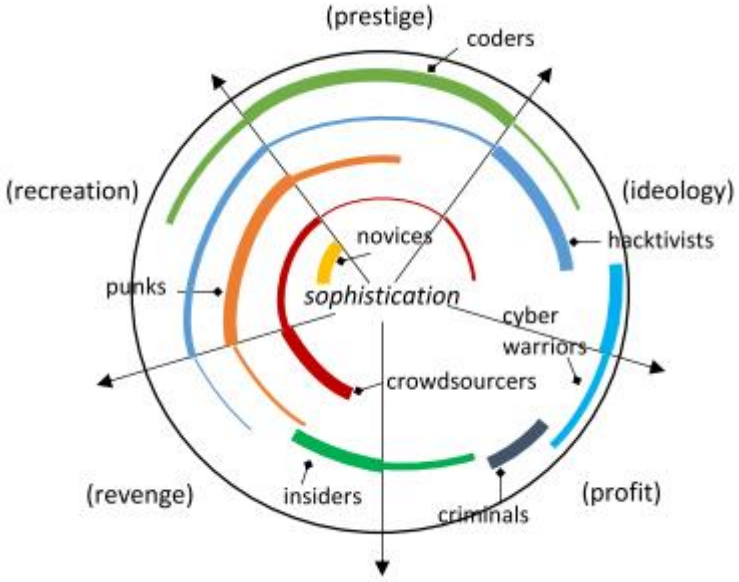
Obrázok 7 Kruhový model zobrazujúci motivácie útočníkov [21]

Takýto model je užitočný na zobrazenie vzťahov medzi útočníkovými schopnosťami a motiváciami, čo pomáha pri identifikácii a kategorizácii útočníkov. Seebruck teda navrhol tieto typy útočníkov a zoradil ich podľa narastajúcich technických zručností: **nováčikovia, crowdsourceri, pankáči, hacktivisti, insideri, kriminálnici a kybernetickí bojovníci.** Na obrázku 7 vidíme, že každá z piatich motivácií- ideológia, profit, pomsta, zábava, prestíž- má vlastnú oblasť.

Na obrázku 7 pozícia hacktivistov vyjadruje ich stredne pokročilé schopnosti, to, že ich motiváciou je určitá ideológia. Pozícia kybernetických bojovníkov indikuje vysoko sofistikovaných útočníkov motivovaných ideológiou a ziskom. Kriminálnici majú schopnosti na vyššej úrovni a primárne sú motivovaní ziskom, sekundárne pomstou.

3.2.3 Vážený circumplex model

I keď sú circumplex modely užitočné na zobrazovanie vzťahov medzi rôznymi typmi útočníkov, ich schopnosťami a motiváciami, neberú do úvahy napríklad to, ak má útočník nie jednu, ale niekoľko motivácií súčasne. Kvôli tomu bol navrhnutý vážený circumplex model, ktorý umožňuje zobraziť niekoľko motivácií útočníka naraz. Namiesto bodov v jednotlivých sektoroch sa využívajú krivky, ktoré môžu prechádzať cez viacero sektorov. To, ako veľmi bol útočník ovplyvnený daným typom motivácie znázorňuje hrúbka krivky.



Obrázok 8 Vážený kruhový model [21]

3.2.4 Využitie circumplex modelu

Tieto modely môžu byť využívané na odhaľovanie vzťahov, ako napríklad čierne trhy v oblasti profitu dosiahnuteľné cez anonymnú sieť **Tor (The Onion Router)**. Môžu byť použité na zobrazenie nepriamych vzťahov a na odhaľovanie komplexných vzťahov medzi hackermi a počítačovými systémami. Sú veľmi užitočným investigatívnym nástrojom vďaka ich rýchlej vizualizácii hrozieb.

4 Určenie typov útočníkov

V tejto kapitole rozoberáme prístupy hĺbkovej analýzy údajov k určeniu typu útočníkov. Následne sa venujeme len prístupom založeným na zhlukovaní. Poslednú časť kapitoly predstavuje popis k-means algoritmu a prístupov k určeniu vhodného k, teda počtu zhlukov (v našom prípade typov útočníkov).

4.1 Určenie typov útočníkov pomocou hĺbkovej analýzy údajov

Kybernetická bezpečnosť poskytuje rôzne technológie a procesy navrhnuté na ochranu počítačov, počítačových sietí, počítačových programov a údajov pred neautorizovaným prístupom, modifikáciou alebo poškodením. Na analýzu údajov z bezpečnostných systémov, ako sú systémy na detekciu prienikov (Intrusion Detection Systems, IDS) alebo honeypoty, existuje niekoľko prístupov z oblasti hĺbkovej analýzy údajov. Tie väčšinou pozostávajú zo štyroch fáz [25]:

- identifikácia atribútov jednotlivých inštancií,
- identifikácia atribútov potrebných pre klasifikáciu (redukcia dimenzie),
- „učenie“ modelu na tréningových údajoch,
- využívanie neučeného modelu pre klasifikáciu nových inštancií.

Nasledujúce podkapitoly popisujú metódy hĺbkovej analýzy údajov, ktoré je možné využiť pre vykonávanie rôznych analýz nad bezpečnostnými údajmi. Každý z nich je možné použiť aj na určovanie typov útočníkov tak, ako je popísané v týchto podkapitolách.

4.1.1 Umelé neurónové siete

Umelé neurónové siete sú inšpirované činnosťou mozgu a sú tvorené poprepájanými umelými neurónmi, ktoré sú schopné vykonávať výpočty na základe vstupných hodnôt. Vstupné údaje aktivujú neuróny na prvej vrstve siete, ktorej výstup je zároveň vstupom pre druhú vrstvu atď. Výstup poslednej vrstvy predstavuje výsledok celého výpočtu [25].

Umelé neurónové siete je možné použiť ako klasifikátory. Inými slovami výstupná vrstva generuje konečnú kategóriu, do ktorej je zaradený vstup. Ich

nevýhodou je však to, že pri veľkom množstve vstupných údajov je čas potrebný na naučenie siete veľmi dlhý. Napriek tomu je možné túto metódu využiť na detekciu zneužitia systému [38] alebo detekciu neobvyklej činnosti v systéme [39].

V prípade určovania typov útočníkov je možné tento prístup využiť tým spôsobom, že na vstupnej vrstve je prezentovaný útočník ako vektor hodnôt jeho atribútov. Ak sú pravidlá siete nastavené korektne a tak, aby zodpovedali požadovanej klasifikácii útočníkov do skupín, tak na výstupnej vrstve je možné priamo získať zaradenie útočníka do jednej z definovaných skupín.

4.1.2 Združovacie pravidlá a fuzzy združovacie pravidlá

Cieľom týchto prístupov je nájsť v údajoch nové, doteraz neznáme pravidlá. **Združovacie pravidlo** popisuje vzťahy medzi rôznymi atribútmi. Napríklad pravidlo IF (A AND B) THEN C vyjadruje, že ak sú prítomné atribúty A aj B, tak potom je prítomný aj atribút C pre danú inštanciu [25]. Podstatou tejto metódy je hľadanie množín objektov, ktoré sa často vyskytujú spolu (napr. {X, Y}) a následne generuje združovacie pravidlá (napr. X->Y a/alebo Y->X).

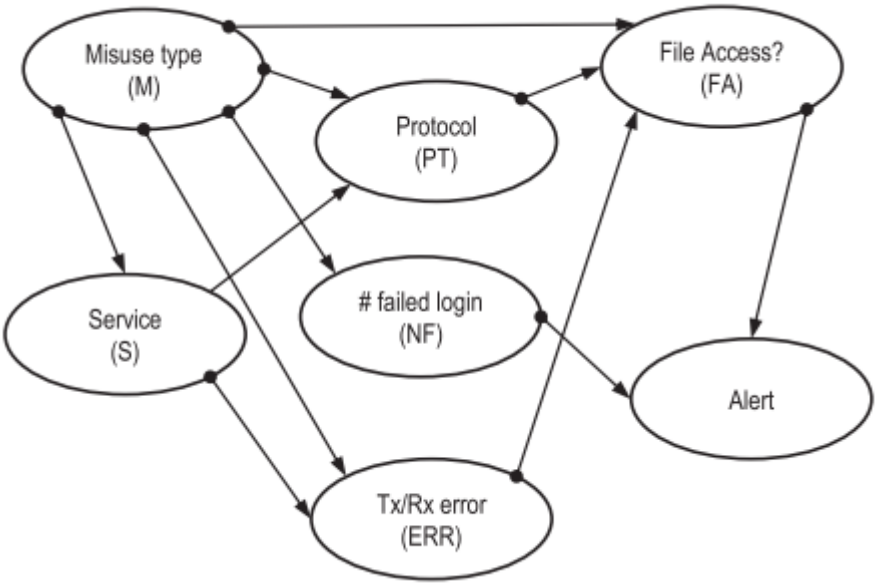
Nevýhodou tohto prístupu je to, že je možné ho aplikovať len na binárne údaje. Nie je ho možné použiť na numerické údaje, avšak tento nedostatok odstraňuje **fuzzy prístup** [40]. Napr. Brahmi [41] využil túto metódu na popísanie vzťahu medzi TCP/IP parametrami a typmi útokov.

Ak majú dvaja útočníci podobné atribúty, tak tento prístup by ich mal „združiť“. Je teda možné použiť túto metódu na hľadanie takých množín útočníkov, ktorí sú si podobní a tým pádom ich istým spôsobom klasifikovať.

4.1.3 Bayesova sieť

Bayesova sieť je pravdepodobnostný grafický model, ktorý reprezentuje premenné a vzťahy medzi nimi [42,43]. Sieť je tvorená uzlami predstavujúcimi diskrétne alebo spojité náhodné premenné, ktoré sú prepojené orientovanými hranami, vytvárajúc tak orientovaný acyklický graf. Bayesove siete sú väčšinou vytvárané s využitím poznatkov znalca v danej oblasti.

Na obrázku 9 je znázornená Bayesova sieť pre detekciu útokov. V závislosti od aplikácie môže byť využitá na popis vzájomného pôsobenia náhodných premenných alebo na výpočet pravdepodobného výstupu v koncovom stave.



Obrázok 9 Bayesova sieť [25]

Podobným spôsobom, aký je zobrazený na obr. 9 je možné klasifikovať aj útočníkov. Jednotlivé uzly by v tom prípade reprezentovali napr. konkrétne kroky útočníka. Na základe nich by bol potom útočník zaradený do niektorej zo skupín.

4.1.4 Rozhodovacie stromy

Rozhodovací strom je stromová štruktúra, v ktorej listy reprezentujú jednotlivé klasifikácie a vnútorné uzly vlastnosti, ktoré vedú ku konkrétnej klasifikácii [25]. Konkrétna inštancia je zaradená do niektorej triedy podľa toho, či spĺňa alebo nespĺňa atribúty vo vnútorných uzloch.

Najznámejšie metódy na automatické vybudovanie rozhodovacieho stromu sú **ID3 algoritmus** [44] a **C4.5 algoritmus** [45]. Oba tieto algoritmy budujú rozhodovacie stromy nad tréningovou vzorkou údajov s využitím konceptu informačnej entropie. Výhodou rozhodovacích stromov je vysoká presnosť klasifikácie, intuitívne používanie a jednoduchá implementácia.

Na detekciu zneužitia systému je možné rozhodovacie stromy využiť tým spôsobom, že každý vstup sa porovná so všetkými pravidlami. Ak budú teda súčasťou

stromu pravidlá na rozhodnutie o type útočníka, tak je možné tento prístup využiť aj na identifikáciu typov útočníkov.

4.1.5 Evolučné výpočty

Pojem **evolučný výpočet** v sebe zahŕňa genetické algoritmy, genetické programovanie, evolučné stratégie a iné postupy [25]. V článku [25] sa autori zameriavajú najmä na genetické algoritmy a programovanie, ktoré sú založené na princípe prežitia zdravšieho, odolnejšieho jedinca. Väčšinou sa tieto algoritmy začínajú s náhodnou populáciou, ktorej jedince sú reprezentované ako bitové reťazce (v genetických algoritmoch) alebo programy (v genetickom programovaní).

Evolučné výpočty, konkrétne genetické algoritmy, využil Khan [46] na vývoj pravidiel pre detekciu vniknutia do systému.

4.1.6 Zhlukovanie

Zhlukovanie je technika dolovania v údajoch, ktorú sme sa rozhodli použiť v našej práci na hľadanie skupín útočníkov v údajoch. Hlavným dôvodom je to, že tento prístup nevyžaduje žiadne definovanie pravidiel ani žiadnu špeciálnu úpravu údajov. Zhlukovanie bližšie popisujeme v nasledujúcej kapitole.

4.2 Určenie typov útočníkov pomocou zhlukovania (clustering)

Zhlukovanie (clustering) je množina algoritmov na hľadanie vzorov vo vysoko-rozmerných neklasifikovaných údajoch [25]. Ide o prístup bez „učiteľa“, v ktorom sú údaje zoskupované na základe podobnosti [25]. Hlavnou výhodou využívania zhlukovacích techník v kyberbezpečnosti, resp. pri odhaľovaní neautorizovaného vniknutia do systému, je schopnosť učenia sa z údajov bez toho, aby systémový administrátor musel poskytnúť explicitný popis rôznych tried útokov.

4.2.1 Porovnanie zhlukovacích algoritmov

Rôzne techniky zhlukovania sú vhodné pre rôzne typy bezpečnostných údajov. V modeloch založených na prepojení rôznych konceptov sú dátové body spájané na základe vzájomných vzdialeností. Pre tieto modely je najvhodnejším zhlukovacím

prístupom **hierarchické zhlukovanie** [61]. Medzi algoritmy pre hierarchické konceptuálne zhlukovanie patrí aj algoritmus **COBWEB** [68]. COBWEB inkrementálne organizuje inštancie do tzv. klasifikačného stromu, ktorý môže byť použitý napr. na predikciu chýbajúcich hodnôt. V modeloch, kde je výhodné každú skupinu reprezentovať podľa jej **centroidu**, teda stredového vektora, je vhodné použiť **k-means algoritmus** [25].

V distribučných modeloch jednotlivé skupiny zodpovedajú štatistickému rozdeleniu. Pre tento typ údajov je vhodný zhlukovací algoritmus **Expectation Maximization (EM)** [62]. Takzvaný **DBSCAN (Density-Based Spatial Clustering of Applications with Noise)** [63] zhlukovací algoritmus je určený pre modely, kde sú údaje zoskupované do hustých a navzájom prepojených oblastí. Podobným algoritmom je aj algoritmus **OPTICS** [67], ktorý sa od DBSCAN algoritmu líši len tým, že sa neurčuje príslušnosť inštancií do zhlukov, ale iba sa ukladá poradie v akom boli inštancie spracované. Grafové modely definujú každý zhluk ako množinu prepojených uzlov, kde každý uzol je spojený prostredníctvom hrany aspoň s jedným iným uzlom v množine. Pre zhlukovanie v týchto modeloch je vhodným prístupom **clique algoritmus** [64].

Ďalším populárnym prístupom k zhlukovaniu bezpečnostných údajov je **k-NN (k nearest neighbours)** algoritmus [65]. Výhodou tohto algoritmu je jednoduchosť, avšak vysoko-rozmerné údaje majú na tento prístup negatívny vplyv [25].

4.2.2 Zhlukovacie algoritmy implementované v nástroji Weka

V práci využívame nástroj Weka [47], ktorý okrem iného obsahuje aj rôzne zhlukovacie algoritmy určené pre dolovanie v údajoch. Jednotlivé algoritmy sme aplikovali na naše údaje vo formáte IDEA. Bližšie sa týmto údajom budeme venovať v kapitole 5.1. Nasledujúca tabuľka zobrazuje výsledok použitia na testovacom datasete.

Tabuľka 3 Nevýhody zhlukovacích algoritmov aplikovaných na údaje z Wardenu

Zhlukovací algoritmus	Testovací dataset
Cobweb [66]	Príliš veľa nájdených zhlukov (rádovo v stovkách)
DBSCAN [63]	Nie všetky inštancie zaradené do zhlukov
EM [62]	Veľmi dlhý čas výpočtu, málo nájdených zhlukov
HierarchicalClusterer [61]	Žiadne nájdené zhluky
OPTICS [67]	Veľmi dlhý čas výpočtu
K-means [57]	Prijateľný čas výpočtu aj počet zhlukov

Keďže pre účely našej práce potrebujeme pracovať so stredovými vektormi jednotlivých zhlukov a naše údaje sú tvorené až 40-rozmernými vektormi, vhodným algoritmom sa javí k-means algoritmus. Ako je možné vidieť z predchádzajúcej tabuľky 3, k-means algoritmus najlepšie pracuje s našim datasetom, ktorý zohľadňuje údaje o útokoch.

4.3 K-means algoritmus a určenie parametra k

***K-means* algoritmus** rozdeľuje danú množinu údajových bodov X do *k* zhlukov, pričom každý bod je viac podobný centroidu zhluku, do ktorého bol zaradený, ako akémukoľvek inému centroidu [57]. Vo všeobecnosti *k-means* algoritmus pozostáva z nasledujúcich krokov:

- **krok 1:** Vyber *k* počiatočných centroidov c_1, c_2, ..., c_k.
- **krok 2:** Každú inštanciu x z X priraď do zhluku, ktorého centroid je najbližšie k x.
- **krok 3:** Pre každý zhluk prepočítaj jeho centroid podľa prvkov, ktoré doň boli zaradené.
- **krok 4:** Opakuj kroky 2 a 3 pokiaľ nebude dosiahnutá konvergencia.

Pri implementácii k-means algoritmu je potrebné vopred určiť k, teda počet zhlukov a tiež metriku pre určovanie vzdialenosti. V k-means zhlukovaní sa najčastejšie používa **Euklidovská vzdialenosť**. Ak **p**=(p1,p2,...,pn) a **q**=(q1,q2,...,qn) sú dva body euklidovského n-rozmerného priestoru, tak vzdialenosť medzi nimi je daná vzťahom:

$$d(\mathbf{p},\mathbf{q}) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}.$$

Výber parametra k je daný empiricky. Na určenie vhodnej konštanty „k“ pre k-means zhlukovanie existuje niekoľko možných prístupov, ktoré sú popísané v nasledujúcich podkapitolách.

4.3.1 X-means zhlukovanie

V štatistike a dolovaní v údajoch je **X-means** zhlukovanie obmenou k-means zhlukovania, kde sa zjemňuje priradenie do zhlukov tým, že sa postupne skúša rozdeľovanie aktuálnych zhlukov. Ukladajú sa najlepšie získané výsledky, teda rozdelenia, až pokiaľ nie je splnená nejaká podmienka, napr. Bayesovo informačné kritérium (BIC) alebo Akaikeho informačné kritérium (AIC) [58].

Po aplikácii tohto prístupu na naše údaje s využitím nástroja Weka bolo k určené na hodnotu štyri. Avšak rozdelenie útočníkov do skupín bolo príliš nerovnomerné (v jednej zo skupín až 85% všetkých útočníkov), čo je pre našu analýzu nevyhovujúce.

4.3.2 Siluetová metóda (Silhouette method)

Priemerná silueta údajov je užitočné kritérium pre určovanie prirodzeného počtu zhlukov. Pod **siluetou inštancie** rozumieme, ako „blízko“ je k inštanciám nachádzajúcim sa v rovnakom zhluku a ako ďaleko je od inštancií susedného zhluku, čiže od zhluku, ktorého priemerná vzdialenosť od danej inštancie je najmenšia [59]. Hodnota siluety blízka k 1 implikuje, že inštancia je v správnom zhluku, kým silueta blízka -1 implikuje, že inštancia je v nesprávnom zhluku. Optimalizačné techniky ako

napr. genetické algoritmy sú vhodné pre určenie počtu zhlukov tak, aby silueta jednotlivých inštancií bola čo najvyššia [59].

4.3.3 Krížová validácia (Cross validation)

Krížová validácia (cross validation) [60] je ďalšia možnosť, ako určiť čo najvhodnejší počet zhlukov. Jej podstata spočíva v rozdelení údajov na v častí. Každá z nich je následne „odložená“ bokom ako testovacia vzorka. Na zvyšných v-1 častiach je natrénovaný model a na testovacej vzorke je vypočítaná hodnota nejakej účelovej funkcie, napr. súčet druhých mocnín vzdialeností k centroidom jednotlivých zhlukov. Týchto v hodnôt je spriemerovaných pre každý alternatívny počet častí v a počet zhlukov je následne určený tak, že ďalšie zvýšenie počtu častí v by viedlo len k nepatrnej zmene v hodnote účelovej funkcie [60].

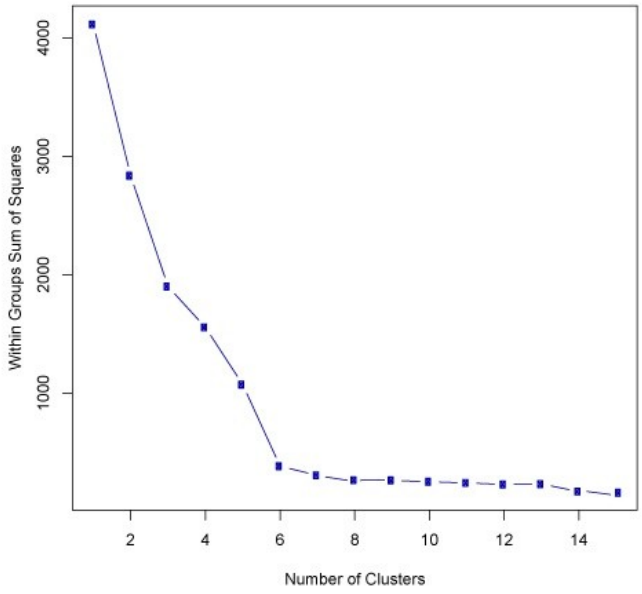
4.3.4 Elbow metóda

Na určenie vhodnej konštanty „k“ pre k-means zhlukovanie na predspracovaných údajoch sme sa rozhodli využiť tzv. „elbow“ metódu [35]. Jej hlavná myšlienka spočíva v spustení k-means algoritmu určitý počet krát a pre každé „k“ vypočítať tzv. „sum of squared errors“ (SSE) podľa vzorca:

$$SSE = \sum_{i=1}^{K} \sum_{x \in c_i} dist(x, c_i)^2$$

.

SSE je teda definované ako súčet vzdialeností umocnených na druhú medzi každým prvkom klastra a stredom príslušného klastra. Zväčšovaním „k“ sa SSE zmenšuje ako je vidno na nasledujúcom obrázku:



Obrázok 10 Graf na určenie "k" pre k-means zhlukovanie

Dôvodom je to, že s narastajúcim počtom zhlukov sa tieto zhluky zmenšujú, a teda aj skreslenie sa zmenšuje. Cieľom „elbow“ metódy je zvoliť také „k“, pre ktoré hodnota SSE náhle poklesne. Pre prípad na obrázku 10 by táto metóda ako „k“ zvolila 6, pretože pre k=7 a vyššie je zmena v SSE už len nepatrná.

V našej práci pre určenie k využívame túto metódu.

Výhodou tejto metódy je:

- jednoduchá implementácia,
- časová zložitosť (O(n^{dk+1} * log n), kde k je počet klastrov, d je dimenzia údajov a n je počet entít, ktoré sú na vstupe algoritmu.

Naopak, nevýhodou metódy je nutnosť určenia počtu zhlukov pred samotným zhlukovaním. Nevýhodu tiež predstavuje skutočnosť, že výsledné zhluky závisia od počiatočnej voľby centroidov.

5 Systém na identifikáciu útočníkov pomocou k-means algoritmu

Táto kapitola popisuje systém na identifikáciu útočníkov do skupín. V prvom rade je popísaný datatset , ktorý máme k dispozícii a spôsob, akým boli údaje spracovávané. Bližšie sa venujeme špeciálnemu formátu, v ktorom tieto údaje sú. V kapitole navrhujeme vlastný spôsob založený na využití zhlukovacieho algoritmu k-means, ktorý nájde v údajoch zhluky, teda skupiny útočníkov. V závere kapitoly je uvedená analýza získaných výsledkov.

5.1 Dataset

Pre účely diplomovej práce máme k dispozícii 2-týždňové údaje zachytené v rámci projektu WARDEN [56] v počítačovej sieti CESNET (združenie vysokých škôl a Akadémie vied Českej republiky, ktoré prevádzkuje a rozvíja národnú e-infraštruktúru pre vedu, výskum a vzdelávanie) [34]. Tento dataset obsahuje najmä údaje zachytené honeypotmi, resp. inými podobnými detekčnými systémami. Dôležitým aspektom je skutočnosť, že tieto údaje predstavujú zachytené správanie sa útočníkov. Teda nie je potrebné určiť, ktoré údaje predstavujú legitímnu prevádzku a ktoré nie. Dataset obsahuje približne **72 miliónov** záznamov, pričom každý riadok je vo formáte IDEA [31]. Prakticky ide o upravený JSON formát. IDEA je deskriptívny dátový model, pre ktorý je typický key:value formát. Okrem toho má definovanú maximálnu hĺbku vnorenia 2.

Aby sa nám s údajmi lepšie pracovalo, sú uložené v PostgreSQL databáze [33]. Tabuľka pozostáva z dvoch stĺpcov: ID a ideadata. Záznamy v stĺpci ideadata sú priamo v IDEA formáte, ale vďaka PostgreSQL vieme efektívne pracovať aj s JSON objektami.

Pomocou nasledujúcich dopytov sme z údajov, ktoré máme k dispozícii získali všetky kľúče, ktoré sa v našich údajoch vyskytujú na prvej úrovni IDEA formátu a na druhej úrovni pre kľúč „Source“ a „Target“, teda získali sme informácie o zdroji útoku a jeho cieli:

select distinct jsonb_object_keys(ideadata) from ideas2;

select distinct jsonb_object_keys(ideadata->'Source'->0) from ideas2;

select distinct jsonb_object_keys(ideadata->'Target'->0) from ideas2;

Z výsledkov týchto dopytov, teda z dostupných údajov o útoku vo formáte IDEA budeme uvažovať nasledovné:

- kategóriu (category),
- trvanie útoku (duration), resp. rozdiel medzi CeaseTime a EventTime,
- zdroj (source) vo formáte IPv4,
- cieľ (target) vo formáte IP4, Proto aPort.

Okrem týchto údajov budeme pre každého útočníka brať do úvahy aj **počet rôznych cieľov** teda rôznych IP adries, na ktoré útočník útočil a tiež **maximálny časový rozdiel** medzi útokmi daného útočníka. Okrem toho budeme na základe cieľovej IP adresy brať do úvahy aj **počet rôznych počítačových sietí**, na ktoré útočník útočil.

5.1.1 IDEA formát

Administrátori prevádzkujú rôzne detekčné systémy, ktoré generujú obrovské množstvá údajov každý deň, ktoré však väčšinou zostávajú uložené lokálne na serveroch. Administrátori z nich filtrujú len určitú časť potrebných údajov a zvyšok je nevyužitý. Avšak tieto na prvý pohľad zbytočné údaje môžu byť v kombinácii s údajmi z iných zdrojov užitočné. Za posledné roky sa objavilo niekoľko systémov na takéto automatické výmeny reportov o bezpečnostných incidentoch ako napríklad Warden [56], Megatron, n6 alebo Prelude.

5.1.2 Popis údajov IDEA formátu využívaných v práci

Kategorizácia daného útoku je dôležitým krokom pri vytváraní reportu. Pre klasifikáciu bezpečnostného incidentu je využívaný zoznam názvov kategórií, pričom sa v reporte uvádza aj **kategória (category)** aj **podkategória (subcategory)** oddelené bodkou (napríklad "Abusive.Spam", "Fraud.Phishing"). V prípade, že podkategória nie je jednoznačne určiteľná, je možné uviesť len kategóriu. Ak žiadna z kategórií v zozname nie je použiteľná, tak tvorca reportu môže použiť novú, vlastnú kategóriu. Ak nie je možné kategorizovať daný útok, tak je možné použiť „Other". V názvoch kategórií by nemali byť použité číslice, podčiarkovníky ani pomĺčky.

Trvanie útoku (duration) je reťazec reprezentujúci rozdiel medzi dvoma časovými pečiatkami.

IP adresa (Source.IP4): predstavuje IP adresu vo verzii 4, z ktorej bol útok zaznamenaný.

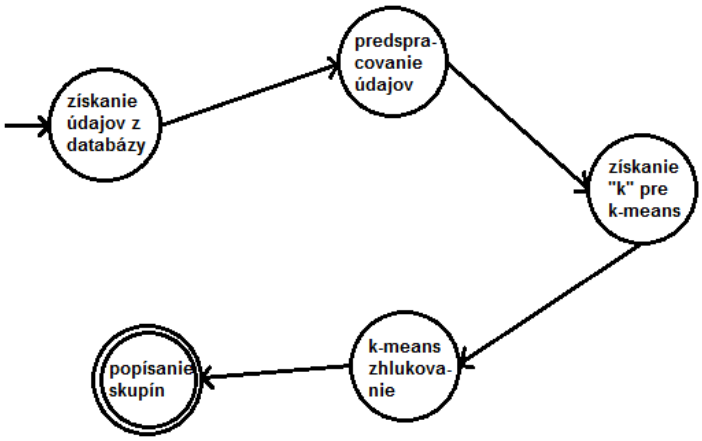
Protokol (Target.Proto) je uvedený v časti "Source/Proto" alebo "Target/Proto". Protokoly sa uvádzajú od najnižšieho po najvyšší podľa ISO/OSI modelu.

Port (Target.Port): označuje sieťový port, ktorý je využitý pri útoku.

5.2 Návrh systému

Cieľom tejto práce je určiť, aké typy útočníkov sa nachádzajú v údajoch, ktoré máme k dispozícii. To znamená, že v prvom rade je potrebné nájsť skupiny útočníkov a následne popísať vlastnosti týchto skupín.

Na dosiahnutie tohto cieľa sme sa rozhodli využiť metódu hĺbkovej analýzy údajov (data mining) – zhlukovanie (clustering). Dôvod výberu tejto metódy aj konkrétneho algoritmu je popísaný v predchádzajúcej kapitole. Nasledujúci diagram zobrazuje jednotlivé stavy, v ktorých sa náš navrhovaný systém bude nachádzať počas procesu hľadania skupín útočníkov v údajoch.



Obrázok 11 Diagram zobrazujúci návrh riešenia

Jednotlivé záznamy, ktoré sa nachádzajú v údajoch z Warden-u, predstavujú jednotlivé útoky. Keďže sa však naša práca zaoberá útočníkmi, nie útokmi, tak vo fáze predspracovania údajov vytvárame vektory, ktorými je každý útočník reprezentovaný. Kľúčovou informáciou pre tento krok je zdrojová IP adresa, t.j. IP adresa, z ktorej útočník útočil. Každý útočník na vstupe bude reprezentovaný práve jedným vektorom,

kde jednotlivé zložky sú hodnoty atribútov, ktorými je útočník popísaný. Všetky tieto atribúty a aj ich presné poradie vo vektore je uvedené v prílohe B.

Pre každú IP adresu (Source.IP4) bude vo vstupnom súbore jeden záznam-riadok, ktorý bude obsahovať nasledujúce informácie:

- **IP** – IP adresa útočníka,
- **Category** – pre každú kategóriu útoku bude uvedený počet vykonania daného typu útoku daným útočníkom,
- **Duration** – celkové trvanie všetkých útokov daného útočníka,
- **MaxIdleness** – maximálny rozdiel medzi dvoma za sebou idúcimi útokmi daného útočníka,
- **Proto** – pre každý protokol, ktorý sa v údajoch vyskytuje bude uvedené, koľkokrát daný útočník využil daný protokol pri svojich útokoch,
- **Port** – koľkokrát bol použitý port z intervalu 0-1023 a z intervalu 1024-65535,
- **Počet rôznych cieľov** – počet zariadení, na ktoré bol od daného útočníka zaznamenaný útok,
- **ISP** – počet rôznych sietí, na ktoré útočník útočil.

Následne sme kvôli lepším výsledkom všetky hodnoty všetkých atribútov normalizovali do intervalu <0, 1>. To znamená, že pre každý stĺpec, resp. atribút sme našli minimálnu hodnotu a maximálnu hodnotu a aplikovali nasledovný vzorec pre normalizáciu:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Na takto pripravenú množinu útočníkov, teda vektorov, aplikujeme zhlukovací algoritmus, ktorý každého útočníka zaradí do nejakej skupiny na základe podobnosti s ostatnými, už zaradenými, útočníkmi.

Každú skupinu je následne potrebné popísať, t.j. analyzovať vlastnosti útočníkov danej skupiny. Aby bolo možné zrealizovať tento cieľ, každá skupina bude reprezentovaná jedným vektorom, a to **stredovým vektorom** (**centroidom**) určeným ako stred všetkých vektorov, ktoré boli do danej skupiny zaradené. Tento **„ideálny“ útočník** danej skupiny má hodnoty jednotlivých atribútov, na základe ktorých popíšeme celú skupinu.

5.3 Implementácia systému

Pre implementáciu praktickej časti našej práce sme sa rozhodli využiť programovací jazyk JAVA [49]. Hlavným dôvodom bolo to, že v práci využívame pre účely zhlukovania nástroj Weka [47], ktorého algoritmy je možné využívať priamo z JAVA kódu. V rámci systému vyžívame nasledujúce technológie:**PostgreSQL** [50] – open-source objektovo-relačný databázový systém vhodný na prácu s JSON formátom.

- **PostgreSQL JDBC** [51] – ovládač umožňujúci JAVA programom pripojiť sa k PostgreSQL databáze.
- **Weka** [47] – softvér napísaný v jazyku JAVA určený na dolovanie v údajoch.
- **JFreeChart** [52] – knižnica na zobrazovanie grafov napísaná v jazyku JAVA.
- **Swing** [53] – GUI (Graphical User Interface) nástroj pre programovací jazyk JAVA.
- **IP-API** [48] – poskytuje prístup ku geografickým informáciám súvisiacim s IP adresami.

V nasledujúcich podkapitolách sa budeme venovať implementácii našej aplikácie postupne od získavania údajov z databázy, ich predspracovania, získavania parametra k pre zhlukovanie až po samotné zhlukovanie a získavanie výsledkov.

5.3.1 Príprava údajov

Údaje, s ktorými pracujeme, sú uložené v PostgreSQL databáze [50]. Tabuľka obsahuje dva stĺpce: id a stĺpec obsahujúci údaje v IDEA formáte. Pre PostgreSQL sme sa rozhodli z dôvodu, že IDEA formát je v konečnom dôsledku JSON formát, s ktorým sa pomocou PostgreSQL dá veľmi efektívne pracovať. Pomocou PostgreSQL dopytov sa dá pristupovať k jednotlivým zložkám JSON formátu, a teda nie je potrebné iné manuálne spracovávanie textových reťazcov.

Avšak kvôli jednoduchšej manipulácii s len potrebnými údajmi sme vytvorili pomocnú tabuľku so stĺpcami:

- **id**,
- **ip** (zdrojová IP adresa),
- **iptarget** (cieľová IP adresa),

- **category** (typ útoku),
- **countcategory** (počet útokov typu category z danej IP adresy),
- **protocol** (protokol využitý pri útoku),
- **countproto** (koľkokrát bol daný protokol pri danom útoku použitý),
- **port** (cieľový port),
- **duration** (trvanie útoku v sekundách),
- **start** (čas začiatku útoku),
- **end** (čas konca útoku),
- **isp** (poskytovateľ internetového pripojenia pre zariadenie, na ktoré bol zaznamenaný útok).

Všetky tieto hodnoty okrem isp sme získali a vložili do tabuľky pomocou nasledujúceho dopytu:

create table abc as select row_number() over (order by null) as id, q. from (select s.IP::jsonb::text, s.IPtarget::jsonb::text, s.category::jsonb::text, s.countCategory, s.protocol::jsonb::text, s.countProto, s.port::jsonb::text, extract('epoch' from age(s.end, s.start)) as duration, s.start, s.end from (select t.IP, t.IPtarget,t.category,t.countCategory,t.protocol,t.countProto,t.port,to_timestamp(substr ing(replace(replace(t.start::jsonb::text, 'T', ' '), 'Z', ' ') from 2 for 19) , 'YYYY-MM-DD HH24:MI:SS') as start, to_timestamp(substring(replace(replace(t.end::jsonb::text, 'T', ' '), 'Z', ' ') from 2 for 19) , 'YYYY-MM-DD HH24:MI:SS') as end from (select ideadata->'Source'->0->'IP4' as IP, ideadata->'Target'->0->'IP4' as IPtarget, ideadata->'Category' as category, count(ideadata->'Category') as countCategory, ideadata->'Target'->0->'Proto' as protocol, count(ideadata->'Target'->0->'Proto') as countProto, ideadata->'Target'->0->'Port' as port, ideadata->'EventTime' as start, ideadata->'CeaseTime' as end from ideas2 group by ideadata->'Category', ideadata->'Target'->0->'Proto', ideadata->'Source'->0->'IP4', ideadata->'Target'->0->'IP4', ideadata->'Target'->0->'Port', ideadata->'EventTime', ideadata->'CeaseTime' order by ideadata->'Source'->0->'IP4', ideadata->'EventTime') as t where (t.protocol is not null or t.port is not null) and t.category::jsonb::text not like '%Test%' and t.start is not null and t.end is not null and t.IPtarget is not null and t.IP is not null) as s*

Do tejto tabuľky sme pridali stĺpec ISP, a to pomocou PHP skriptu, kde sme využili nástroj IP-API [48]. Ten na základe cieľovej IP adresy zistí názov poskytovateľa

internetového pripojenia. Tento názov následne pomocou UPDATE príkazu vložíme do tabuľky. Tieto kroky vykonáme pre každý riadok tabuľky. Skript je uvedený v prílohe D.

5.3.2 Získavanie údajov a ich predspracovanie

Naša aplikácia umožňuje pracovať s údajmi v dvoch módoch: online a offline móde. Rozdiel medzi nimi je v tom, že v online prístupe sa údaje o útočníkoch získavajú priamo z databázy, teda používateľ musí zadať, na akom serveri sa databáza nachádza a aký je jej názov. Súčasne musí zadať názov tabuľky, s ktorou bude pracovať a napokon musí uviesť prihlasovacie údaje. Všetky tieto používateľom zadané parametre využíva **PostgreSQL JDBC ovládač** na pripojenie k PostgreSQL databáze.

Na získanie potrebných atribútov popisujúcich jednotlivé útoky (zatiaľ nie jednotlivých útočníkov, pretože v tabuľke máme záznam pre každý útok, pričom jeden útočník mohol vykonať viac útokov) sa na základe vytvoreného pripojenia a metódy **executeQuery()** zavolanej na inštancii triedy **Statement** vykoná dopyt:

"*SELECT current.id as id,current.IP as ip, current.IPtarget as iptarget, current.category as category, current.countCategory as countCategory, current.protocol as protocol, current.countProto as countProto, current.port as port, extract('epoch' from age(current.end,current.start)) as duration, case when next.ip != current.ip then NULL else extract('epoch' from age(next.start, current.end)) end as idleness, current.isp as isp FROM "+this.table+" AS current LEFT JOIN "+this.table+" AS next ON next.id = (SELECT MIN(id) FROM abc WHERE id > current.id) order by current.ip, current.isp*",

kde **table** je inštančná premenná, ktorá obsahuje používateľom zadaný názov tabuľky. Takto získané údaje sa nasledovne upravujú, t.j. odstraňujú sa nepotrebné symboly, ako napríklad „[", „]", ktoré sú súčasťou IDEA formátu, v ktorom boli údaje pôvodne uložené.

Každý útok (resp. neskôr útočníka) budeme reprezentovať ako 40-rozmerný vektor. Ako si však môžeme všimnúť v predchádzajúcom dopyte, priamo z tabuľky toľko atribútov nevyberáme. V tomto kroku ale atribút category, atribút protocol a atribút port rozpisujeme do niekoľkých zložiek tak, že pre každú hodnotu atribútu category, protocol a port uvedieme počet výskytov daného atribútu v útokoch

konkrétneho útočníka. Na tento účel sme využili metódy triedy **java.util.Scanner** [54] a **java.io.PrintWriter** [55]. Nevyhnutnosťou sú podmieňujúce príkazy (if), ktoré na základe hodnoty atribútu rozhodli, akým vektorom bude daný atribút reprezentovaný.

Ďalší krok pri predspracovaní údajov je reprezentácia každého útočníka práve jedným záznamom, teda vektorom. Na to, aby sme jednotlivé vektory zlúčili, sme využili informáciu o zdrojovej IP adrese, ktorou je každý útočník reprezentovaný. Pre rovnaké IP adresy sme hodnoty jednotlivých atribútov, resp. zložiek vektorov sčítali, s výnimkou atribútu určujúceho maximálny čas medzi dvoma útokmi. V tomto prípade sme do výsledného vektora vzali maximálnu hodnotu, teda pre daného útočníka máme informáciu o tom, aký najdlhší čas mu trvalo, kým sa vrátil a znova vykonal útok. Toto je možné využiť na **rozlíšenie medzi automatickým a manuálnym útokom**. Súčasťou vektora je tiež informácia o poskytovateľovi internetového pripojenia(Internet Service Provider, ISP), na základe ktorej sme do výsledného vektora zaznamenali informáciu o tom, na koľko rôznych počítačových sietí sa útočník zameral.

V ďalšom kroku sme ku každému útočníkovi pridali informáciu o počte rôznych cieľov (teda IP adries), na ktoré daný útočník útočil. Opäť sme využili PostgreSQL JDBC a vykonali nasledujúci dopyt:

"*select t.ip as ip, count(t.ip) as count from (select distinct ip, iptarget from "+this.table+") as t group by t.ip order by t.ip*"

ip	count
["101.215.239.44"]	1
["103.5.140.189"]	1
["103.66.147.42"]	1
["103.7.28.92"]	52
["103.7.30.237"]	62
["104.152.52.55"]	1
["104.152.52.61"]	1
["104.156.228.190"]	1
["104.156.240.140"]	4
["104.156.240.214"]	1
["104.197.197.61"]	1
["104.200.151.16"]	3
["104.214.149.179"]	1
["105.100.11.24"]	1

Obrázok 12 Ukážka výstupu predchádzajúceho dopytu

Výsledok tohto dopytu sme zapísali do súboru a následne sme jednotlivé hodnoty doplnili do súboru s vektormi reprezentujúcimi útočníkov na základe zdrojových IP adries.

Aby bolo možné nad doteraz získanými a upravenými údajmi spustiť zhlukovací algoritmus, bolo potrebné z každého záznamu reprezentujúceho útočníka odstrániť IP adresu, na čo sme využil opäť Scanner a metódy triedy String. Odstránenie IP adresy súčasne predstavuje anonymizačný aspekt.

Posledným krokom predspracovania údajov je **normalizácia**. Využili sme **„min-max“ normalizáciu**, ktorá všetky hodnoty všetkých vektorov zobrazila do intervalu [0,1]. Tento krok, na rozdiel od tých predchádzajúcich, sa vykonáva aj v offline móde.

Offline mód spočíva vo využití už vopred pripraveného súboru obsahujúceho údaje o útočníkovi tak, ako je to uvedené v používateľskej príručke (príloha C). Výhodou tohto prístupu je, že nie je potrebné internetové pripojenie a zároveň čas potrebný na získanie výsledkov je omnoho kratší (asi trinástina trvania online prístupu). Po normalizácii hodnôt vo vstupnom súbore (alebo bez normalizácie ak používateľ na vstup dal už predspracovaný súbor vo formáte .arff) je možné prejsť priamo k výberu vhodného „k“ a k samotnému zhlukovaniu.

5.3.3 Implementácia „elbow“ metódy

Pri implementácii „elbow“ metódy sme využili v prvom rade nástroj Weka. Jej podstata spočíva v tom, že sa „skúšajú“ rôzne hodnoty „k“ (od 1 po nejaké zvolené n), pre každé „k“ sa vykoná zhlukovanie, a následne sa počíta chyba (SSE – sum of squared errors). Všetky tieto hodnoty SSE pre „k“ od 1 po 20 sme uložili do poľa typu double. Hodnotu n=20 sme zvolili na základe toho, že je dostatočne veľká na to, aby bolo možné vidieť, že hodnoty SSE sa už nemenia, resp. menia sa len veľmi málo. (Je možné zvoliť aj akúkoľvek väčšiu hodnotu, resp. menšiu, ale postačujúcu).

Následne sme využili **JFreeChart knižnicu** [52], ktorá hodnoty z tohto poľa zobrazila v grafe závislosti SSE od „k“. Keďže „elbow“ metóda je empirická metóda, tak je na používateľovi, akú hodnotu „k“ zvolí. Vykreslený graf nám napovie, ktoré „k“ je pre dané údaje najvhodnejšie.

5.3.4 Aplikácia k-means algoritmu na údaje a popis výsledkov

Na predspracované údaje sa aplikuje **k-means algoritmus**, pričom ako parameter „k“ sa použije hodnota zadaná používateľom. Na inštancii triedy

SimpleKMeans, ktorá je súčasťou knižnice Weka, sa zavolá metóda, ktorá nastaví „k“, teda výsledný počet zhlukov. Načítajú sa údaje z predpripraveného súboru a vybuduje sa nad nimi tzv. **Clusterer**, t.j. každá inštancia, teda útočník, bude zaradený do niektorej z „k“ skupín. Pre každú skupinu umožňuje Weka získať centroid, t.j. stredový vektor, ktorý bude v našom prípade reprezentovať „ideálneho“ útočníka danej skupiny. Avšak zložky týchto stredových vektorov sú normalizované hodnoty. Keďže pre ďalšiu analýzu je nevyhnutné poznať reálne, t.j. nie normalizované hodnoty, na každý centroid sme následne aplikovali metódu **denormalize(),** ktorá v pôvodných údajoch pre každú zložku nájde minimálnu aj maximálnu hodnotu a danú normalizovanú hodnotu „denormalizuje“.

V tomto okamihu máme teda „k“ „ideálnych“ útočníkov reprezentujúcich „k“ skupín, pričom každý z nich je daný ako 40-rozmerný vektor. Keďže takáto reprezentácia používateľovi veľa nenapovie, v nasledujúcom kroku sme každý z týchto „k“ vektorov rozpísali do čitateľnejšej podoby. Pre každú nenulovú zložku vektora sme na základe jej poradia vo vektore určili, o ktorý atribút ide a do výpisu sme uviedli názov atribútu a jeho hodnotu. Na základe tohto výpisu si používateľ môže sám vytvoriť predstavu o tom, aké vlastnosti majú útočníci, ktorí sa v jeho údajoch nachádzajú.

Podrobnejšia analýza získaných skupín útočníkov na údajoch zo systému Warden je popísaná v kapitole 5.5.

5.4 Zhlukovanie a určenie „ideálneho útočníka“

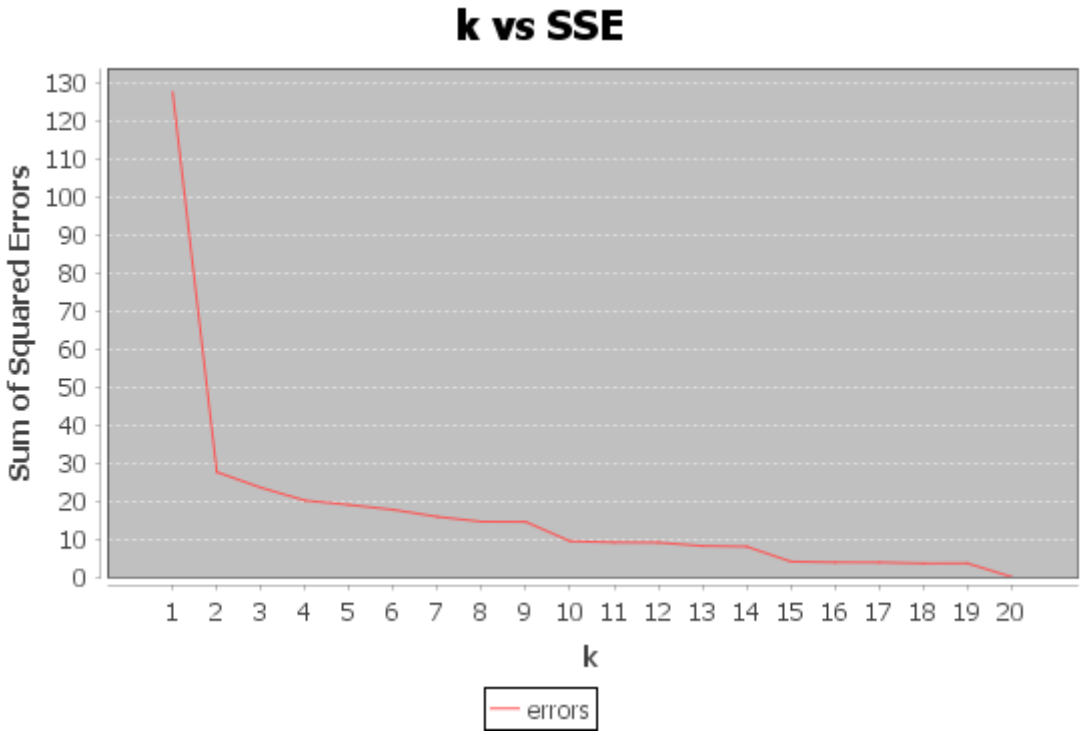
Dôležitým aspektom pri zhlukovaní je rovnomerné výsledné rozdelenie inštancií do skupín.. Našim cieľom bolo dosiahnuť čo najrovnomernejšie rozdelenie útočníkov do skupín, aby následná analýza jednotlivých skupín bola čo najpresnejšia.

Z tohto dôvodu sme údaje upravovali a následne na ne aplikovali zhlukovací algoritmus spôsobmi, ktoré sú popísané v nasledujúcich podkapitolách.

5.4.1 Zhlukovanie nad pôvodnými údajmi

V tomto kroku sme údaje nijak neupravovali, nevynechávali žiadne záznamy ani atribúty. K-means zhlukovanie sme spustili s parametrom k = 10, pre ktorý sme sa

rozhodli na základe nasledovného grafu „elbow“ metódy. Empiricky je z tohto grafu vidieť, že pri tejto hodnote nastal prudší pokles hodnoty SSE.



Obrázok 13Graf závislosti SSE od "k"

Pre tento zvolený parameter bolo rozdelenie do skupín nasledovné:

Tabuľka 4 Rozdelenie útočníkov do skupín

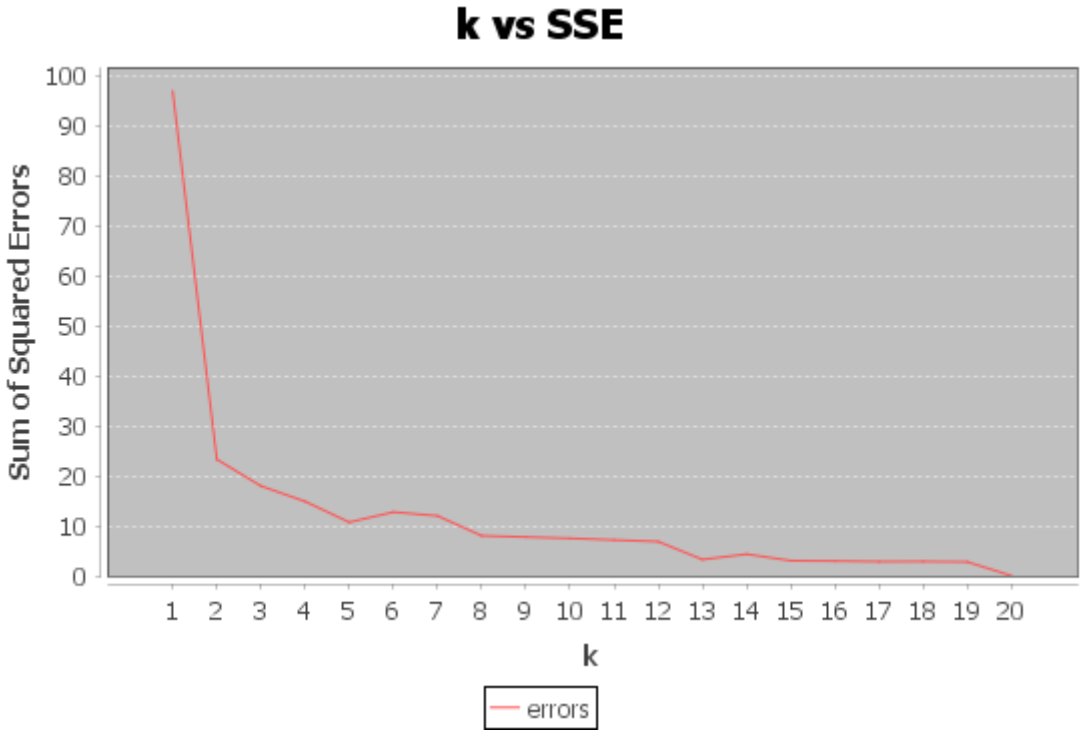
Skupina útočníkov	**Počet**
0	27 (1%)
1	13798 (84%)
2	22 (0%)
3	357 (1%)
4	47 (0%)
5	5147 (3%)
6	689 (2%)
7	72 (0%)
8	82 (0%)
9	9398 (9%)

Ako vidíme, tak rozdelenie útočníkov do skupín je veľmi nerovnomerné. V skupine č. 1 je až 84% všetkých útočníkov, čo je pre účely ďalšej analýzy

nevyhovujúce. Rovnako sme zhlukovanie spustili aj pre iné hodnoty „k“ (8,9,15) avšak s podobným nerovnomerným výsledným rozdelením útočníkov do skupín.

5.4.2 Vynechanie najväčšieho zhluku

Z pôvodných údajov, ktoré sme dávali na vstup pre zhlukovanie, sme vynechali všetky tie záznamy, ktoré boli zaradené do najväčšieho zhluku. V tom prípade „k“ pre takto upravený súbor záznamov predstavujúcich útočníkov bol na základe „elbow“ metódy a nasledujúceho grafu zvolené rovné šiestim (vyskúšané boli aj iné hodnoty, ale pre k=6 bolo rozdelenie do zhlukov najrovnomernejšie).



Obrázok 14 Graf závislosti SSE od "k"

Výsledné rozdelenie do zhlukov bolo v nasledujúcich pomeroch:

Tabuľka 5 Rozdelenie útočníkov do skupín

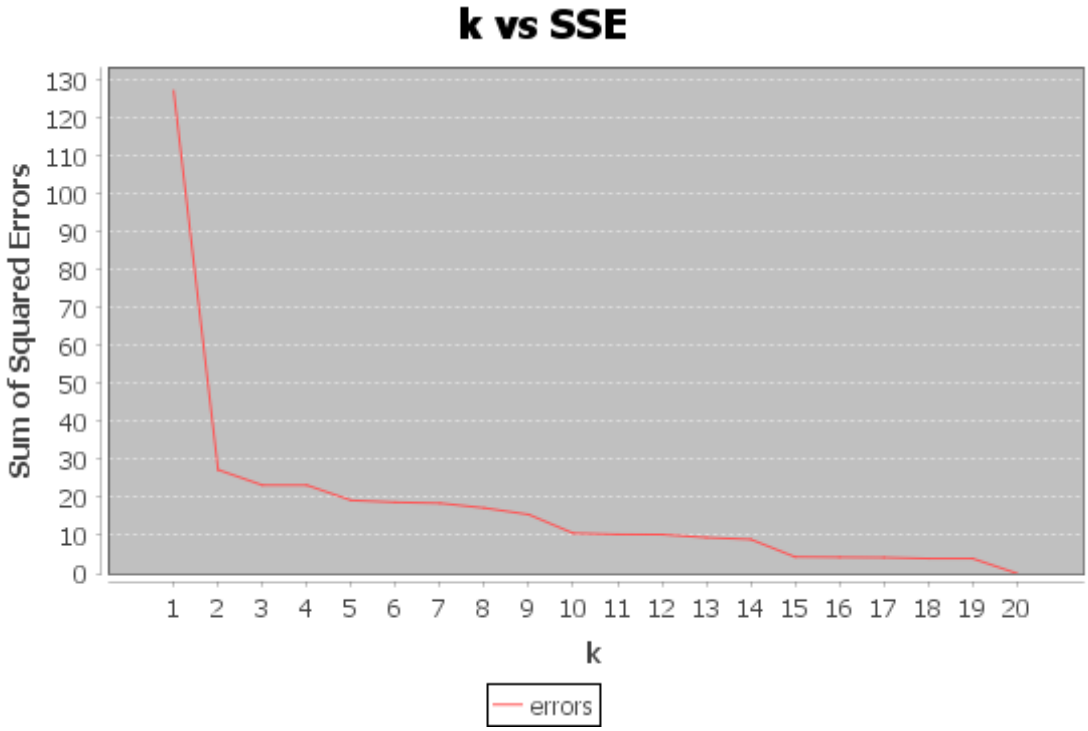
Skupina útočníkov	Počet
0	141 (20%)
1	51 (7%)
2	174 (24%)
3	244 (34%)
4	95 (13%)
5	9 (1%)

Ako vidíme, tak v tomto prípade je už rozdelenie do skupín o niečo rovnomernejšie.

5.4.3 Uvedenie informácie typu 0/1 pre útok typu Recon.Scanning

Keďže sa v údajoch nachádza najviac útokov **kategórie Recon.Scanning**, čo môže do značnej miery ovplyvňovať výsledky zhlukovania, zvolili sme prístup, že pre tento atribút nebudeme uvádzať koľkokrát útočník tento typ útoku vykonal, ale len informáciu, či tento typ útoku vykonal alebo nie. Ide o útoky, v ktorých útočník posiela požiadavky na systém s cieľom odhaliť jeho zraniteľnosti. I keď táto činnosť sama o sebe nie je škodlivá, útočník získané informácie môže využiť pri vykonaní iných, závažnejších útokov.

Pre takto upravené záznamy sme podľa nasledujúceho grafu určili „k“ pre k-means zhlukovanie rovné 10 (vyskúšali sme aj hodnotu 5, avšak rozdelenie do zhlukov bolo rovnako nerovnomerné ako pre hodnotu 10).



Obrázok 15 Graf závislosti SSE od "k"

Rozdelenie útočníkov do zhlukov bolo v tomto prípade nasledovné:

Tabuľka 6 Rozdelenie útočníkov do skupín

Skupina útočníkov	Počet
0	29 (1%)
1	3795 (84%)
2	2 (0%)
3	60 (1%)
4	7 (0%)
5	147 (3%)
6	87 (2%)
7	2 (0%)
8	2 (0%)
9	398 (9%)

Keďže ani toto rozdelenie nie je rovnomerné, pokračovali sme nasledujúcim prístupom.

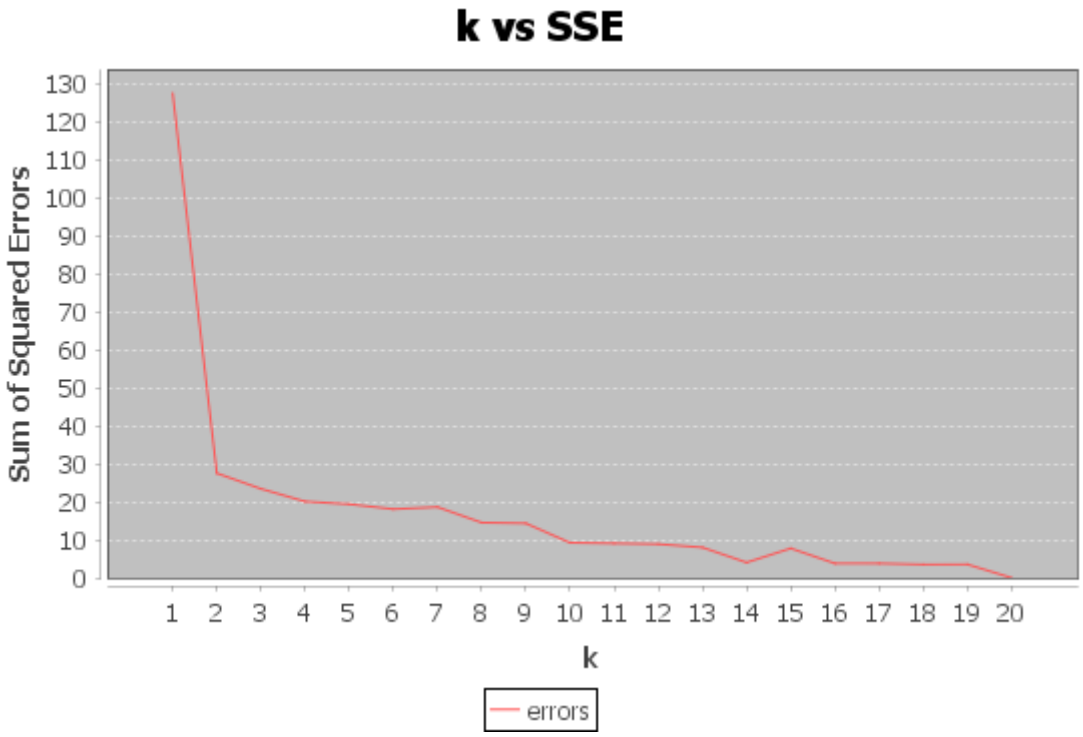
5.4.4 Odstránenie atribútov s nulovými hodnotami centroidov vo všetkých zhlukoch

Na nasledujúcom obrázku môžeme vidieť, že väčšina atribútov má v centroidoch vo všetkých nájdených zhlukoch nulové hodnoty. Preto sme vo vstupnom súbore pre zhlukovanie vynechali tieto atribúty a ďalej sme pracovali len s atribútmi Recon.Scanning, Availability.DDoS, TCP, (udp, dns), duration, maxIdleness, isp, count.

Recon.Scanning	0.0013	0.1294	0.0003	0.0116	0.0006	0.0879	0.0021	0.0035	0.0002	0.8128	0.0018
Attempt.Login	0	0	0	0	0	0	0	0	0	0	0
Attempt.Exploit	0	0	0	0	0	0	0	0	0	0	0
Intrusion.Botnet	0	0	0	0	0	0	0	0	0	0	0
Anomaly.Traffic	0	0	0	0	0	0	0	0	0	0	0
(Malware, Test)	0	0	0	0	0	0	0	0	0	0	0
(Other, Test)	0	0	0	0	0	0	0	0	0	0	0
Abusive.Spam	0	0	0	0	0	0	0	0	0	0	0
(Fraud.Phishing, Test)	0	0	0	0	0	0	0	0	0	0	0
Availability.DoS	0	0	0	0	0	0	0	0	0	0	0
Anomaly.Connection	0	0	0	0	0	0	0	0	0	0	0
(Recon.Scanning, Anomaly.Protocol, Test)	0	0	0	0	0	0	0	0	0	0	0
Vulnerable.Config	0	0	0	0	0	0	0	0	0	0	0
Availability.DDoS	0.0005	0	0	0	0	0	0	0.0275	0	0	0
(Attempt.Login, Test)	0	0	0	0	0	0	0	0	0	0	0
(Attempt.Exploit, Test)	0	0	0	0	0	0	0	0	0	0	0
(Recon.Scanning, Test)	0	0	0	0	0	0	0	0	0	0	0
(Itrusion.Botnet, Test)	0	0	0	0	0	0	0	0	0	0	0
(Itrusion.Botnet, Malware)	0	0	0	0	0	0	0	0	0	0	0
(Abusive.Spam, Test)	0	0	0	0	0	0	0	0	0	0	0
(Attempt.Exploit, Malware)	0	0	0	0	0	0	0	0	0	0	0
(Availability.DoS, Test)	0	0	0	0	0	0	0	0	0	0	0
TCP	0.0013	0.1294	0.0003	0.0116	0.0006	0.0879	0.0021	0.0035	0.0002	0.8128	0.0018
SSH	0	0	0	0	0	0	0	0	0	0	0
SMTP	0	0	0	0	0	0	0	0	0	0	0
(tcp, telnet)	0	0	0	0	0	0	0	0	0	0	0
FTP	0	0	0	0	0	0	0	0	0	0	0
HTTP	0	0	0	0	0	0	0	0	0	0	0
UDP	0	0	0	0	0	0	0	0	0	0	0
IMAP	0	0	0	0	0	0	0	0	0	0	0
(udp, dns)	0.0005	0	0	0	0	0	0	0.0275	0	0	0
SIP	0	0	0	0	0	0	0	0	0	0	0
(tcp, ssh)	0	0	0	0	0	0	0	0	0	0	0
(tcp, ms-wbt-server)	0	0	0	0	0	0	0	0	0	0	0
port 0-1023	0	0	0	0	0	0	0	0	0	0	0
port 1024-65535	0	0	0	0	0	0	0	0	0	0	0
duration	0.0006	0.0533	0.0001	0.0048	0.0002	0.0361	0.0007	0.0032	0.0001	0.5007	0.0007
maxIdleness	0.0548	0.0289	0.001	0.0351	0.7444	0.09	0.086	0.2362	0.0272	0.1123	0.4194
isp	0.0116	0.0451	0.001	0.1795	0.0312	0.6823	0.0395	0.0267	0	0.0235	0.072
count	0.0008	0.0009	0.0002	0.0019	0.0005	0.0075	0.0029	0.0006	0.8069	0.0001	0.0018

Obrázok 16 Centroidy pre jednotlivé zhluky

Na základe elbow metódy a nasledujúceho grafu sme ako parameter „k“ určili 7.



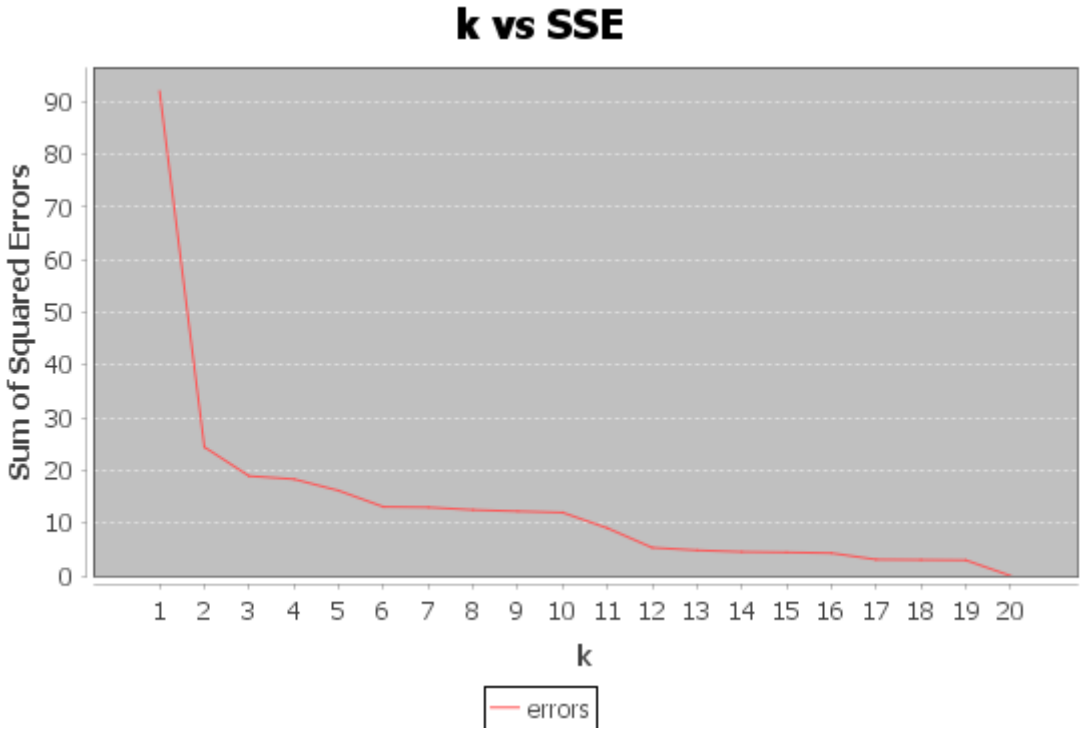
Obrázok 17 Graf závislosti SSE od "k"

Pre k=7 vyšlo nasledovné rozdelenie:

Tabuľka 7 Rozdelenie útočníkov do skupín

Skupina útočníkov	Počet
0	27 (1%)
1	3672 (81%)
2	187 (4%)
3	446 (10%)
4	7 (0%)
5	101 (2%)
6	89 (2%)

Keďže aj toto rozdelenie je nerovnomerné, spustili sme zhlukovací algoritmus na týchto údajoch, avšak bez záznamov patriacich do najväčšieho zhluku, teda zhluku č. 1. Na základe nasledujúceho grafu elbow metódy sme zvolili k=6.



Obrázok 18 Graf závislosti SSE od "k"

Výsledné rozdelenie bolo nasledovné:

Tabuľka 8 Rozdelenie útočníkov do skupín

Skupina útočníkov	Počet
0	36 (4%)
1	329 (38%)
2	98 (11%)
3	10 (1%)
4	109 (13%)
5	275 (32%)

5.5 Analýza výsledkov

Keďže najrovnomernejšie rozdelenie útočníkov do skupín vyšlo po tom, čo sme z údajov vynechali najväčší zhluk, tak ďalšiu analýzu a popis jednotlivých skupín budeme robiť nad výsledkom zhlukovania, ktoré bolo vykonané nad takto upravenými

údajmi. Jednotlivé skupiny označíme A1,A2,...,A6. Hodnoty ich atribútov sú nasledovné:

Tabuľka 9 Hodnoty atribútov "ideálnych" útočníkov

Skupi-na	Recon. Scanning	Availability. DDoS	TCP	udp, dns	Duration	maxIdleness	ISP	number of different targets
A1	130	0	130	0	87512.0	87659.0	7	141
A2	13	0	13	0	6948.0	922764.0	4	6
A3	20	0	20	0	8976.0	577478.0	7	16
A4	48	5	48	5	52534.0	462826.0	8	21
A5	58	0	58	0	50456.0	257000.0	3	14
A6	4731	0	4731	0	4571637.0	116383.0	48	59

Pre lepší prehľad o hodnotách jednotlivých atribútov sme vytvorili grafy, ktoré sú súčasťou prílohy E.

Analýza popísaná v nasledujúcich podkapitolách vychádza z teórie popísanej v kapitolách 2.4 a 3. Dosiahnuté výsledky odzrkadľujú fakt, že v údajoch, s ktorými sme pracovali, sa nenachádza celé spektrum typov útočníkov. Teda nenachádzajú sa v nich útoky, za ktorými stáli napr. útočníci zo skupiny kyberzločincov, štátov a pod. Väčšinou ide o menej nebezpečných útočníkov typu script-kiddie.

5.5.1 Popis správania sa útočníka zaradeného do skupiny A1

Útočník zo **skupiny A1** sa vyznačuje najmä vysokým počtom rôznych cieľových zariadení, na ktoré útočil. Útoky, ktoré vykonal boli typu Recon.Scanning, t.j. išlo o útoky, v ktorých útočník posiela požiadavky na systém s cieľom odhaliť jeho zraniteľnosti, ktoré je možné neskôr využiť pri ďalších útokoch. Útočník touto aktivitou získava informácie o systéme, službách a účtoch na danom systéme. Keďže útočník tohto typu vykonal útok z kategórie Recon.Scanning na veľké množstvo rôznych

cieľov, ktoré patrili do viacerých sietí, je možné vydedukovať, že jeho cieľom nebolo spôsobiť škodu, ale len získať informácie o zraniteľnostiach na jednotlivých systémoch. Keďže ide o veľké množstvo rôznych počítačov, je možné, že dôvodom tejto jeho činnosti bolo ich využitie pre neskoršie vykonanie DDoS útoku, čiže jeho **motiváciou** mohla byť **pomsta**. S veľkou pravdepodobnosťou **nejde o útočníka** z kategórie **Insider**, keďže sa útočník zameral na viacero sietí a o všetkých následne skenovaním získaval ďalšie informácie. Celkové trvanie útokov tohto útočníka bolo relatívne krátke a takisto maximálny časový odstup medzi dvoma za sebou idúcimi útokmi bol v tejto skupine útočníkov najkratší. Z toho je možné odvodiť, že **útoky boli vykonávané automaticky**, nie manuálne. Útočník z tejto skupiny využíval len protokol TCP, čo je však pri skenovaní siete pochopiteľné.

5.5.2 Popis správania sa útočníka zaradeného do skupiny A2

Útočníci zo **skupiny A2** sa vyznačujú najmä tým, že medzi dvoma za sebou idúcimi útokmi uplynie dlhý čas. Z toho môžeme usúdiť, že tento typ útočníka vykonával **manuálne útoky**. Avšak celkové trvanie jeho útokov bolo v porovnaní najmä so skupinou A6 veľmi krátke, okrem toho vykonával len útoky typu Recon.Scanning, ktoré boli namierené len na šesť rôznych cieľov zo štyroch rôznych sietí. Útočník tohto typu využíval len TCP protokol. Na základe týchto vlastností môžeme dedukovať, že útočník sa zameral na konkrétne, dopredu vytipované ciele, o ktorých sa pokúšal získať viac informácií. Druhá možnosť je, že ciele, na ktoré útočil boli vybrané náhodne a útočník sa na nich len „učil“, t.j. je **možné**, že patrí k skupine **script-kiddies**.

5.5.3 Popis správania sa útočníka zaradeného do skupiny A3

Skupina A3 sa vyznačuje nízkym počtom rôznych cieľov, na ktoré útočníci útočia a tiež relatívne malým množstvom počítačových sietí, na ktoré sa zamerali. Môžeme **predpokladať**, že útoky boli vykonávané **manuálne**, keďže hodnota atribútu maxIdleness je po skupine A2 druhá najvyššia. Celkové trvanie útokov týchto útočníkov bolo veľmi krátke a vykonávané boli len útoky typu Recon.Scanning, ktorých bolo len malé množstvo. Z týchto dôvodov môžeme povedať, že táto skupina zahŕňa

útočníkov s nižšími schopnosťami, avšak **nie úplných začiatočníkov**, keďže útoky vykonávajú pravdepodobne manuálne.

5.5.4 Popis správania sa útočníka zaradeného do skupiny A4

Útočníci zo **skupiny A4** ako jediní vykonali aj útok rôzny od Recon.Scanning, a to Availability.DDoS. To znamená, že útočníci z tejto skupiny okrem skenovania siete cielene posielali obrovské množstvo požiadaviek na konkrétny systém, aby tento systém spomalili alebo úplne vyradili z prevádzky s cieľom **spôsobiť napríklad finančnú stratu** ich prevádzkovateľom. Ako jediní tiež nevyužili iba TCP protokol, ale aj kombináciu protokolov UDP a DNS. Ich **motiváciou** bola teda **pravdepodobne pomsta** a ich **schopnosti a skúsenosti** s útočením boli **na vyššej úrovni** ako pri predchádzajúcich skupinách. V priemere sa títo útočníci pokúsili vyradiť z prevádzky len dvadsať systémov, teda nemusí ísť nutne o nebezpečných útočníkov zo skupiny cyber-criminals, ktorých ciele a motivácie predstavujú rozsiahlejšiu hrozbu. Jednou z možností je, že títo útočníci sú súčasťou **skupiny hacktivistov.**

5.5.5 Popis správania sa útočníka zaradeného do skupiny A5

Útočníci zo **skupiny A5** sú veľmi podobní útočníkom zo skupiny A3. Napriek tomu ich zhlukovací algoritmus rozdelil. Výraznejšie sa líšia len v maximálnom časovom odstupe medzi dvoma za sebou idúcimi útokmi. Ten je v skupine A5 asi o päť šestín kratší v porovnaní so skupinou A3. To znamená, že tento typ útočníka vykonáva **automatické útoky**. Táto kombinácia vlastností môže znamenať že táto skupina s najväčšou pravdepodobnosťou zahŕňa typických útočníkov typu **script-kiddies.**

5.5.6 Popis správania sa útočníka zaradeného do skupiny A6

Útočník zo **skupiny A6** útočil na najväčší počet počítačových sietí, avšak v priemere na relatívne málo rôznych systémov. To znamená, že cieľom jeho záujmu je veľký počet navzájom rôznych sietí. Táto skupina sa vyznačuje dlhým časom trvania útokov, avšak krátkymi rozostupmi medzi jednotlivými útokmi, z čoho je možné povedať, že útoky sú **automatizované**, avšak skripty, ktoré sa spúšťajú pri útoku bežia dlhší čas. To, a tiež fakt, že táto skupina útočníkov len skenuje počítačovú sieť, resp.

zariadenia v nej, môže signalizovať, že útočník sa zameriava na detailnejšie informácie o počítačovej sieti, a teda že ide **o skúsenejšieho útočníka**.

Tabuľka 10 Predpokladané vlastnosti jednotlivých skupín útočníkov

skupina	**automizovaný / manuálny útok**	**motivácia**	**zručnosti**	**vedomosti o obeti**	**typ útočníka**
A1	automatizovaný	pomsta	nízke	žiadne	script-kiddie/ hacktivist
A2	manuálny	zábava/“učenie sa“	nízke	žiadne/nízke	script-kiddie
A3	manuálny	zábava	priemerné	žiadne	script-kiddie
A4	automatizovaný	finančná strata/pomsta	vysoké	žiadne/nízke	hacktivist
A5	automatizovaný	zábava	nízke	žiadne	script-kiddie
A6	automatizovaný	získanie informácií	priemerné	žiadne/nízke	script-kiddie

Tabuľka 10 predstavuje v konečnom dôsledku celkový výsledok našej práce. Pre každú zo siedmich nájdených skupín zobrazuje predpokladané vlastnosti každej skupiny a tiež predpokladané zaradenie do skupiny útočníkov z existujúcich modelov útočníkov.

Záver

Otázke bezpečnosti je potrebné venovať väčšiu pozornosť najmä kvôli narastajúcemu počtu útokov, ale aj zvyšujúcej sa sofistikovanosti postupov útočníkov pri neautorizovanom vniknutí do systému. Každý útočník sa vyznačuje istými vlastnosťami, ako je napríklad motivácia, ktorá ho viedla k útoku, schopnosti, ktoré dokáže využiť na dosiahnutie svojho cieľa, znalosti, ktoré má o systéme či napríklad nástroje, ktoré pri útoku využil. V práci sme popísali niekoľko existujúcich modelov útočníkov, ktoré uvažujú rôzne typy útočníkov na základe rôznych kritérií. Medzi ne patrí napr. delenie útočníkov podľa ich motivácie, miery účasti v komunikácii, rizika, ktoré predstavujú pre systém, účasti v sieti atď. Bližšie sme sa venovali aj tzv. circumplex modelu, ktorého autori klasifikujú útočníkov na základe ich motivácie, pričom vo váženom circumplex modeli je možné zohľadniť aj viacero motivácií útočníka.

V druhej časti práce sme navrhli vlastný prístup k identifikácii útočníkov. V prvých fázach práce sme uvažovali o prístupe, kedy by sme najprv navrhli, aké skupiny budeme uvažovať a potom jednotlivých útočníkov do týchto skupín zaraďovali. Za týmto účelom sme mali v pláne využiť napr. rozhodovacie stromy. Avšak tento prístup si vyžadoval, aby sme mali dopredu definované rozhodovacie pravidlá. Definovať tieto pravidlá na základe atribútov ako motivácia, schopnosti, či znalosti nebol vo všeobecnosti problém. Problematické by však bolo pre každého útočníka určiť tieto atribúty z údajov, ktoré sme mali k dispozícii. Preto sme sa rozhodli pre prístup opačným smerom - najprv nájsť skupiny útočníkov na základe údajov, ktoré sa v našich údajoch vyskytujú a až následne tieto skupiny bližšie popísať.

Pred tým, než je možné robiť akúkoľvek analýzu bezpečnostných údajov, je potrebné vyfiltrovať len tie údaje, ktoré predstavujú bezpečnostné incidenty. Robiť analýzu nad regulárnou prevádzkou by nemalo zmysel (obsahuje aj správanie legitímnych používateľov). Avšak údaje, ktoré máme k dispozícii obsahujú len útoky, keďže boli zachytené honeypotmi, resp. inými zariadeniami ako napr. IDS. Z toho dôvodu nebola potrebná žiadna ďalšia filtrácia údajov. Zvolený prístup je založený na využití k-means zhlukovania. V práci sme popísali niekoľko zhlukovacích algoritmov, ktoré sme zároveň aj aplikovali na naše údaje. Ako najvhodnejší pre naše údaje vyšiel práve k-means algoritmus. Jeho veľkou výhodou je najmä to, že systémový

administrátor nemusí poskytnúť explicitný popis rôznych tried útočníkov a zároveň je vhodný pre modely, kde sú skupiny reprezentované stredovým vektorom.

Každý útočník je v našom prípade reprezentovaný ako 40-rozmerný vektor, kde každá zložka predstavuje niektorú z vlastností útočníka. Na základe atribútov ako IP adresa, port, protokol, trvanie útoku či typ útoku a vhodne zvoleného „k“ pre k-means algoritmus sme množinu útočníkov rozdelili do niekoľkých skupín. Na určenie vhodného k pre k-means algoritmus sme zvolili empirickú metódu s názvom „elbow method“, ktorej podstatou je skúšanie rôznych hodnôt k a počítanie chyby SSE. Každá z nájdených skupín má svojho reprezentanta, t.j. stredový vektor, na základe ktorého sme každú skupinu ďalej analyzovali. Do úvahy sme brali atribúty ako typ útoku, protokol, celkové trvanie útoku, maximálny odstup medzi dvoma za sebou idúcimi útokmi, počet rôznych cieľov a počet rôznych cieľových sietí.

V rámci našej práce bola vytvorená aplikácia, ktorá vykonáva tri základne funkcie: predspracovanie údajov, nájdenie vhodného „k“ pre k-means zhlukovanie a vykonanie samotného zhlukovania. Výstupom je „k“ skupín útočníkov, kde pre každú skupinu sú uvedené hodnoty jednotlivých atribútov „ideálneho“ útočníka danej skupiny.

Analýza týchto nájdených skupín ukázala, že v údajoch z Wardenu, s ktorými sme pracovali, sa vo väčšine prípadov nenachádzajú žiadni vysoko nebezpeční kybernetickí útočníci. Ich činnosť spočívala vo väčšine prípadov v preskenovaní siete, teda v získaní potrebných informácií o systéme, na ktorý možno v budúcnosti zaútočia. V údajoch bola zhlukovacím algoritmom k-means nájdená aj skupina útočníkov vykonávajúca DDoS útok. V tomto prípade môžeme hovoriť o útočníkoch s väčšími schopnosťami v oblasti vykonávania útokov a pravdepodobne aj so závažnejšími motiváciami a cieľmi.

Identifikácia útočníkov má veľký význam pre ochranu informačných systémov a počítačových sietí. Ak poznáme, aký typ útočníka neautorizovane vnikol do systému, tak vieme s určitou pravdepodobnosťou predpovedať jeho ďalšie správanie a tomu prispôsobiť kroky zabraňujúce spôsobeniu škôd. Práve to je najväčším prínosom našej práce a zároveň aj námetom na ďalší výskum v tejto oblasti. Ak je totiž systém už „naučený“ a pozná, aké typy útočníkov naň môžu útočiť, tak výskyt nového útoku môže byť automaticky asociovaný s konkrétnym typom útočníka. Na tomto princípe by mohli byť navrhnuté nástroje, ktoré nielen upozornia administrátora na útočníka, ktorý práve napadol systém, ale aj automaticky vykonajú určité bezpečnostné opatrenia.

Zoznam použitej literatúry

[1] KRAUTSEVICH, Leanid; MARTINELLI, Fabio; YAUTSIUKHIN, Artsiom. Towards modelling adaptive attacker's behaviour. In: International Symposium on Foundations and Practice of Security. Springer Berlin Heidelberg, 2012. p. 357-364.

[2] HECKMAN, R. Attacker Classification to Aid Targeting Critical Systems for Threat Modelling and Security Review

[3] SACHS, M. H.; PARKER, T.; MILLER, T. Adversary Characterization and Scoring System. Presentation at Black Hat, 2003, 28-31.

[4] LENIN, Aleksandr; WILLEMSON, Jan; SARI, Dyan Permata. Attacker profiling in quantitative security assessment based on attack trees. In: Nordic Conference on Secure IT Systems. Springer International Publishing, 2014. p. 199-212.

[5] PANCHENKO, Andriy; PIMENIDIS, Lexi. Towards practical attacker classification for risk analysis in anonymous communication. In: IFIP International Conference on Communications and Multimedia Security. Springer Berlin Heidelberg, 2006. p. 240-251.

[6] VAN HOLSTEIJN, Rick, et al. A framework for the motivation of attackers in attack tree analysis.

[7] HIRT, A.; JACOBSON, M. J.; WILLIAMSON, C. Survey and analysis of anonymous communication schemes. Submitted to ACM Computing Surveys, Department of Computer Science, University of Calgary, December 2003.

[8] RAYMOND, J.-F. Traffic Analysis: Protocols, Attacks, Design Issues, and Open Problems. In H. Federrath, editor, Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability, pages 10–29. Springer-Verlag, LNCS 2009, July 2000

[9] JOSHI, R. C.; SARDANA, Anjali (ed.). Honeypots: a new paradigm to information security. CRC Press, 2011.

[10] GRUDZIECKI, T., et al. Proactive detection of security incidents: Honeypots. European Network and Information Security Agency2012, 2012.

[11] Symantec Corporation, White paper: Web based attacks

[12] MARINOS, L., BELMONTE, A., REKLEITIS, E. ENISA Threat Landscape 2015

[13] NIKITINA, Svetlana. Hackers as tricksters of the digital age: creativity in hacker culture. The Journal of Popular Culture, 2012, 45.1: 133-152.

[14] FRIEDMAN, Jon; HOFFMAN, Daniel V. Protecting data on mobile devices: A taxonomy of security threats to mobile computing and review of applicable defenses. Information Knowledge Systems Management, 2008, 7.1, 2: 159-180.

[15] BUYENS, Koen; DE WIN, Bart; JOOSEN, Wouter. Empirical and statistical analysis of risk analysis-driven techniques for threat management. In: Availability, Reliability and Security, 2007. ARES 2007. The Second International Conference on. IEEE, 2007. p. 1034-1041

[16] FARAHMAND, Fariborz, et al. A management perspective on risk of security threats to information systems. Information Technology and Management, 2005, 6.2: 203-225.

[17] LANDRETH, Bill; RHEINGOLD, Howard. Out of the inner circle: a hacker's guide to computer security. Bellevue, Washington: Microsoft Press, 1985.

[18] CHANTLER, Nicholas. Profile of a computer hacker. Florida: infowar, 1996.

[19] ROGERS, Marcus K. The psyche of cybercriminals: A psycho-Social perspective. In: Cybercrimes: A Multidisciplinary Analysis. Springer Berlin Heidelberg, 2011. p. 217-235.

[20] GLENNY, Misha. Darkmarket: how hackers became the new mafia. Random House, 2012.

[21] SEEBRUCK, Ryan. A typology of hackers: Classifying cyber malfeasance using a weighted arc circumplex model. Digital Investigation, 2015, 14: 36-45.

[22] LIN, Tom CW. Financial Weapons of War. Minn. L. Rev., 2015, 100: 1377.

[23] JHA, Somesh; SHEYNER, Oleg; WING, Jeannette. Two formal analyses of attack graphs. In: Computer Security Foundations Workshop, 2002. Proceedings. 15th IEEE. IEEE, 2002. p. 49-63.

[24] OU, Xinming; SINGHAL, Anoop. Attack graph techniques. Quantitative Security Risk Assessment of Enterprise Networks, 2011, 5-8, [online]. [cit. 2017-06-28]. Dostupné z: http://www.springer.com/cda/content/document/cda_downloaddocument/9781461418597-c1.pdf?SGWID=0-0-45-1277175-p174200278

[25] BUCZAK, Anna L.; GUVEN, Erhan. A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications Surveys & Tutorials, 2016, 18.2: 1153-1176.

[26] Compilation of existing cybersecurity and information security related definitions [online]. [cit. 2017-06-02]. Dostupné z: https://www.newamerica.org/cyber-global/compilation-of-existing-cybersecurity-and-information-security-related-definitions/

[27] JAIN, Anil K.; MURTY, M. Narasimha; FLYNN, Patrick J. Data clustering: a review. ACM computing surveys (CSUR), 1999, 31.3: 264-323.

[28] MEYERS, Carol; POWERS, Sarah; FAISSOL, Daniel. Taxonomies of cyber adversaries and attacks: a survey of incidents and approaches. Lawrence Livermore National Laboratory (April 2009), 2009, 7: 1-22, [online]. [cit. 2017-06-24]. Dostupné z: https://e-reports-ext.llnl.gov/pdf/379498.pdf

[29] NAMRATHA, M.; PRAJWALA, T. R. A comprehensive overview of clustering algorithms in pattern recognition. IOR Journal of Computer Engineering, 2012, 4.6.

[30] 5 Protocols That Should Be Closely Watched [online]. [cit. 2017-06-28]. Dostupné z: http://www.darkreading.com/analytics/security-monitoring/5-protocols-that-should-be-closely-watched/d/d-id/1140977?

[31] IDEA [online]. [cit. 2017-06-28]. Dostupné z: https://idea.cesnet.cz/en/index

[32] Glastopf [online]. [cit. 2017-06-28]. Dostupné z: https://github.com/mushorg/glastopf

[33] PosgreSQL [online]. [cit. 2017-06-28]. Dostupné z: https://www.postgresql.org/

[34] Cesnet [online]. [cit. 2017-06-28]. Dostupné z: https://www.cesnet.cz/

[35] Elbow method [online]. [cit. 2017-06-28]. Dostupné z: https://bl.ocks.org/rpgove/0060ff3b656618e9136b

[36] Kippo [online]. [cit. 2017-06-28]. Dostupné z: https://github.com/desaster/kippo

[37] Dionaea [online]. [cit. 2017-06-28]. Dostupné z: https://github.com/rep/dionaea

[38] CANNADY, James. Artificial neural networks for misuse detection. In: National information systems security conference. 1998. p. 368-81.

[39] LIPPMANN, Richard P.; CUNNINGHAM, Robert K. Improving intrusion detection performance using keyword selection and neural networks. Computer Networks, 2000, 34.4: 597-603.

[40] KUOK, Chan Man; FU, Ada; WONG, Man Hon. Mining fuzzy association rules in databases. ACM Sigmod Record, 1998, 27.1: 41-46.

[41] BRAHMI, Hanen; BRAHMI, Imen; BEN YAHIA, Sadok. OMC-IDS: at the cross-roads of OLAP mining and intrusion detection. Advances in Knowledge Discovery and Data Mining, 2012, 13-24.

[42] HECKERMAN, David. A tutorial on learning with Bayesian networks. In: Learning in graphical models. Springer Netherlands, 1998. p. 301-354.

[43] NIELSEN, Thomas Dyhre; JENSEN, Finn Verner. Bayesian networks and decision graphs. Springer Science & Business Media, 2009.

[44] QUINLAN, J. R. Induction of decision trees. Machine Learning, 1: 81-106. 1986.

[45] QUINLAN, J. Ross. C4. 5: programs for machine learning. Elsevier, 2014.

[46] KHAN, M. Sadiq Ali. Rule based network intrusion detection using genetic algorithm. International Journal of Computer Applications, 2011, 18.8: 26-29.

[47] Weka [online]. [cit. 2017-06-28]. Dostupné z: http://www.cs.waikato.ac.nz/ml/weka/

[48] IP-API [online]. [cit. 2017-06-28]. Dostupné z: http://ip-api.com/

[49] Java [online]. [cit. 2017-06-28]. Dostupné z: https://www.oracle.com/java/index.html

[50] PostgreSQL [online]. [cit. 2017-06-28]. Dostupné z: https://www.postgresql.org/

[51] JDBC [online]. [cit. 2017-06-28]. Dostupné z: https://jdbc.postgresql.org/

[52] JFreeChart [online]. [cit. 2017-06-28]. Dostupné z: http://www.jfree.org/jfreechart/

[53] Swing [online]. [cit. 2017-06-28]. Dostupné z: https://docs.oracle.com/javase/7/docs/api/javax/swing/ package-summary.html

[54] Java Util Scanner [online]. [cit. 2017-06-28]. Dostupné z: https://docs.oracle.com/javase/7/docs/api/java/util /Scanner.html

[55] Java Print Writer [online]. [cit. 2017-06-02]. Dostupné z: https://docs.oracle. com/javase/7/docs/api/java/io/PrintWriter.html

[56] Warden projekt [online]. [cit. 2017-06-02]. Dostupné z: https://warden. cesnet.cz/cs/index

[57] DUA, Sumeet; DU, Xian. Data mining and machine learning in cybersecurity. CRC press, 2016.

[58] PELLEG, Dan, et al. X-means: Extending K-means with Efficient Estimation of the Number of Clusters. In: ICML. 2000. p. 727-734.

[59] LLETI, R., et al. Selecting variables for k-means cluster analysis by using a genetic algorithm that optimises the silhouettes. Analytica Chimica Acta, 2004, 515.1: 87-100.

[60] Finding the Right Number of Clusters in k-Means and EM Clustering: v-Fold Cross-Validation [online]. [cit. 2017-06-02]. Dostupné z: http://www.statsoft.com/Textbook/Cluster-Analysis#vfold

[61] Hierarchical clustering [online]. [cit. 2017-06-02]. Dostupné z: http:// www.solver.com/xlminer/help/hierarchical-clustering-intro

[62] ZHU, Xiaojin. CS838-1 Advanced NLP: The EM Algorithm, cit. 2017-06-24. Dostupné na internete: http://pages.cs.wisc.edu/~jerryzhu/cs838/EM.pdf

[63] ESTER, Martin, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Kdd. 1996. p. 226-231, cit. 2017-06-24. Dostupné na internete: https://www.aaai.org/Papers/KDD/1996/KDD96-037.pdf

[64] AGRAWAL, Rakesh, et al. Automatic subspace clustering of high dimensional data for data mining applications. ACM, 1998 [online]. [cit. 2017-06-24]. Dostupné z: http://www.leg.ufpr.br/~leonardo/artigos/sigmod1998-clique.pdf

[65] BUBECK, Sébastien; LUXBURG, Ulrike von. Nearest neighbor clustering: A baseline method for consistent clustering with arbitrary objective functions. Journal of Machine Learning Research, 2009, 10.Mar: 657-698, cit. 2017-06-24. Dostupné na internete: https://www.microsoft.com/en-us/research/wp-content/uploads/2017/01/bubeck09a.pdf

[66] Cobweb [online]. [cit. 2017-06-02]. Dostupné z: http://weka.sourceforge.net/ doc.dev/weka/clusterers/Cobweb.html

[67] ANKERST, Mihael, et al. OPTICS: ordering points to identify the clustering structure. In: ACM Sigmod record. ACM, 1999. p. 49-60, [online]. [cit. 2017-06-02]. Dostupné z: http://www.cs.ucsb.edu/~veronika/MAE/optics_ ankerstbreunigkriegelsander99.pdf

[68] FISHER, Douglas H. Knowledge acquisition via incremental conceptual clustering. Machine learning, 1987, 2.2: 139-172.

Prílohy

Príloha A: CD médium – diplomová práca v elektronickej podobe, prílohy v elektronickej podobe, aplikácia vytvorená v rámci práce, kódy implementované v práci, vstupné súbory na testovanie

Príloha B: Poradie atribútov útočníka vo vektore, ktorým je reprezentovaný

Príloha C: Používateľská príručka

Príloha D: Skript na doplnenie stĺpca ISP do tabuľky

Príloha E: Výsledky podľa kategórií útočníkov

Príloha A: CD médium

Obsah CD média:

- Práca spolu s prílohami v elektronickej podobe (diplomovaPraca.pdf)
- Aplikácia vytvorená v rámci práce (súbor Diploma_Thesis.jar v priečinku app)
- Kódy implementované v práci (súbory v priečinku project)
- Testovací súbor (input.csv)
- Testovací súbor (inputFinal.arff)

Príloha B: Poradie atribútov útočníka vo vektore, ktorým je reprezentovaný

@attribute IP string

@attribute Recon.Scanning numeric

@attribute Attempt.Login numeric

@attribute Attempt.Exploit numeric

@attribute Intrusion.Botnet numeric

@attribute Anomaly.Traffic numeric

@attribute '(Malware, Test)' numeric

@attribute '(Other, Test)' numeric

@attribute Abusive.Spam numeric

@attribute '(Fraud.Phishing, Test)' numeric

@attribute Availability.DoS numeric

@attribute Anomaly.Connection numeric

@attribute '(Recon.Scanning, Anomaly.Protocol, Test)' numeric

@attribute Vulnerable.Config numeric

@attribute Availability.DDoS numeric

@attribute '(Attempt.Login, Test)' numeric

@attribute '(Attempt.Exploit, Test)' numeric

@attribute '(Recon.Scanning, Test)' numeric

@attribute '(Itrusion.Botnet, Test)' numeric

@attribute '(Itrusion.Botnet, Malware)' numeric

@attribute '(Abusive.Spam, Test)' numeric

@attribute '(Attempt.Exploit, Malware)' numeric

@attribute '(Availability.DoS, Test)' numeric

@attribute TCP numeric

@attribute SSH numeric

@attribute SMTP numeric

@attribute '(tcp, telnet)' numeric

@attribute FTP numeric

@attribute HTTP numeric

@attribute UDP numeric

@attribute IMAP numeric

@attribute '(udp, dns)' numeric

@attribute SIP numeric

@attribute '(tcp, ssh)' numeric

@attribute '(tcp, ms-wbt-server)' numeric

@attribute 'port 0-1023' numeric

@attribute 'port 1024-65535' numeric

@attribute duration numeric

@attribute maxIdleness numeric

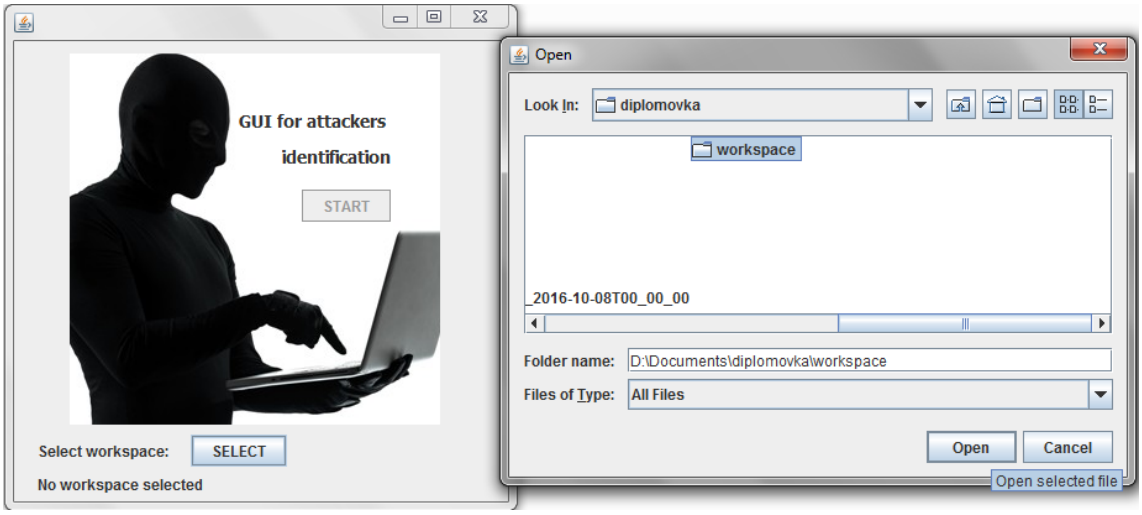
@attribute isp numeric

@attribute count numeric

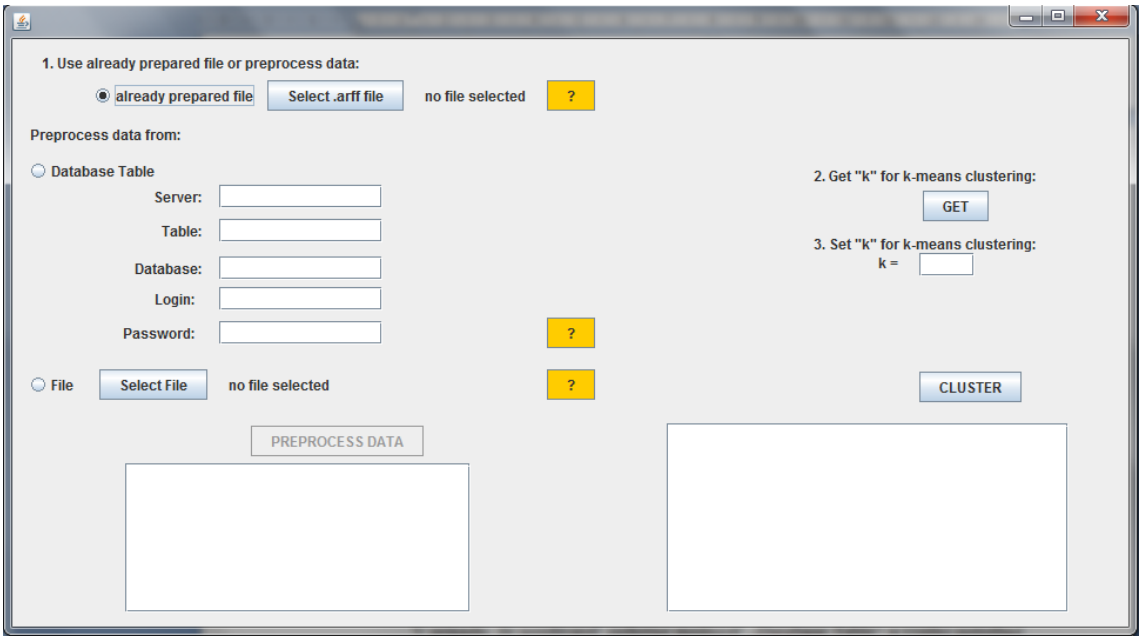
Príloha C: Používateľská príručka

Vytvorená aplikácia slúži na jednoduchú a intuitívnu prácu s bezpečnostnými údajmi. Je možné pracovať s údajmi uloženými v tabuľke s predpísanými stĺpcami uloženej v databáze, so súborom, ktorý je potrebné predspracovať do požadovaného formátu alebo priamo so súborom obsahujúcim údaje v predpísanom formáte (vo formáte .arff).

V úvodnom okne je potrebné zvoliť, kam sa budú ukladať a odkiaľ sa budú čítať súbory, ktoré sa vytvárajú v priebehu celého procesu predspracovania vstupného súboru, resp. údajov z databázovej tabuľky. Kliknutím na tlačidlo SELECT používateľ zvolí nejaký priečinok. Následne môže kliknutím na tlačidlo START vstúpiť do aplikácie.



V hlavnom okne aplikácie má používateľ na výber tri možnosti. Prvá z nich je výber predpripraveného súboru vo formáte .arff, v ktorom sú všetky atribúty pomenované tak, ako je uvedené v prílohe B, všetky hodnoty sú normalizované a každý riadok predstavuje jedného útočníka bez uvedenia IP adresy. V tomto prípade už používateľ nemá možnosť predspracovať údaje, ale priamo môže v druhom kroku získať *k* pre *k-means* zhlukovanie.



V prípade, že používateľ zaškrtne možnosť „Database Table“ a vyplní potrebné údaje, t.j. IP adresa servra, na ktorom beží jeho databáza, názov databázy, názov tabuľky, prihlasovacie meno a heslo, je nutné, aby tabuľka obsahovala nasledujúcu štruktúru:

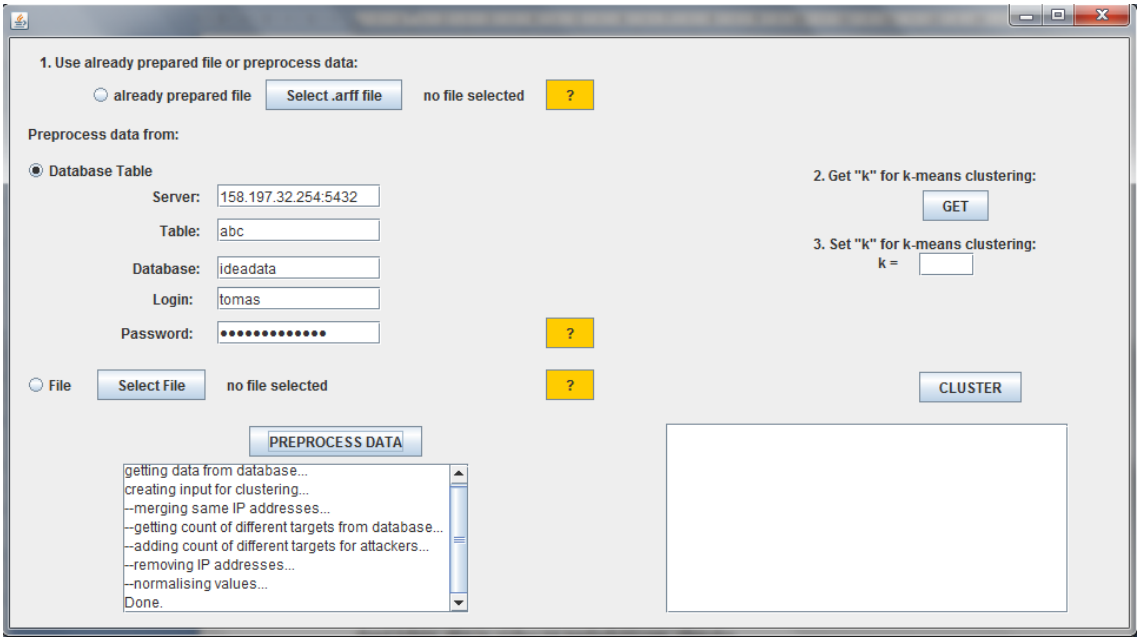
Stĺpec	Typ
id	bigint
ip	text
iptarget	text
category	text
countcategory	bigint
protocol	text
countproto	bigint
port	text
duration	double precision
start	timestamp with time zone
end	timestamp with time zone
isp	text

Ak chce používateľ pracovať s predpripraveným súborom, musí byť tento súbor vo formáte .CSV, kde jednotlivé hodnoty sú oddelené čiarkami. Každý riadok musí obsahovať 41 hodnôt v poradí, ako je uvedené v prílohe B, pričom úplne prvá hodnota musí byť IP adresa útočníka uvedená v apostrofoch. Príklad jedného riadku v tomto súbore:

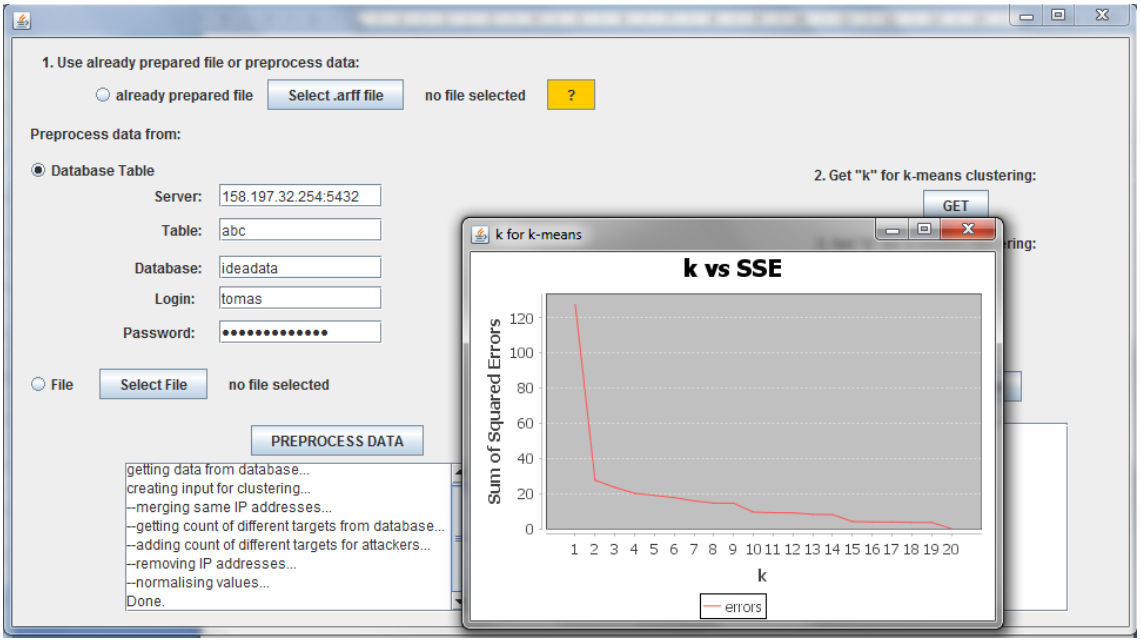
'103.7.30.237',52,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,52,0,0,0,0,0,0,0,0,0,0,0,0,0,900,534043,14,6

V prípade, že používateľ zaškrtne možnosť *File*, musí následne vybrať takto predpripravený súbor.

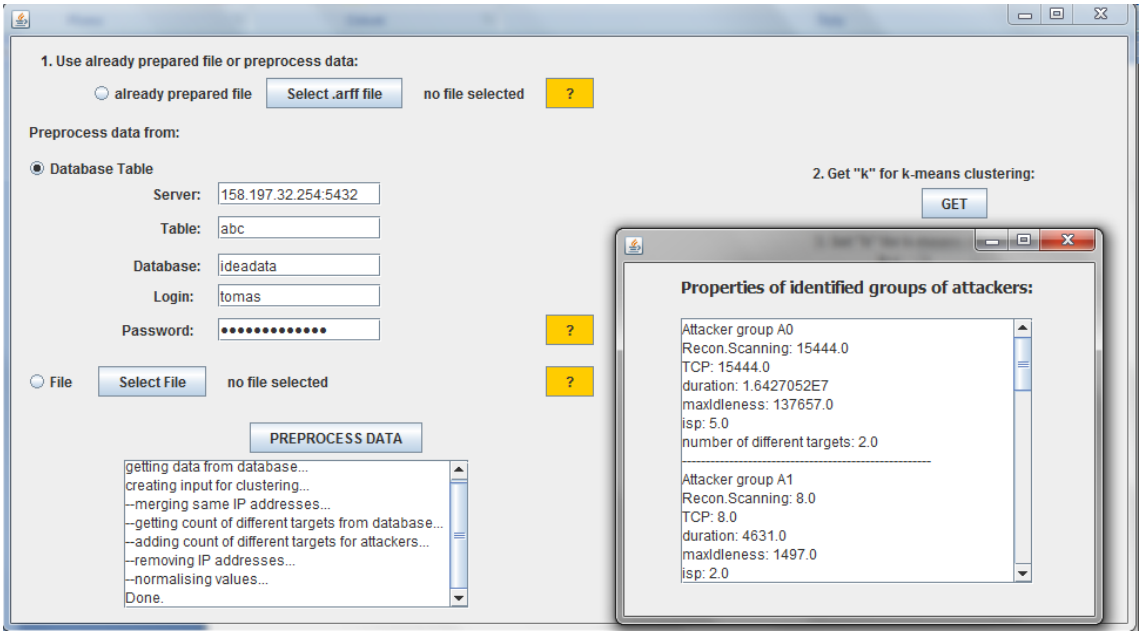
Po zaškrtnutí druhej alebo tretej možnosti sa automaticky aktivuje tlačidlo *PREPROCESS DATA*, a po kliknutí naň sa údaje predspracujú tak, aby bolo možné nad nimi spustiť zhlukovací algoritmus (krok 2 a 3). Na nasledujúcom obrázku vidíme ukážku prípadu, že údaje sa predspracúvajú z databázovej tabuľky. Celý proces predspracovania údajov je dokončený až po výpise *Done*.



Ďalší krok je získanie vhodného parametra *k* pre k-means zhlukovanie. Po kliknutí na tlačidlo *GET* sa používateľovi po uplynutí potrebného času na výpočet vykreslí graf, z ktorého má možnosť vyčítať, ktorá hodnota *k* bude najvhodnejšia pre dané údaje, ako je vidno na nasledujúcom obrázku.



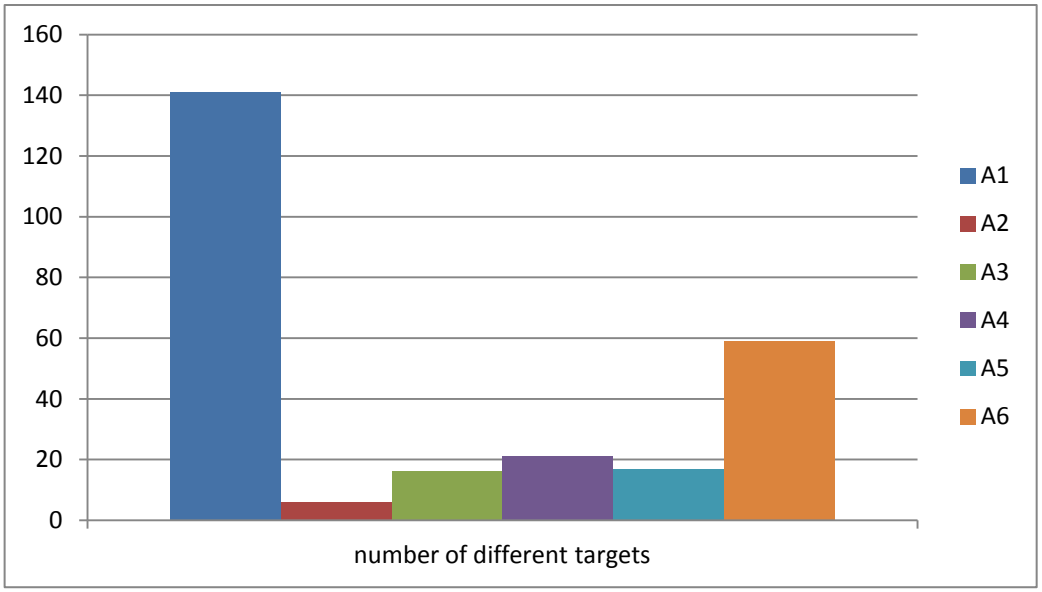
Na základe zobrazeného grafu (a znalosti „*elbow*“ *metódy*) si používateľ zvolí nejaké *k*, napr. 7. Nastaví túto zvolenú hodnotu do určeného textového poľa a po kliknutí na tlačidlo *CLUSTER* začne výpočet, ktorého výsledok sa vypíše do textového poľa pod ním tak, ako to zobrazuje nasledujúci obrázok. Celý proces je ukončený až po výpise slova *Done*. Po ukončení výpočtu sa celý výsledok zobrazí v samostatnom okne.



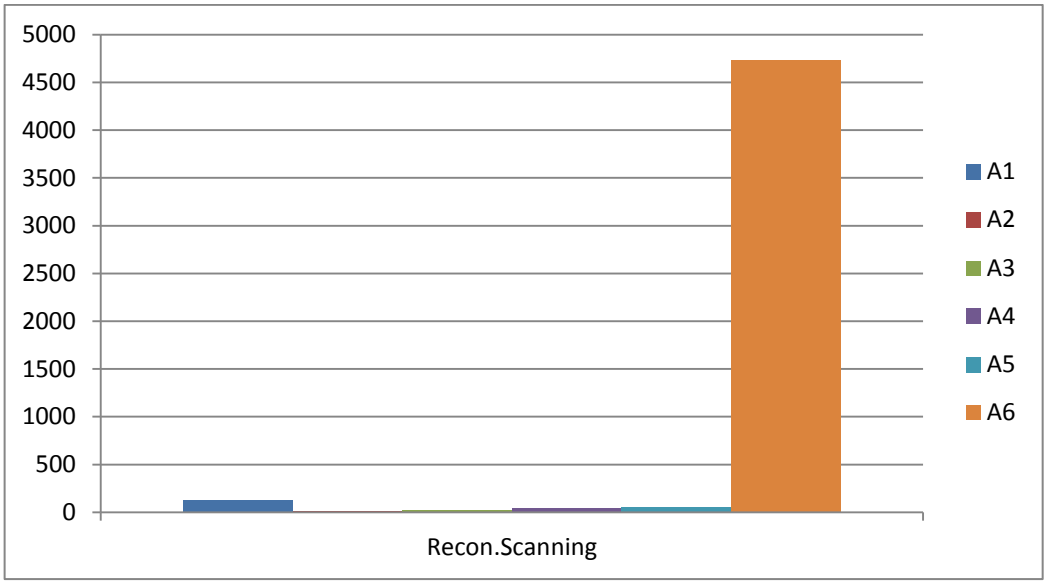
Príloha D: Skript na doplnenie stĺpca ISP do tabuľky

```
<?php
    $username = "tomas";
    $password = "quest1qtomat5";
    $host = "localhost";
    $database="ideadata";
    $conn     =     pg_connect("host=".$host."     dbname=".$database."
user=".$username." password=".$password);
    $sql = pg_query($conn, "select distinct iptarget from abc order by
iptarget");
    $count = 0;
    while ($row = pg_fetch_assoc($sql)) {
        $iptarget                =str_replace('"',                  "",
str_replace("]","",str_replace("[","",$row['iptarget'])));
        $iptarget_db = $row['iptarget'];
        $isp        =        @unserialize(file_get_contents("http://ip-
api.com/php/".$iptarget));
        if( $isp['status']=='success') {
                $update = "update abc set isp='".$isp['isp']."' where
iptarget='".$iptarget_db."'";
                if(pg_query($conn, $update)!=false) {
                        echo    'updated    '.$iptarget.'    with    isp
'.$isp['isp'];
                        echo $update."\n";
                } else {
                        echo 'not updated'.$iptarget."\n";
                }
        }
        $count++;
                if ($count % 200 == 0){
                        sleep(40);
                        echo 'Spiiim'."<br/>";
          } else {
          echo 'Unable to get location';
          }
    }
?>
```

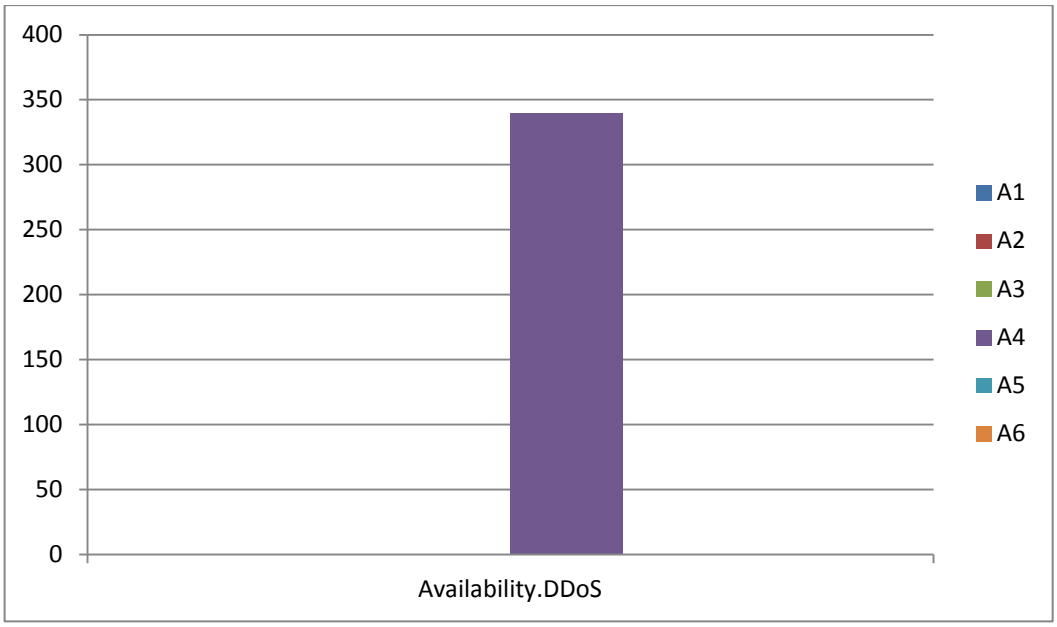
Príloha E: Výsledky podľa kategórií útočníkov



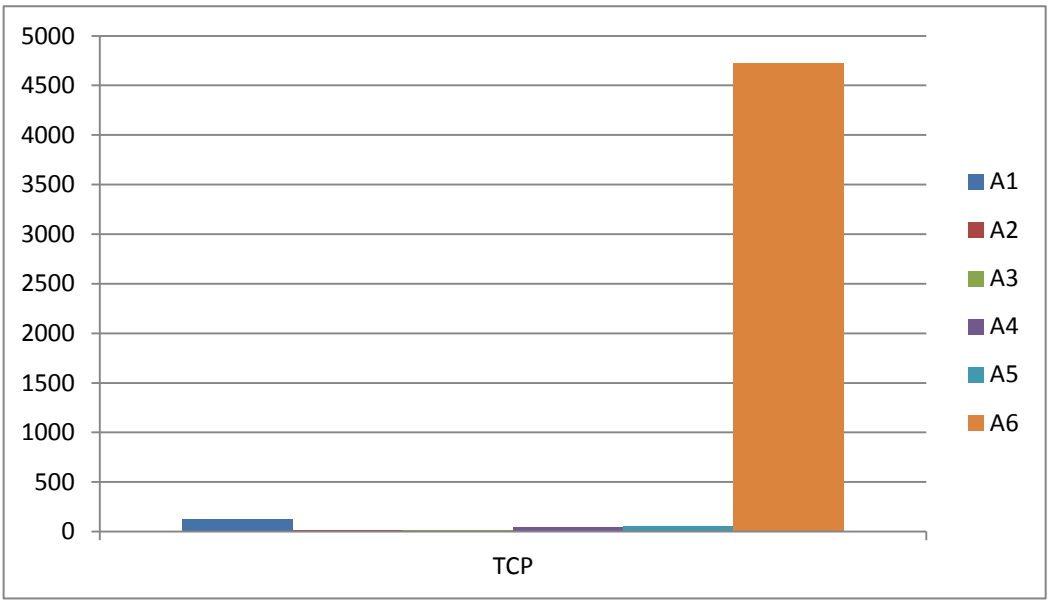
Obrázok 19 Počet rôznych cieľov



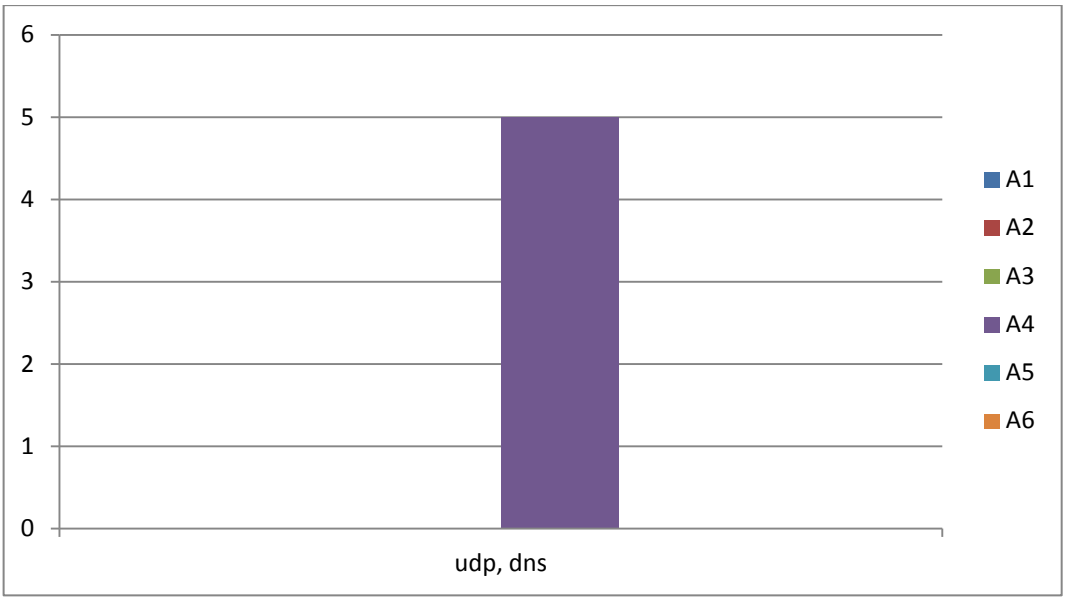
Obrázok 20 Typ útoku: Recon.Scanning



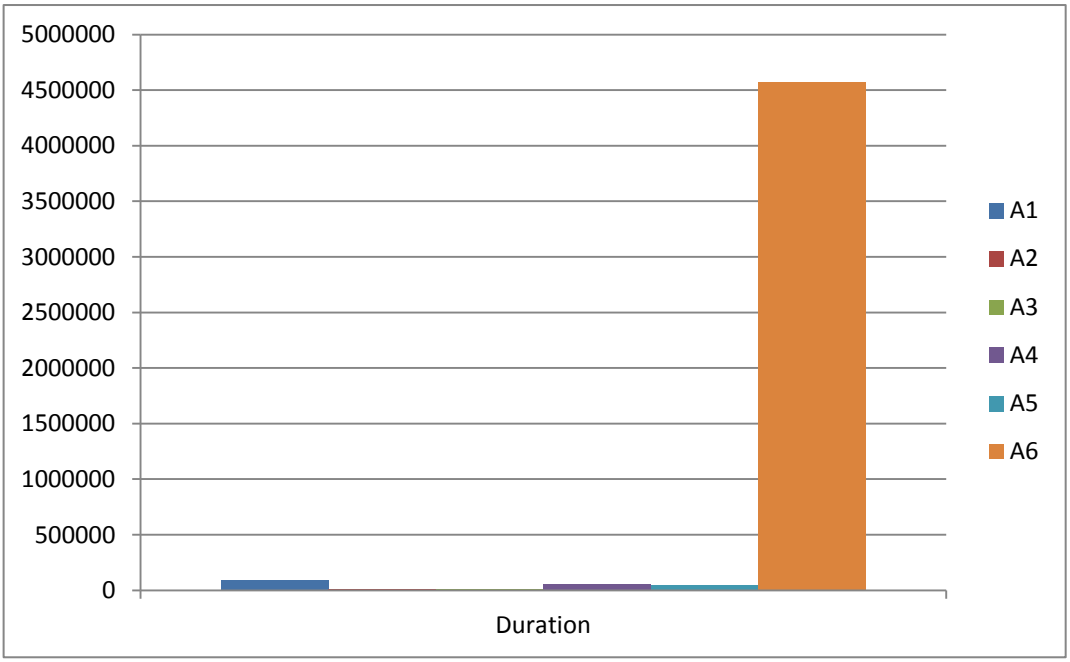
Obrázok 21 Typ útoku: Availability.DDoS



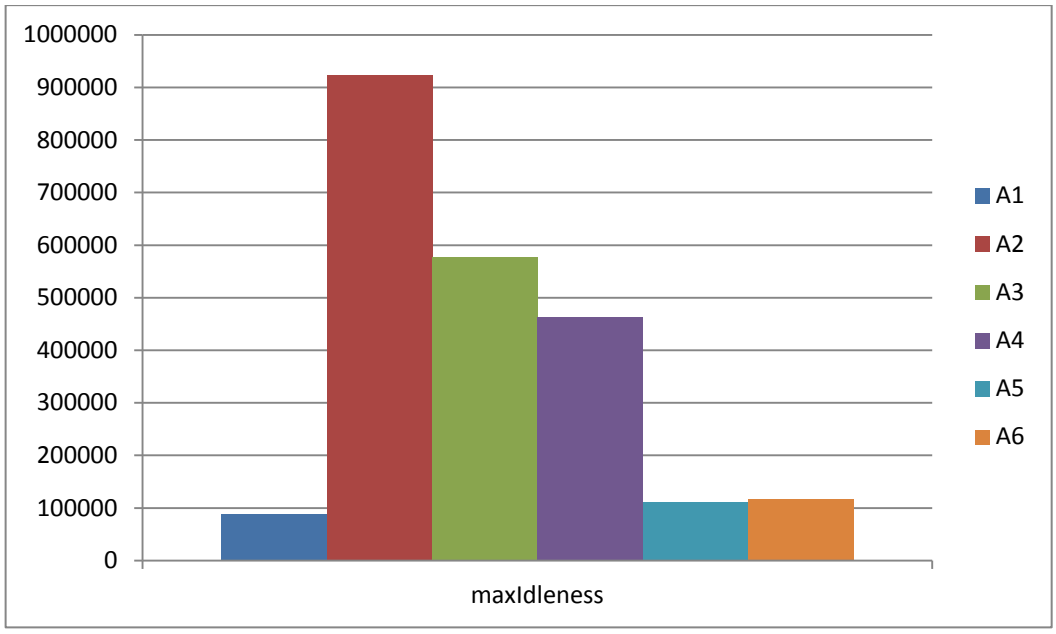
Obrázok 22 Protokol: TCP



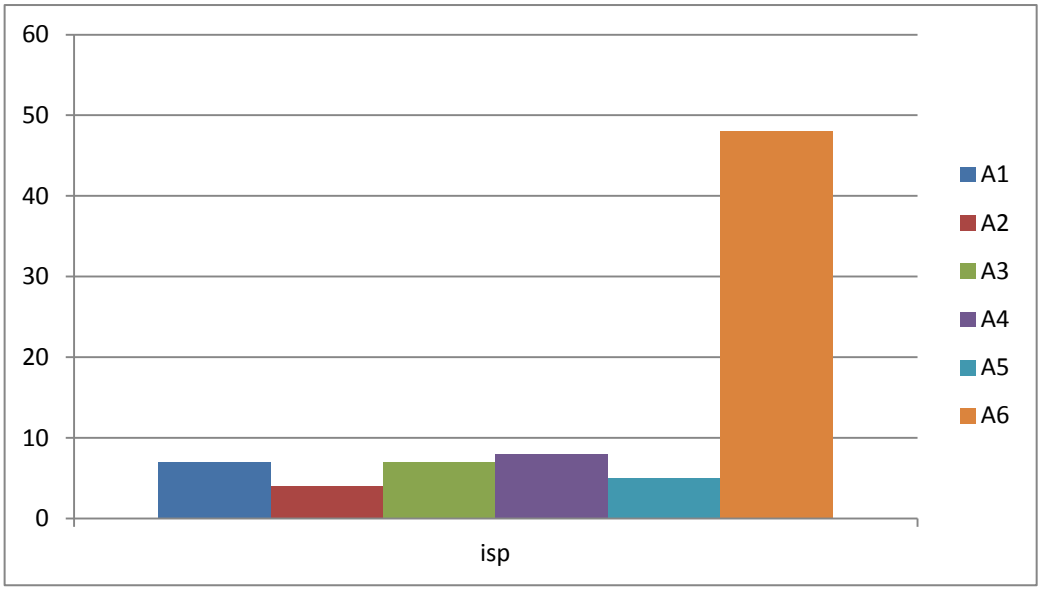
Obrázok 23 Protokol: UDP, DNS



Obrázok 24 Trvanie útokov (v sekundách)



Obrázok 25 Maximálny časový rozostup medzi dvoma za sebou nasledujúcimi útokmi (v sekundách)



Obrázok 26 Počet sietí, na ktoré útočník útočil