PROJEKT

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

V9 Popis vybraného algoritmu spĺňajúceho stanovené kritériá

KPB3 – Návrh ML algoritmov

typ – dokumentácia

mesiac odovzdania – M17

Financované
Európskou úniou
NextGenerationEU

PLÁN OBNOVY

ÚRAD VLÁDY
SLOVENSKEJ REPUBLIKY

Obsah

1 Úvod ... 3
2 Detekcia sieťových anomálií ... 8
2.1 Detekcia DoS/DDoS útokov ... 8
2.1.1 DoS útoky - datasety, experimenty a výsledky, popis vybraného algoritmu ... 11
2.1.1.1 Štatistické parametre ... 14
2.1.1.2 3σ-Tunel ... 19
2.2 Záver ... 22
3 Detekcia phishingových e-mailov a spoofingu ... 24
3.1 Experimenty a výsledky ... 25
3.2 Popis vybraného algoritmu ... 26
3.3 Záver ... 28
4 Spam/ham klasifikácia ... 30
4.1 Experimenty a výsledky ... 31
4.2 Popis vybraného algoritmu ... 31
4.3 Záver ... 32
5 Detekcia pokročilých APT techník ... 34
5.1 Experimenty a výsledky ... 35
5.2 Popis vybraného algoritmu ... 35
5.3 Záver ... 38
6 Klasifikácia a identifikácia malware vzorov ... 40
6.1 Experimenty a výsledky ... 41
6.2 Popis vybraného algoritmu ... 41
6.3 Záver ... 41
7 Záver – návrh ... 43
Prílohy ... 44

1 ÚVOD

Strojové učenie ML (Machine Learning) a umelá inteligencia AI (Artificial Intelligence) všeobecne sa používa pre účely spracovania a analýzy dát niekoľko desaťročí v širokej oblasti aplikácií a kybernetická bezpečnosť nie je výnimkou.

Bezpečnostné operačné centrum SOC (Security Operation Center) predstavuje centralizovanú jednotku zodpovednú za monitorovanie, detekciu, reakciu na a zmierňovanie alebo mitigáciu kybernetických hrozieb v reálnom čase. Rastúci objem, rýchlosť a komplexnosť kybernetických hrozieb spôsobili, že tradičná analýza založená na pravidlách a manuálna analýza sú nedostatočné.

Algoritmy a metódy AI/ML umožňujú rozšíriť SOC predovšetkým o nasledovné funkcie:

- automatizácia opakujúcich sa analytických a operačných úloh (napr. obohacovanie alertov, normalizácia logov, triáž incidentov)
- klasifikácia bezpečnostných udalostí a incidentov (rozlišovanie benigná vs. škodlivá sieťová prevádzka, rozlišovanie známych hrozieb a podozrivých anomálií)
- detekcia neznámych hrozieb a odchýlok od normálneho správania (zero-day útoky, novel attack patterns, insider threats)
- redukcia počtu falošne pozitívnych hlásení (false positive) alebo alertov
- prioritizácia a hodnotenie rizika bezpečnostných upozornení (risk scoring, threat scoring, contextual severity)
- podpora prediktívnej analýzy bezpečnostných rizík (predikcia pravdepodobnosti incidentov alebo eskalácie útokov)
- zrýchlenie reakcie na bezpečnostné incidenty (podpora rozhodovania v reálnom čase, odporúčanie reakčných krokov).

AI/ML metódy a algoritmy sa integrujú do pracovného postupe SOC systému vo viacerých fázach:

- **Monitorovanie**: V rámci fázy monitorovania sa AI/ML metódy využívajú predovšetkým na priebežný zber, predspracovanie a základnú analytickú interpretáciu bezpečnostných dát z rôznych zdrojov infraštruktúry. Cieľom tejto fázy nie je finálna identifikácia útoku, ale včasné odhalenie odchýlok od normálneho správania a vytvorenie kontextu pre následné detekčné mechanizmy. Cieľom monitorovania podporeného AI/ML je minimalizovať objem nerelevantných dát, identifikovať potenciálne rizikové správanie v ranom štádiu a pripraviť kvalitné vstupy pre fázu detekcie a analýzy incidentov. AI/ML v tejto fáze podporuje najmä:
  - detekcia anomálií v log-och a v telemetrii v reálnom čase (sieťové logy, systémové a aplikačné logy, cloudová a endpoint telemetria; štatistické modely, unsupervised learning, baseline modely správania)
  - modelovanie normálneho správania systémov, služieb a sietí (časové vzory, objemové charakteristiky, frekvenčné a sezónne správanie)
  - behaviorálna analýza správania entít používateľov UEBA (User Entity Behavior Analytics) (profilovanie správania identít, zariadení a účtov; detekcia neštandardných aktivít a potenciálnych insider hrozieb)
  - automatizované filtrovanie a obohacovanie monitorovaných dát (normalizácia logov, redukcia šumu, kontextualizácia udalostí)
- **Detekcia:** Fáza detekcie nadväzuje na monitorovanie a predstavuje proces identifikácie a potvrdenia bezpečnostnej udalosti alebo incidentu. AI/ML

metódy sa v tejto fáze používajú na vyhodnocovanie podozrivých signálov, ktoré vznikli počas monitorovania, a na rozhodovanie, či pozorovaná aktivita predstavuje bezpečnostnú hrozbu. Cieľom fázy detekcie je presne identifikovať bezpečnostné incidenty, minimalizovať počet falošných pozitívnych hlásení a poskytnúť kvalitné vstupy pre následnú fázu triedenia a analýzy (triage).
AI/ML v tejto fáze podporuje najmä:

- klasifikáciu bezpečnostných udalostí a sieťovej prevádzky (rozlišovanie benignych udalostí, známych útokov a podozrivých aktivít; supervised learning, ensemble metódy)
- koreláciu udalostí naprieč viacerými zdrojmi dát (sieťové, aplikačné, endpointové a identitné dáta; kontextualizácia incidentu)
- systémy detekcie a prevencie narušenia IDS (Intrusion Detection System)/IPS (Intrusion Prevention System) založené na ML
- detekciu známych aj neznámych hrozieb bez použitia signatúr (behaviorálne modely, anomálie, heuristiky, zero-day útoky)

- **Triedenie a analýza:** Fáza triedenia a analýzy predstavuje prechod medzi detekciou a detailným vyšetrovaním incidentov. Jej cieľom je znížiť množstvo bezpečnostných upozornení, určiť ich relevanciu a prioritu a optimalizovať využitie analytických kapacít SOC. AI/ML metódy sa v tejto fáze uplatňujú najmä na automatizovanú podporu rozhodovania a redukciu manuálnej práce analytikov. Cieľom fázy triedenia a analýzy je zabezpečiť, aby sa analytické kapacity SOC sústredili na najkritickejšie incidenty, pričom menej relevantné alebo duplicitné udalosti sú automaticky potlačené alebo agregované.
  AI/ML v tejto fáze podporuje najmä:
  - stanovenie priorít upozornení alebo alertov (risk scoring, severity prediction, regresné a klasifikačné modely zohľadňujúce kontext, historické dáta a dopad na organizáciu)
  - zoskupovanie a korelácia podobných upozornení (clustering, deduplikácia, časová a kontextová korelácia udalostí)
  - redukcia falošných pozitívnych hlásení (false positives) (učenie sa z historických rozhodnutí analytikov, adaptívne prahové hodnoty)
  - obohacovanie incidentov o kontextové informácie (integrácia CTI, asset kritickosti, identít, geolokácie, reputačného skóre)
- **Vyšetrovanie**: Fáza vyšetrovania predstavuje hlbšiu analytickú a forenznú prácu, ktorej cieľom je potvrdiť incident, určiť jeho rozsah, príčinu, priebeh a dopad, a získať dôkazy pre nápravné opatrenia. AI/ML metódy sa v tejto fáze používajú najmä na koreláciu heterogénnych dát, rekonštrukciu útočnej reťaze a podporu analytika pri rozhodovaní, nie ako autonómna náhrada vyšetrovania.
  Cieľom fázy vyšetrovania je poskytnúť vysvetliteľný a reprodukovateľný analytický výstup (čo sa stalo, prečo, ako, s akým dopadom), ktorý umožní presnú reakciu a následnú prevenciu.
  AI/ML v tejto fáze podporuje najmä:
  - korelácia udalostí a rekonštrukciu priebehu incidentu (timeline a attack chain reconstruction) (spájanie udalostí naprieč logmi, endpoint telemetriou, sieťovou prevádzkou a identitami; identifikácia sekvencií typických pre TTP)
  - spracovanie prirodzeného jazyka NLP (Natural Language Processing) na koreláciu informácií o hrozbách (triáž a sumarizácia tiketov, CTI reportov, popisov incidentov, korelácia IOC/TTP z textu; extrakcia entít

a relevantných indikátorov)
    - analýza grafov pre účely vizualizácie útokov a vzťahov medzi entitami (identity–host–proces–sieťové spojenie–objekt; detekcia laterálneho pohybu, pivotovanie, vizualizácia útočných ciest)
    - detekcia podobnosti a zhlukovanie incidentov podľa vzorcov správania (case similarity, grouping podľa TTP/IOC; podpora opakovaného vyšetrovania a znalostnej bázy SOC)
- **Reakcia a náprava:** Fáza reakcie a nápravy nadväzuje na vyšetrovanie a jej cieľom je obmedziť dopad incidentu, obnoviť bezpečný stav systému a zabrániť jeho opakovaniu. AI/ML metódy sa v tejto fáze využívajú primárne ako podpora rozhodovania a automatizácie v rámci vopred definovaných procesov, nie ako autonómny náhradník ľudského rozhodovania. Cieľom tejto fázy je skrátiť čas reakcie (MTTR), zabezpečiť konzistentné a opakovateľné reakčné postupy a podporiť analytika pri rozhodovaní bez straty kontroly nad procesom.
  AI/ML v tejto fáze podporuje najmä:
    - automatizované reakčné stratégie prostredníctvom SOAR s podporou rozhodovania ML (izolácia endpointu, blokovanie IP adries, deaktivácia účtov, aktualizácia pravidiel; vykonávanie len v rámci schválených playbookov)
    - adaptívne prispôsobovanie reakčných stratégií (učenie sa z výsledkov predchádzajúcich zásahov, optimalizácia poradia a načasovania reakcií)
    - prediktívne odporúčania
- **Reakcia po incidente:** Fáza reakcie po incidente predstavuje retrospektívnu analytickú fázu, ktorej cieľom je pochopiť príčiny incidentu, vyhodnotiť účinnosť prijatých opatrení a zlepšiť pripravenosť SOC na budúce bezpečnostné hrozby. AI/ML metódy sa v tejto fáze využívajú na systematickú analýzu historických incidentov a podporu kontinuálneho zlepšovania bezpečnostných procesov. Cieľom fázy reakcie po incidente je zvýšiť odolnosť bezpečnostného prostredia, znížiť pravdepodobnosť opakovania podobných incidentov a zabezpečiť systematické učenie SOC z vlastných skúseností.
  AI/ML v tejto fáze podporuje najmä:
    - analýza základných príčin incidentov (root cause analysis) (identifikácia opakujúcich sa vzorcov, zraniteľností a slabých miest prostredníctvom zhlukovania, korelačnej a retrospektívnej analýzy)
    - vyhodnocovanie efektívnosti detekčných a reakčných mechanizmov (analýza metrík ako MTTD, MTTR, počet falošných pozitívnych/negatívnych hlásení, spätná väzba z incidentov)
    - predpovedanie trendov a evolúcie budúcich útokov (časové rady, porovnanie incidentov v čase, zmeny v TTP a útokových vzoroch)
    - podpora aktualizácie detekčných modelov, pravidiel a playbookov (využitie poznatkov z incidentov na adaptáciu ML modelov a operačných postupov)

Typické oblasti využitia AI/ML algoritmov a metód a ich aplikácie zahŕňa nasledovné:

- skvalitnenie informácií o hrozbách (Threat Intelligence Enhancement)
- skvalitnenie správy bezpečnostných informácií a udalostí SIEM (Security Information and Event Management (SIEM) Enhancement)

- detekcia a reakcia na koncové body EDR (Endpoint Detection & Reaction) (behaviorálna analýza procesov, detekcia laterálneho pohybu, automatizované zásahy v rámci schválených politík)
- bezpečnosť siete (analýza sieťovej prevádzky, detekcia anomálií, identifikácia C&C komunikácie, DoS/DDoS a laterálnych aktivít)
- detekcia phishingu a podvodov (klasifikácia e-mailov a správ, analýza obsahu a metadát, behaviorálne vzory komunikácie)
- detekcia vnútorných hrozieb (Insider Threats) (analýza správania používateľov a entít, korelácia identitných a systémových udalostí)
- automatizovaná reakcia na incident (SOAR, optimalizácia reakčných postupov, učenie sa z predchádzajúcich zásahov)
- správa zraniteľností (prioritizácia zraniteľností na základe kontextu prostredia, pravdepodobnosti zneužitia a historických incidentov).

Vzhľadom k tomu, že podrobný rozbor AI/ML algoritmov a metód je uvedený vo výstupnom dokumente V7 Výsledok analýzy vhodných algoritmov.docx, na tomto mieste len v skratke sumarizujeme najčastejšie používané prístupy a príklady použitia.

- učenie s učiteľom (Supervised Learning)
  - rozhodovacie stromy (Decision Tree), náhodné lesy (Random Forest), logistická regresia, metóda podporných vektorov SVM
  - detekcia malvéru, filtrovanie emailových správ typu spam, klasifikácia emailových správ typu phishing
- učenie bez učiteľa (Unsupervised Learning)
  - klastrovacie algoritmy k-means, HDBSCAN, izolovaný les (Isolation Forest), jedno-triedne SVM
  - detekcia útoku zvnútra (Insider Threat Detection), odhalenie neznámej hrozby (Unknown Threat Discovery)
- hlboké učenie (Deep Learning)
  - rekurentné neurónové siete RNN (Reccurent Neural Network), siete s pamäťou LSTM (Long Short-term Memory), konvolučné neurónové siete CNN (Convolutional Neural Networks), autoenkódery
  - detekcia pokročilých hrozieb APT
- spracovanie prirodzeného jazyka NLP (Natural Language Processing)
  - modely BERT, Transformer
  - automatizácia prijímania informácií o hrozbách, analýza údajov z dark web-u
- posilnené učenie (Reinforcement Learning)
  - Q-učenie, hlboké Q-siete DQN (Deep Q Networks)
  - dynamická úprava pravidiel firewall-u, automatizované penetračné testy
- metódy založené na grafoch
  - neurónové siete grafov GNN (Graph Neural Networks)
  - mapovanie šírenia útokov, detekcia podvodov

V rámci navrhovaného SOC riešenia predpokladáme využitie bezpečnostných log-ov získaných zo SIEM platforiem a ich súvisiacich zberných komponentov (ELK Stack, Wazuh alebo sieťové senzory typ Suricata) ktoré slúžia ako primárne zdroje dát pre našu analýzu a detekciu bezpečnostných udalostí. Výstupy navrhovanej analytickej vrstvy založenej na AI/ML metódach sú koncipované ako doplnkový analytický a hodnotiaci mechanizmus, ktorý nadväzuje na existujúce detekčné schopnosti SIEM systému. Cieľom tejto vrstvy nie je nahradiť SIEM, ale rozšíriť jeho funkcionalitu o pokročilé analytické schopnosti, najmä v oblasti

hodnotenia závažnosti incidentov, korelácie udalostí a identifikácie komplexných alebo neštandardných vzorcov správania.

Výsledky ML analýzy môžu byť využité na dodatočné posúdenie miery rizika a kontextu bezpečnostného incidentu alebo hrozby, ktorá bola predtým detegovaná existujúcimi mechanizmami SIEM, a tým podporiť rozhodovanie analytikov SOC v ďalších fázach spracovania incidentu.

V ďalšej časti dokumentu sa zameriavame na konkrétne kategórie bezpečnostných hrozieb, ktoré boli na základe analýzy prostredia a dostupných dát identifikované ako najvýznamnejšie z pohľadu navrhovaného inovatívneho SOC systému.

2 DETEKCIA SIEŤOVÝCH ANOMÁLIÍ

Detekcia behaviorálnych anomálií v počítačových systémoch je nevyhnutným krokom v mnohých obranných mechanizmoch.

Hlavným cieľom detekcie anomálií AD (angl. Anomaly Detection) je naučiť sa normálne správanie systému počas určitého časového obdobia a potom označiť akúkoľvek aktivitu, ktorá sa od tohto naučeného správania odchyľuje. V mnohých prípadoch si učenie sa normálneho správania vyžaduje monitorovanie a zaznamenávanie všetkých aktivít vykonávaných v systéme, aby sa extrahoval vzorec alebo profil pre každého používateľa/aplikáciu.

Vďaka svojej širokej použiteľnosti sa detektory anomálií úspešne používajú v mnohých oblastiach, ako sú napríklad monitorovanie sietí alebo operačných systémov.

Detektory anomálií sa zvyčajne skladajú z dvoch modulov, ktorými sú dátový kolektor a dátový analyzátor. Prvý modul (dátový kolektor) zhromažďuje dáta z protokolov/logov z monitorovaného systému, zatiaľ čo úlohou druhého modulu je analyzovať zhromaždené údaje. V závislosti od umiestnenia dátového kolektora existujú dva typy detektorov anomálií: sieťové (network-based) a hostiteľské (host-based). Tieto dva typy detektorov generujú rôzne typy protokolov a preto sa príznaky používané na detekciu anomálií líšia v závislosti od umiestnenia detektorov.

Nad zhromaždenými logmi daný dátový analyzátor vykonáva detekciu anomálií, pričom na túto úlohu sa používajú rôzne druhy metód, ktoré možno všeobecne rozdeliť do kategórií učenie s dohľadom, učenie s čiastočným dohľadom a učenie bez dohľadu. Viac informácií možno nájsť v dokumente V7 Výsledok analýzy vhodných algoritmov.docx. V rámci každej kategórie existuje množstvo rôznych použiteľných algoritmov. Okrem toho sú parametre/príznaky (features), ktoré je možné použiť pre AD, rôznorodé v závislosti od typov logov. Napríklad, zo sieťových paketov môže detektor použiť ako príznaky čas príchodu medzi paketmi alebo počet paketov v jednom smere. Z logov hostiteľa (host logs) môže detektor použiť vyťaženie CPU/pamäte. Príznaky sa taktiež môžu líšiť v závislosti od konkrétnej hrozby, proti ktorej je detektor anomálií nasadený. Napríklad detektor anomálií na detekciu ransomware používa parametre, ktoré sa líšia od parametrov používaných detektorom anomálií na detekciu cryptojacking.

Pri nasadzovaní systémov AD existuje niekoľko výziev. Hlavnou výzvou je, ako získať normálne vzorce správania sa. Je ťažké jednoznačne definovať, čo je to „normálne správanie“ a preto je zhromažďovanie takýchto údajov náročné. Ďalšou výzvou je, že detektor môže trpieť vysokým počtom falošne pozitívnych (false positive) výsledkov a teda vyvolať varovné hlásenie pre akýkoľvek neočakávaný vzor správania sa siete.

Neočakávané vzory bohužiaľ nemusia vždy znamenať, že došlo k narušeniu. Presnejšie povedané, môže existovať veľa neočakávaných normálnych vzorcov správania sa, napríklad v dôsledku preťaženia sietí alebo prekážok či problémov v systémoch. Preto je zníženie počtu falošne pozitívnych výsledkov kľúčové pre úspešné nasadenie systémov AD.

2.1 Detekcia DoS/DDoS útokov

Techniky DoS (Denial of Service) a DDoS (Distributed Denial of Service) sa vyskytujú pri rôznych typoch útokov vykonávaných v IP sieťach a pri incidentoch na rôznych úrovniach siete.

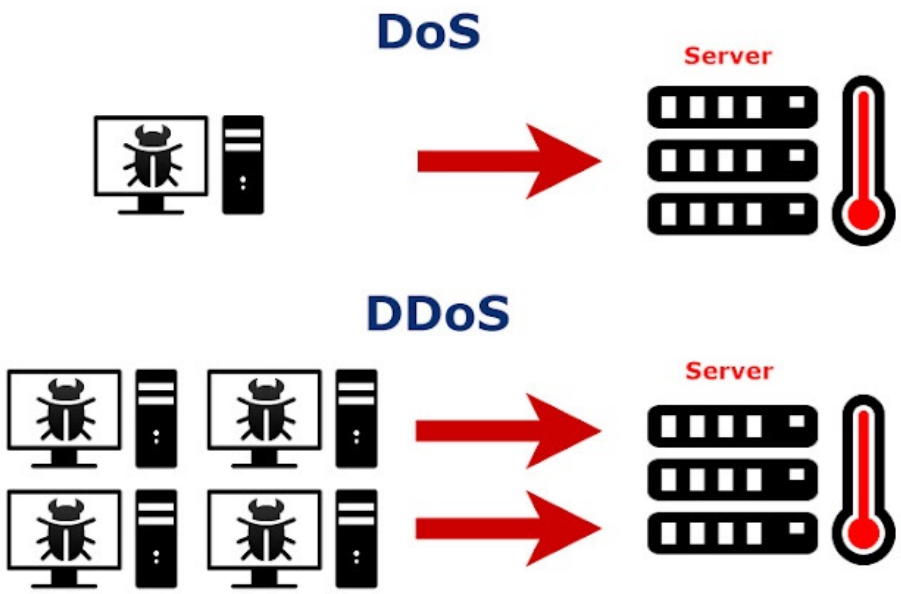
Skratka DoS v preklade znamená útok odmietnutia nejakej služby. Veľmi častým príkladom DoS útoku je zahltenie linky servera (záplava), CPU servera alebo jeho procesu/služby záplavou paketov alebo požiadaviek. V tomto prípade útočník využíva útočné

zariadenie, ktorým sa pokúša narušiť normálnu prevádzku cieľového servera alebo jeho služby či siete vedúcej k serveru. DoS sa veľmi rýchlo 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 spôsobiť pád aplikácie a teda DoS, aj keď primárne sa tento útok využíva na manipuláciu s vnútorným stavom programu, vloženie a spustenie vlastného kódu (tzv. shellcode) či získanie neoprávneného prístupu alebo eskalovanie privilégií.
- ICMP útok (angl. ICMP attack): ICMP je riadiaci protokol IP sietí ktorý využíva ICMP požiadavky (requesty) a odpovede (responses). Všeobecne sa echo-requesty a echo-reply používajú ako aplikácia, tzv. ping sieťového zariadenia, s cieľom overiť konektivitu medzi zariadením a odosielateľom. Ak by útočník zahltil cieľ ICMP 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 datagramov 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 datagram 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 sieťová vrstva hosta sa potom snaží spojiť 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 staršie verzie operačných systémov, ako napr. OS z 90tych rokov obsahujú chybu, pri spätnom zostavení fragmentovaného paketu stanica môže byť práve kvôli tejto zraniteľnosti napadnutý.

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 a ich individuálny tok nie je objemný. Hlavný a kľúčový rozdiel medzi DoS a DDoS útokom je, že pri DDoS s použitím botov môže útočník posielať obrovské objemy prevádzky.



Obr. 1: DoS a DDoS útok

DDoS útoky sa rozdeľujú do troch hlavných kategórií:

- **Objemový útok (angl. Volumetric attack):** Volumetrické útoky sú typom DDoS útokov, ktorých cieľom je vyčerpať dostupnú šírku pásma cieľa alebo jeho upstream poskytovateľa. Útočník generuje extrémne veľké množstvo sieťovej prevádzky (napr. UDP flood, ICMP flood), pričom môže — ale nemusí — využívať aj reflexné alebo amplifikačné mechanizmy prostredníctvom služieb ako NTP, DNS či SSDP. Moderné volumetrické útoky často dosahujú stovky Gb/s až jednotky Tb/s, čo môže viesť k úplnej nedostupnosti služby alebo zahlteniu transita ešte pred tým, ako pakety dorazia k cieľu.
- **Protokolový útok (angl. Protocol attack):** Protokolové útoky sú typom DDoS útokov, ktoré zneužívajú spôsob, akým sieťové protokoly a zariadenia spracúvajú prevádzku. Cieľom nie je zaplniť šírku pásma, ale vyčerpať stavové alebo systémové zdroje servera, firewallu, load balancera či iného sieťového prvku. Typickým príkladom je SYN flood, pri ktorom útočník generuje veľké množstvo TCP spojení, no nikdy ich nedokončí. Zariadenie tak musí držať veľké množstvo polootvorených stavov, až kým nevyčerpá svoju connection-tracking tabuľku. Medzi protokolové útoky patria okrem, SYN flood napríklad ACK flood / RST flood (stavové vyčerpanie firewallu), Ping of Death, Smurf attack, apod. Na rozdiel od volumetrických útokov, tieto typy útokov často nevyžadujú extrémnu šírku pásma, no dokážu efektívne paralyzovať infraštruktúru vyčerpaním jej vnútorných zdrojov
- **Útok na aplikačnú vrstvu (angl. Application layer attack):** Aplikačné útoky sa zameriavajú na konkrétnu službu alebo aplikáciu (napr. webový server, API) a snažia sa vyčerpať jej výpočtové zdroje, databázu alebo logiku aplikácie. Útočník generuje požiadavky, ktoré sú z pohľadu protokolu platné (napr. HTTP GET/POST), no pre server sú výpočtovo alebo databázovo náročné. Cieľom je, aby aplikácia nedokázala obslúžiť legitímnych používateľov, aj keď celková prenosová kapacita siete nie je vyčerpaná. Tieto útoky sa zvyčajne merajú v requests per second (RPS). Aplikačné útoky sú nebezpečné tým, že navonok vyzerajú ako legitímna prevádzka a ťažšie sa odhaľujú klasickými DDoS filtrami.

Existuje niekoľko znakov, ktoré nám môžu signalizovať prebiehajúci DDoS útok na sieť, aplikáciu alebo celú infraštruktúru. Medzi najčastejšie patria:

- Náhly a neobvyklý nárast sieťovej prevádzky (L3/L4), ktorý nezodpovedá bežným vzorom správania.
- Znížený výkon aplikácií alebo oneskorené odozvy, hoci interné zdroje servera nevyzerajú preťažené.
- Neschopnosť komunikovať so službou, prerušované spojenia alebo časté time-outy.
- Výrazný nárast požiadaviek na konkrétne zariadenia (L7), často v stovkách až tisícoch RPS, ktoré nepochádzajú od legitímnych používateľov.
- Abnormálne vyťaženie sieťových prvkov (firewall, load balancer, reverse proxy), napr. plné connection-tracking tabuľky alebo vysoké CPU.
- Náhle zahltenie linky poskytovateľa, čo sa prejaví stratovosťou paketov a zvýšeným oneskorením.

Tieto znaky samy osebe nemusia potvrdiť útok, ale sú silným signálom na okamžité overenie prostredníctvom NetFlow/IPFIX, firewall štatistík, monitoringových nástrojov alebo telemetrie.

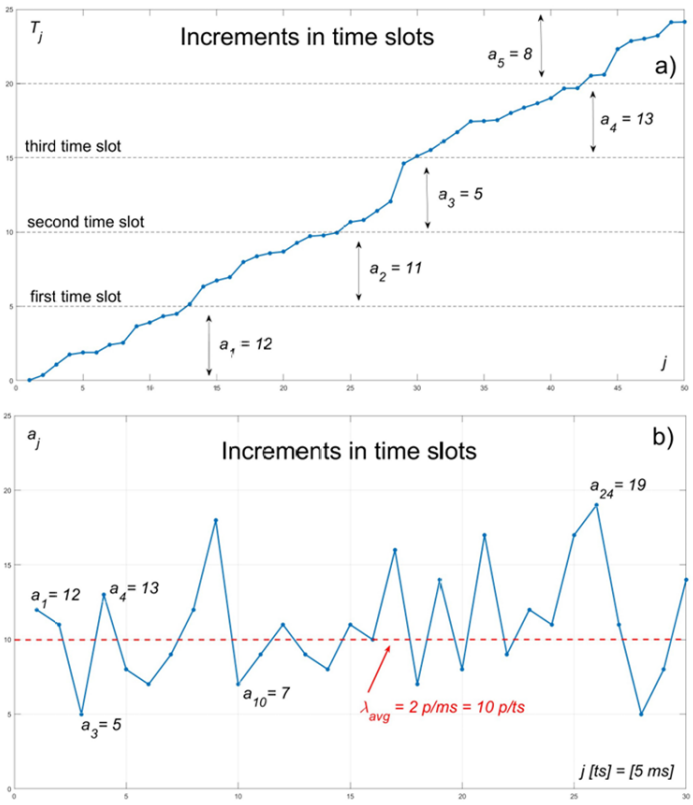
2.1.1 DoS útoky - datasety, experimenty a výsledky, popis vybraného algoritmu

Pre naše experimenty sme použili množstvo rôznych dát získaných z viacerých zdrojov, ako napr. ISCX 2012 alebo CIC-IDA 2017, ale hlavným zdrojom boli naše vlastné dáta získané zo sieťovej prevádzky nielen v rámci FRI UNIZA, ale aj celej siete Žilinskej univerzity.

V deterministických aj stochastických modeloch IP siete sa na analýzu IP toku používa kumulatívny proces A(s, t), ktorý popisuje výskyt paketov v časovom intervale ⟨s, t⟩. Pre modely s diskrétnym časom sú na analýzu dané určité časové úseky [ts] alebo vzorkovacie okná, v ktorých sa zaznamenáva počet vyskytujúcich sa paketov ai (prírastky alebo inkrementy v čase i):

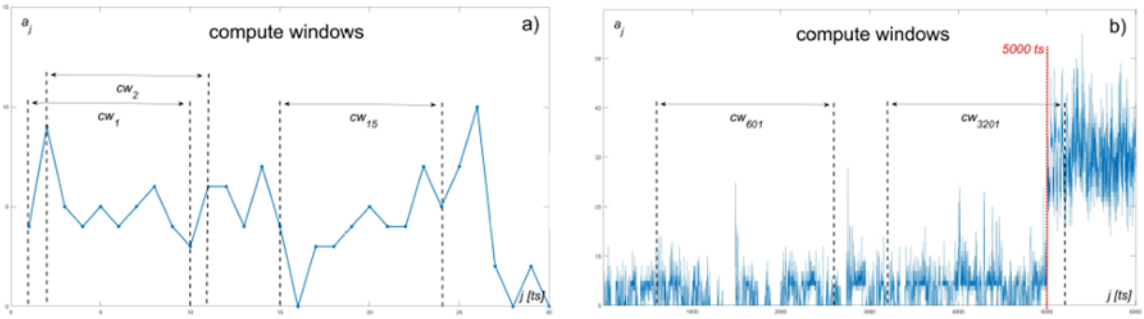
$$\forall t,s \in R^{+}; \quad A(s,t) = A(0,t) - A(0,s) = \sum_{i=s}^{t} a_i$$

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 parametrov, pri cw = 2048 sa významne zväčšila doba výpočtu parametrov, hlavne Hurstovho exponentu.



Obr. 2: Popis IP toku: a) kumulatívny čas, b) prírastky v časových oknách

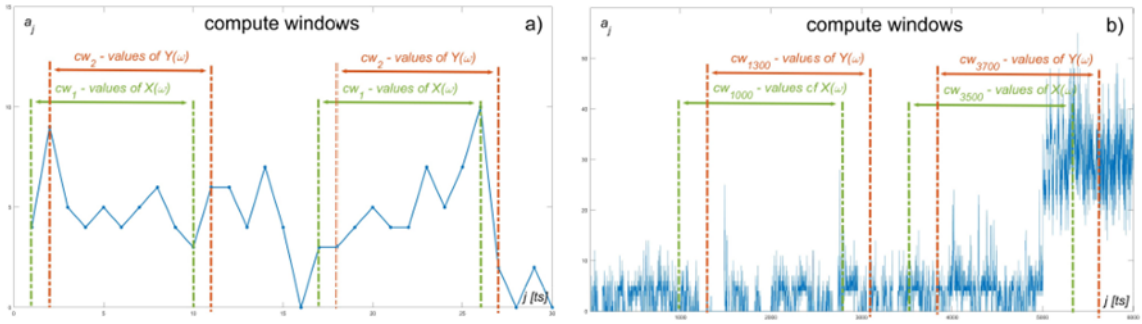
Pre popis pravdepodobnostného charakteru pozorovaného IP toku sme vybrali štatistické koeficienty, ktoré môžeme rozdeliť na dva typy, jednoduché – single, a dvojité – double. Jednoduché koeficienty nadobúdajú svoju hodnotu iba z jedného okna (computer window) nameraných prírastkov paketov, resp. z jedného z jedného náhodného výberu danej náhodnej premennej. Ďalšiu hodnotu získame posunutím cw okna o niekoľko time slotov (shift). Obvykle sme použili iba jeden shift. Postupným posúvaním cw získavame tak hodnoty kĺzavých štatistických koeficientov. Medzi tieto koeficienty patria: priemer, výberový rozptyl, koeficienty variabilnosti, šikmosti a špicatosti, Hurstov exponent a entropia.



Obr. 3: Okná pre výpočet parametrov: a) |cw| = 10 ts, b) |cw| = 2000 ts.

Dvojité štatistické koeficienty potrebujú pre určenie svojej hodnoty dve výpočtové okná, resp. náhodné výbery dvoch náhodných premenných. Z pohľadu matematickej štatistiky ide o koeficienty, ktoré istý spôsobom porovnávajú pravdepodobnostné rozdelenie dvoch

náhodných premenných. Použili sme koeficient korelácie, autoregresný koeficient a divergenciu.

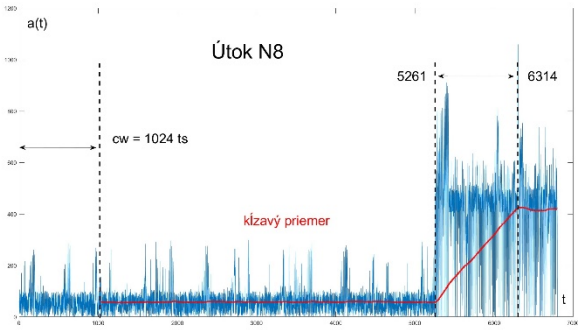


Obr. 4: Pár okien pre výpočet parametrov: a) |cw| = 10 ts, posun 1 ts, b) |cw| = 2800 ts, posun 300 ts.

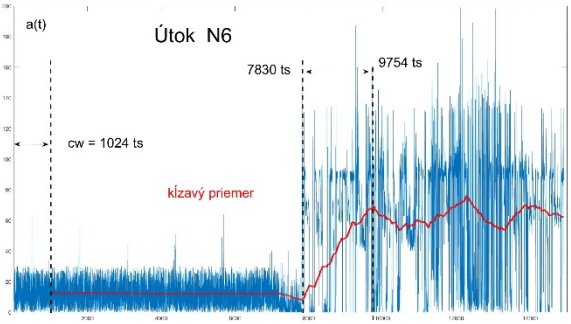
Na rôznych záznamoch DDoS útokov budeme prezentovať reakciu vybraných štatistických parametrov. Vynechali sme množstvo experimentov, ktoré sme realizovali na rôznych simulovaných scenároch. Na týchto scenároch vybrané štatistické parametre fungovali „vzorovo“ a my sme nadobudli prvotnú predstavu o efektívnosti 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.

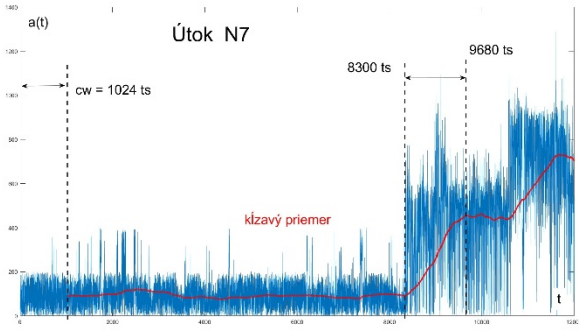
Na grafoch záznamov DDoS útokov sme zobrazili aj priebeh kĺzavého priemeru a zobrazili časový interval, v ktorom kĺzavý priemer m(t) nadobudol od začiatku útoku svoje lokálne maximum. Tento interval využijeme pri subjektívnom hodnotení ostatných štatistických koeficientov. Na všetkých záznamoch DDoS útokov môžeme pozorovať lineárny rast kĺzavého priemeru. Hľadáme koeficienty, ktorým pri začiatku útoku začnú rásť hodnoty rýchlejšie než lineárne. Detailnejšie informácie ohľadom jednotlivých útokov možno nájsť v podkladovom dokumente Machine Recognition of DDoS Attacks Using Statistical Parameters.pdf



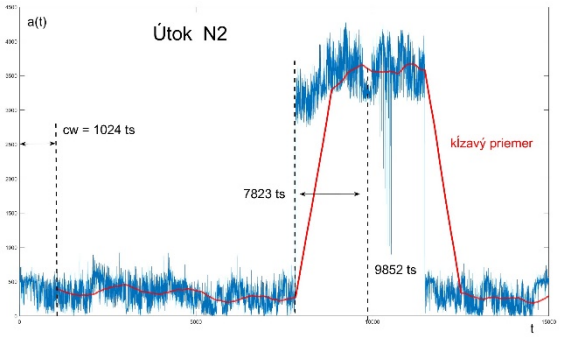
Obr. 5: Útok N8 a kĺzavý priemer



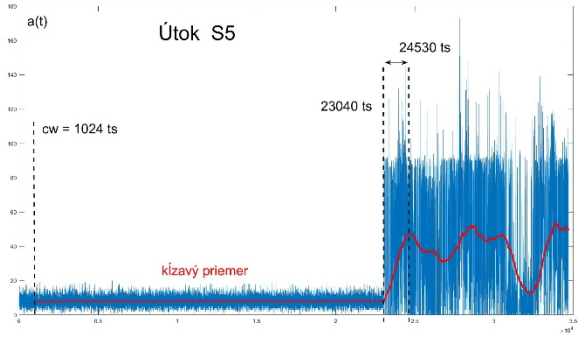
Obr. 6: Útok N6 a kĺzavý priemer



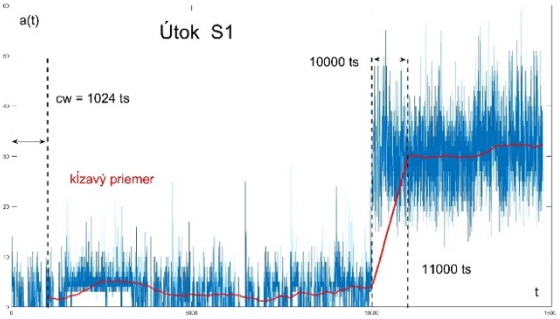
Obr. 7: Útok N7 a kĺzavý priemer



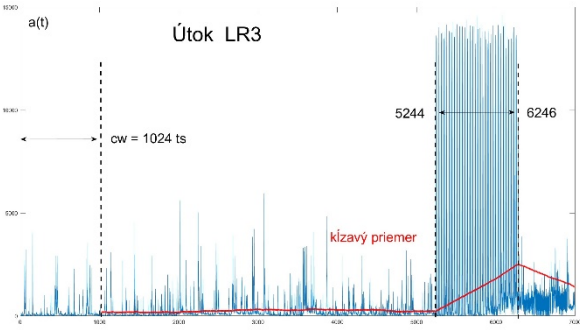
Obr. 8: Útok N2 a kĺzavý priemer



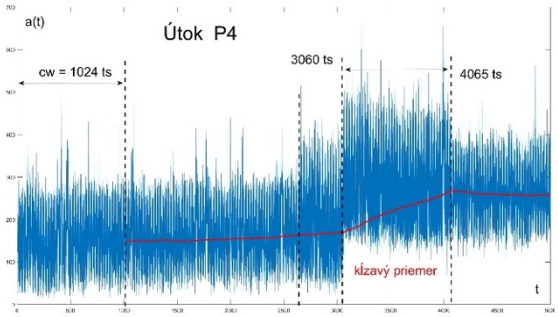
Obr. 9: Útok S5 a kĺzavý priemer



Obr. 10: Útok S1 a kĺzavý priemer



Obr. 11: Útok LR3 a kĺzavý priemer



Obr. 12: Útok P4 a kĺzavý priemer

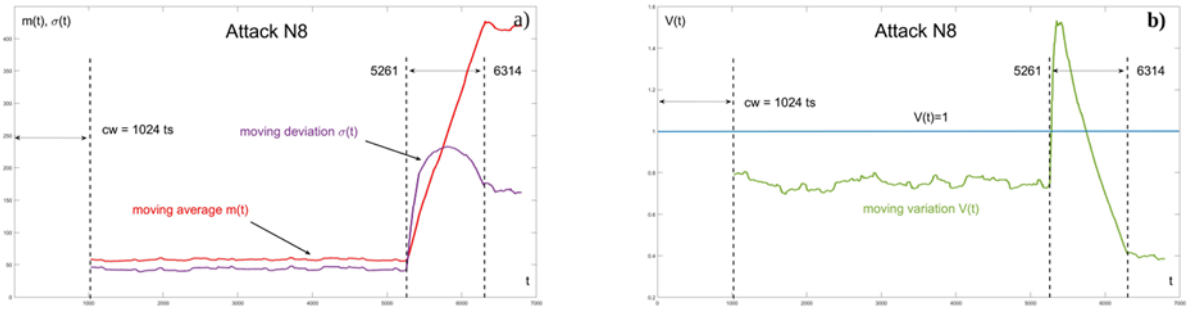
2.1.1.1 Štatistické parametre

2.1.1.1.1 1-oknové parametre

A. Koeficient variácie

Na základe našich skúseností s modelovaním reálnej IP prevádzky môžeme predpokladať, že bežný štandardný IP tok, hlavne v chrbticovej sieti alebo v prípadoch agregovania viacerých IP služieb vykazuje charakter stacionárneho stochastického procesu s nezávislým výskytom paketov, ktorých distribúcia predstavuje jedno-modálne rozdelenie pravdepodobnosti MMP (Markovov modulovaný proces) charakteru. V prípade nástupu DDoS útoku a takýto „prirodzený " pravdepodobnostný charakter štandardnej agregovanej prevádzky podstatne naruší. Vo väčšine prípadov samozrejme nastane výrazné zvýšenie priemernej a špičkovej intenzity ale zároveň sa naruší prirodzený tok dát umelo generovanými paketmi, čo väčšinou spôsobí výrazný pokles rozptylu paketov. Túto zmenu monitoruje koeficient variabilnosti V:

$$\lambda_{avg} = \frac{1}{N}\sum_{i=1}^{N} a_i, \qquad S^2 = \frac{1}{N-1}\sum_{i=1}^{N}(a_i - \lambda_{avg})^2, \qquad V = \frac{S}{\lambda_{avg}}$$



Obr. 13: Útok N8: a) Kĺzavý priemer a odchýlka vzorky, b) kĺzavá variácia vzorky

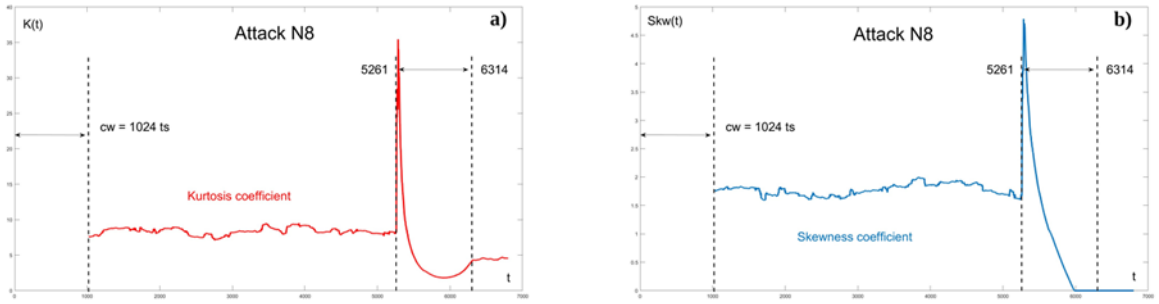
Pri väčšine pozorovaných záznamov, cca 75%, dokázal koeficient variabilnosti rozpoznať nástup DDoS útoku v prvých time-slotoch prekročením hodnoty V=1. Problém mal koeficient pri Low-rate útokoch, kde threshold s hodnotou 1 nebol použiteľný.

B. Šikmosť a špicatosť

Šikmosť (Skewness) a špicatosť (Curtosis) sú ďalšie štatistické koeficienty, ktoré popisujú tvar rozdelenia pravdepodobností v danom výpočtovom okne.

$$K = \frac{\frac{1}{N-1}\sum_{i=1}^{N}(a_i - \lambda_{avg})^3}{\left[\frac{1}{N-1}\sum_{i=1}^{N}(a_i - \lambda_{avg})^2\right]^{3/2}}, \qquad S_{kw} = \frac{\frac{1}{N-1}\sum_{i=1}^{N}(a_i - \lambda_{avg})^4}{\left[\frac{1}{N-1}\sum_{i=1}^{N}(a_i - \lambda_{avg})^2\right]^2}$$

V prípade načítania už aj malého množstva útočnej IP prevádzky predpokladáme výraznú zmenu jedno-modálnej symetrie rozdelenia paketov v danom výpočtovom okne.



Obr. 14: Útok N8: a) Koeficient pohyblivej špicatosti, b) Koeficient pohyblivej šikmosti.

Koeficienty *K* a *Skw* sú odhady tretieho a štvrtého centrálneho momentu náhodnej premennej *X*, preto je priebeh ich hodnôt veľmi podobný. Vo väčšine analyzovaných záznamov reagovali na začiatok DDoS útoku niekoľkonásobným nárastom (špičkou) hodnôt v porovnaní s predchádzajúcim priebehom, avšak oproti koeficientu variability neposkytli nejakú hodnotu thresholdu na detekciu, preto na rozpoznanie výrazného peaku v okamihu útoku bude treba použiť ďalšie detekčné metódy.

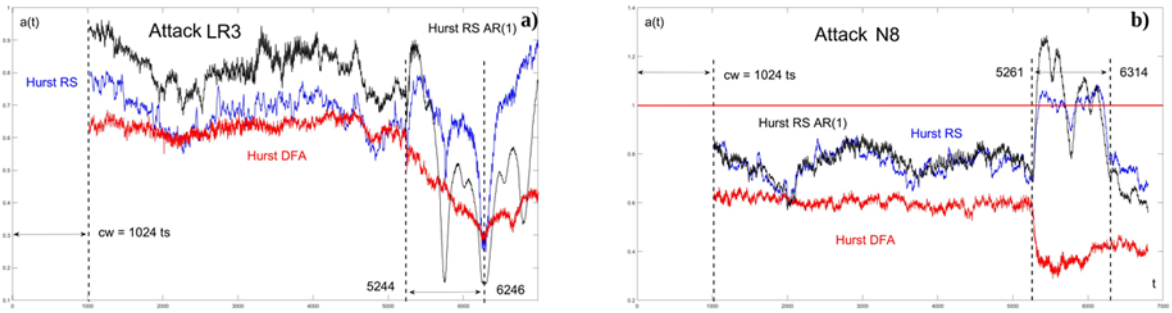
C. Hurstov exponent

Pri testovaní reakcií Hurstovho exponenta na simulované DDoS scenáre a na prvé reálne zaznamenané dáta sme získali zaujímavé výsledky. Exponent mal tendenciu „vyskočiť“ na hodnoty blízke H = 1 na začiatku útoku. Zároveň počas spracovania štandardnej prevádzky zostával v rozmedzí 0,4 až 0,6, čo zodpovedá hodnotám stacionárneho náhodného procesu. Z toho vyplynula možnosť využitia Hurstovho exponenta na strojové rozpoznávanie útoku, ak dôjde k prekročeniu hodnoty blízkej H = 1.

Nevýhodou Hurstovho exponentu je, že koeficient nie je možné vypočítať analyticky, môžeme ho iba štatisticky odhadnúť. Na odhad sa používa viacero metód, medzi hlavné patria: R/S štatistika, metóda agregovanej variability, metóda absolútnej hodnoty, variancia reziduí, Higuchiho metóda, modifikovaná Allanova variancia, metóda zmeny mierky okna, Whittleho odhadovač a ďalšie.

V základe našich skúseností sme sa rozhodli použiť odhady pomocou R/S analýzy a DFA - Detrended Fluctuation Analysis. Jednotlivé postupy odhadu neuvádzame kvôli ich zložitosti – viac informácií možno nájsť v podkladovom dokumente Machine Recognition of DDoS Attacks Using Statistical Parameters.pdf. Tiež sme použili modifikovaný odhad Hurstovho exponenta pomocou R/S analýzy, pričom sme najskôr odstránili lineárny autoregresný trend z inkrementálnych hodnôt.

Na nasledujúcich obrázkoch uvádzame dva prípady, pri ktorých sa podľa nášho subjektívneho hodnotenia reakcia z hľadiska strojového rozpoznávania ukázala ako veľmi neuspokojivá a úplne ideálna. Pri spracovaní záznamu LR3 žiadny z odhadov Hurstovho exponenta nereagoval významne na začiatok DDoS útoku. V prípade N8 všetky tri odhady reagovali výrazne, pričom RS odhady prekročili hodnotu H = 1.



Obr. 15: Hurstov exponent: a) Útok LR3, b) Útok N8

Celkovo môžeme povedať, že pri odhade pomocou DFA hodnoty H-exponenta v momente útoku väčšinou výrazne klesli. Pri R/S odhade bol priebeh hodnôt často nevýznamný z hľadiska strojového rozpoznávania. Avšak po odstránení lineárnej autoregresnej závislosti hodnoty H-exponenta výrazne vzrástli. V niektorých prípadoch hodnoty exponenta v momente útoku dokonca prekročili hodnotu H = 1.

2.1.1.1.2 2-oknové parametre

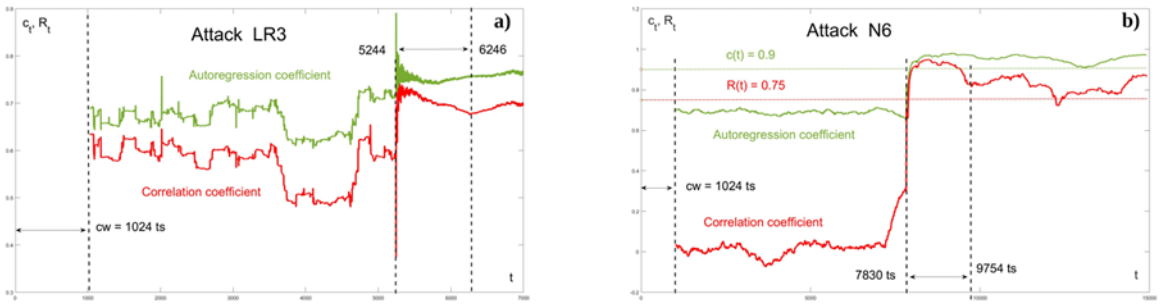
A. Autoregresný a korelačný koeficient

Autoregresný a korelačný koeficienty sú veľmi podobné pravdepodobnostné charakteristiky, ktoré vyjadrujú určitý druh závislosti medzi dvoma náhodnými premennými, v našom prípade medzi premennými, ktorých realizácie predstavujú prírastky IP tokov a_i v prekrývajúcich sa výpočtových oknách.

Korelačný koeficient R predstavuje škálovanú kovarianciu, pričom sa používa smerodajná odchýlka jednotlivých náhodných premenných. Autoregresný koeficient c predstavuje lineárnu závislosť v autoregresnom modeli AR(1).

$$c = \frac{\sum_{i=1}^{N-1} a_i^2}{\sum_{i=1}^{N-1} a_i a_{i+1}}, \qquad R = \frac{\sum_{i=1}^{N} (a_i - m_i)(a_{i+1} - m_{i+1})}{\left[\sum_{i=1}^{N} (a_i - m_i)^2\right]^{1/2} \left[\sum_{i=1}^{N} (a_{i+1} - m_{i+1})^2\right]^{1/2}}$$

Z tvaru výpočtu odhadov týchto dvoch parametrov vyplýva ich podobný priebeh. Opäť sme vybrali dva extrémne záznamy – LR3 a N6 (obrázok 16).



Obr. 16: Autoregresný a korelačný koeficient: a) Útok LR3, b) Útok N6

Pri väčšine záznamov v okamihu nástupu DDoS útoku hodnoty koeficientov začali významne rásť k hodnote 1. Vďaka tomu je možné nastaviť určitú hraničnú hodnotu (threshold)

pre oba koeficienty, ktorej prekročenie by signalizovalo DDoS útok.

Na základe vykonaných experimentov sme stanovili hranice:

- pre pohyblivý autoregresný koeficient: c(t) = 0,9
- pre pohyblivý korelačný koeficient: ϱ(t) = 0,75

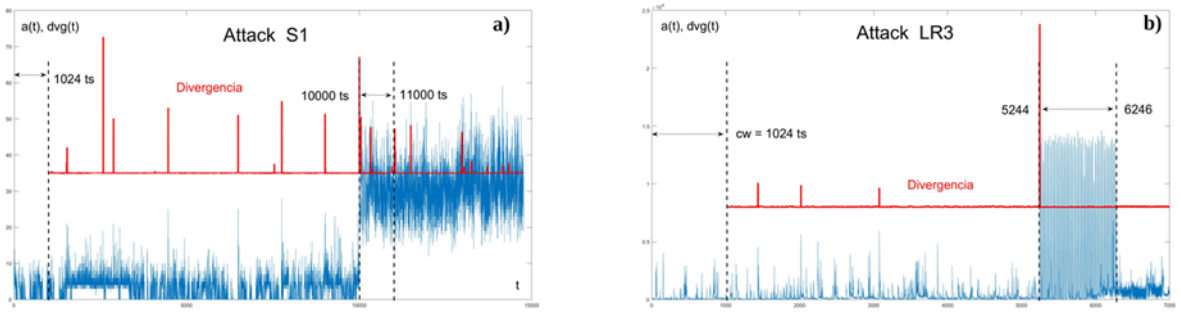
Pre väčšinu analyzovaných záznamov, okrem LR3 a P4, možno takto nastavené prahy použiť na strojovú detekciu útokov.

B. Kullback-Leiblerova divergencia

Ďalším štatistickým koeficientom, ktorý porovnáva rozdelenie pravdepodobnosti v dvoch prekrývajúcich sa výpočtových oknách, je Kullback–Leiblerova divergencia (KD). Divergencia určenie, ako sa jedna pravdepodobnostná distribučná funkcia (P) líši od inej distribučnej funkcie (Q). Pre výpočet divergencie KD a jej odhadu dvg platia nasledujúce vzťahy:

$$KD(P\|Q) = \sum_{k=0}^{m} P(k) \ln \frac{P(k)}{Q(k)}, \qquad dvg = \sum_{k=0}^{m} \left[\frac{n_k}{N}\right] \ln \left[\frac{n_k}{m_k}\right]$$

Divergencia reagovala na väčšinu záznamov vysokým impulzom okamžite pri nástupu útoku. Avšak mala tendenciu reagovať rovnakým spôsobom aj na špičky v normálnej prevádzke (traffic), čo by mohlo spôsobiť mnoho falošných hlásení počas detekcie. Napríklad pri útoku S1 je impulz počas útoku nerozlíšiteľný od impulzov počas normálnej prevádzky. Pri útoku LR3 je priebeh divergencie úplne ideálny pre strojové rozpoznávanie.



Obr. 17: Kullback-Leiblerova divergencia: a) Útok S1, b) Útok LR3.

Ďalšou nevýhodou použitia divergencie je skutočnosť, že vysoký impulz na začiatku útoku trvá veľmi krátko v porovnaní s inými parametrami, takže jej použitie samostatne považujeme za problematické. Preto sme použitie divergencie odložili na ďalší výskum.

2.1.1.1.3 Účinnosť používania štatistických koeficientov

V nasledujúcej tabuľke sme zhrnuli hodnotenie reakcie skúmaných štatistických koeficientov na analyzované záznamy DDoS útokov. Z troch skúmaných odhadov Hurstovho exponenta sme vybrali odhad pomocou R/S analýzy s odstránením lineárneho autoregresného trendu, ktorý považujeme za najefektívnejší.

Attack	$V(t)$ 1.00	$K(t)$	$Skw(t)$	$H(t)$ 1.00	$c(t)$ 0.90	$R(t)$ 0.75	$dvg(t)$
N2	0	PT	PT	9	PT	PT	4
N6	0	PT	PT	0	PT	PT	5
N7	0	x	x	1	PT	PT	4
N8	0	PT	PT	0	PT	PT	0
S1	5	x	x	6			8
S5	0	PT	PT	0	PT	PT	0
LR3	PT	PT	PT	x	PT	PT	3
P4	x	x	x	PT	x	x	x

Tab. 1: Účinnosť štatistických koeficientov, koeficient variancie V(t), špicatosť K(t), šikmosť Skw(t), Hurstov exponent H(t), autoregresný koeficient c(t), korelačný koeficient R(t), Kullback-Leiblerova divergencia dvg(t).

Zelené bunky znamenajú, že útok bol rozpoznaný prekročením prahových hodnôt daných parametrov. Hodnoty prahov sú uvedené v pravej kolónke tabuľky. Číslo v bunke udáva počet falošných hlásení a počet prekročení limitnej hodnoty ešte pred samotným útokom. V prípade divergencie toto číslo predstavuje počet impulzov pred skutočným útokom, resp. počet falošných útokov. Označenie „PT“ znamená, že predpokladáme, že útok by bolo možné rozpoznať pomocou predikčných metód (PT – predicting tunnel).

Označenie „*x*“ znamená, že daný parameter nie je použiteľný pre daný záznam.

Na základe vykonaných experimentov môžeme parametre rozdeliť do dvoch skupín:

1. Prvá skupina – parametre, pri ktorých môžeme použiť prahovú hodnotu (threshold) pre detekciu útoku: V(t), H(t), c(t), R(t)
2. Druhá skupina – parametre, pri ktorých je potrebné použiť predikčné metódy na detekciu útoku: K(t), Skw(t), c(t) a R(t).

Autoregresné a korelačné koeficienty patria do obidvoch skupín. Divergenciu sme kvôli krátkemu trvaniu impulzu počas útoku a zároveň častým falošným reakciám na špičky v bežnej prevádzke predbežne vylúčili z ďalšieho skúmania.

2.1.1.2 3σ-Tunel

Spôsob, ako strojovo rozpoznať náhlu výraznú zmenu priebehu hodnôt daného štatistického parametra pri nástupe DDoS útoku je založená na predstave vytvorenia tzv. predikčného tunela. Pôvodne sme sa zaoberali využitím polynomickej regresie, autoregresnej analýzy a Fourierovej transformácie pre predikciu nových hodnôt daného štatistického koeficienta, no nakoniec sa osvedčil najjednoduchší spôsob, použitie aritmetického priemeru a smerodajnej odchýlky. Pre daný štatistický koeficient θ určíme predikčné okno s veľkosťou |pw| = N. Z hodnôt θ_t parametra v tomto okne vypočítame priemernú hodnotu θ a smerodajnú odchýlku σ_θ. Okolo priemeru θ vytvoríme interval

$$I = [\theta - n\sigma_\theta,\ \theta + n\sigma_\theta]$$

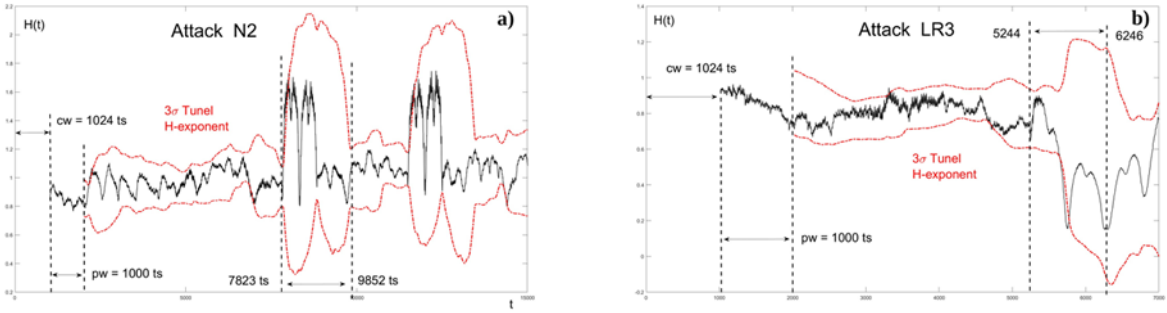
Následne otestujeme, či nová hodnota parametra patrí do tohto predikčného intervalu, teda či θ_{N+1} ∈ I. Ak nepatrí, systém deteguje začiatok útoku. V opačnom prípade posunieme predikčné okno pw o jednu hodnotu θ a celý proces opakujeme:

$$\bar{\theta} = \frac{1}{N}\sum_{i=1}^{N}\theta_i, \qquad \sigma^2 = \frac{1}{N-1}\sum_{i=1}^{N}(\theta_i - \bar{\theta})^2 \qquad \text{Test:} \quad \theta_{N+1} \in \langle \bar{\theta} - n\sigma, \bar{\theta} + n\sigma \rangle$$

Na základe experimentov s rôznou šírkou tunela a veľkosťou predikčného okna môžeme konštatovať, že 3σ-tunel s |pw| = 1000 ts má dostatočne „dlhú pamäť", aby dokázal pokryť lokálnu variabilitu (jitter) štatistického koeficienta bez spôsobovania veľkého počtu falošných hlásení. Zároveň má takto nastavený predikčný tunel dostatočne „dlhú pamäť" na to, aby dokázal reagovať na významné zvýšenie hodnôt daného koeficienta na začiatku DDoS útoku.

Na nasledujúcich obrázkoch ukážeme správanie 3σ-tunela na parametri s najväčšou variabilitou spomedzi uvažovaných štatistických koeficientov, konkrétne na Hurstovom exponente (odhadnutom pomocou R/S štatistiky s odstránením autoregresného trendu).

Na obrázku 18a je zobrazený ideálny prípad, keď detekcia nastala pri načítaní tretieho časového slotu s útočnou prevádzkou (traffic) a nevyskytli sa žiadne falošné hlásenia. Na obrázku 18b útok nebol detegovaný a objavilo sa jedno falošné hlásenie.



Obr. 18: 3σ-tunel pre Hurstov exponent (bez RS): a) Útok N2, b) Útok LR3

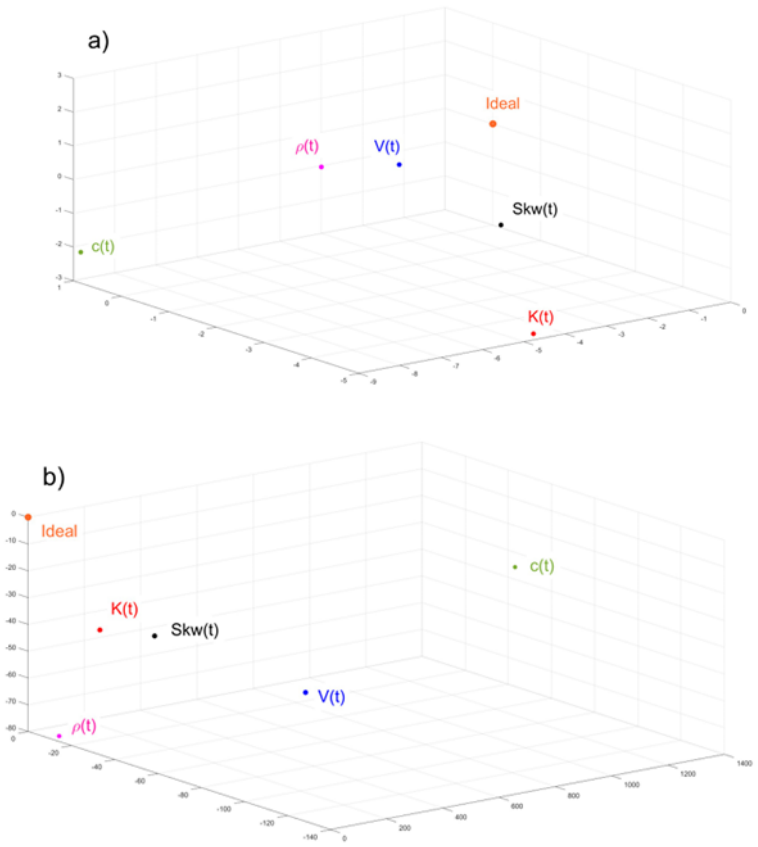
Výpočet Hurstovho exponenta je výrazne časovo náročnejší ako výpočet ostatných štatistických parametrov, preto sme jeho použitie na detekciu dočasne odložili.

V tabuľke 2 uvádzame výsledky detekcie pomocou 3σ-tunela, aplikovaného na vybrané štatistické koeficienty. Symbol F/R predstavuje počet falošných hlásení (F) a čas detekcie útoku (R) v časových slotoch (ts). Napríklad prvá hodnota v tabuľke 2/30 znamená, že pri použití 3σ-tunela aplikovaného na variačný koeficient V(t) bol útok v zázname S1 rozpoznaný do 30 ts od jeho začiatku, pričom predtým sa vyskytli dve falošné hlásenia.

Attack	$V(t)$	$K(t)$	$Skw(t)$	$c(t)$	$\varrho(t)$
S1	2/30	1/5	2/6	1/9	1/9
N2	1/116	5/24	2/47	0/1	1/1
LR3	0/1	2/1	1/1	1/1	1/8
S5	4/18	4/20	3/42	7/1394	6/5
N6	3/1	2/4	1/10	3/17	5/13
N7	1/51	2/40	2/40	1/90	2/76
N8	0/2	1/1	0/1	5/2	0/3

Tab. 2: Detekcia pomocou 3σ-tunela aplikovaná na: variabilitu V(t), šikmosť K(t), špicatosť Skw(t), autoregresiu c(t) a koreláciu ϱ(t).

Parameter R (čas detekcie útoku) považujeme za dôležitejší než parameter F (počet falošných hlásení), pretože vhodnou kombináciou štatistických koeficientov je možné počet falošných hlásení eliminovať. Z tohto dôvodu budeme používať oba parametre na porovnanie efektívnosti jednotlivých koeficientov. Na vizualizáciu podobnosti v 3D zobrazení využijeme metódu PCA (Principal Component Analysis).



Obr. 19: 3D vizualizácia hodnôt parametrov z Tab. 2: a) F (False) – falošné hlásenie, b) R (Recognize) – rozpoznaný útok

Účinnosť jednotlivých štatistických koeficientov je určená na základe vzdialenosti od nulového vektora (pozri tabuľku 3). Pre parameter F (falošné hlásenia) nulový vektor predstavuje nulový počet falošných hlásení pre každý záznam prevádzky a pre parameter R (rozpoznanie útoku) predstavuje rozpoznanie útoku ešte pred jeho začiatkom.

Parameters	$V(t)$	$K(t)$	$Skw(t)$	$c(t)$	$\varrho(t)$
F (False)	0.0173	0.0026	0.0056	1.9517	0.0064
R (Recogn.)	28.05	54.92	19.07	85.35	78.92

Tab. 3: Účinnosť použitia koeficientov pri falošných hláseniach a rozpoznaných útokoch

V prípade falošných hlásení je poradie koeficientov od najúčinnejšieho nasledovné:

K(t), Skw(t), ϱ(t)

Poradie podľa rýchlosti detekcie útoku je nasledovné:

Skw(t), V(t), K(t)

Najhoršie výsledky dosiahol autokorelačný koeficient c(t).

2.2 Záver

V riešení projektu sme otestovali možnosti rozpoznania DDoS útokov pomocou rôznych štatistických koeficientov. Testy sme previedli nie len na štandardných útokoch, cca 75%, ale aj na problematických záznamoch, kde koeficienty reagovali prekvapivo. V správe uvádzame na porovnanie obidva prípady. Ďalšie záznamy experimentov sú uvedené v podkladovom dokumente Machine Recognition of DDoS Attacks Using Statistical Parameters.pdf.

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 nasledovné metódy, ktoré sú popísané vyššie:

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

Pri niektorých koeficientoch sa dá určiť horná hranica, určitá prahová hodnota (threshold), ktorej prekročenie je signál nástupu útoku. Toto je ideálny prípad pre strojové rozpoznanie útoku (V(t), H(t), c(t), R(t).). Pri ďalších parametrov došlo k detekcii nástupu útoku vysokým impulzom (K(t), Skw(t), c(t) a R(t).), ktorý je pre človeka evidentný, avšak problém nastáva naučiť rozpoznať tento impulz stroj a nezameniť ho s bežnou špičkovou hodnotou (impulz-peak) v štandardnej prevádzke. Pre tieto účely sme navrhli a otestovali tzv. predikčný tunel, ktorý s rôznou úspešnosťou rôznym parametrom umožňuje strojovo rozpoznať impulz pri nástupu útoku a zároveň potláča falošné hlásenia pri bežných prevádzkových peak-och.

Ďalší krok výskumu bude predstavovať návrh a otestovanie detekčných funkcii, ktoré zohľadnia poznatky získane pri testovaní štatistických parametrov.

Z pohľadu architektúry SOCaaS majú tieto výsledky niekoľko praktických dôsledkov:

- **Detekcia DDoS musí byť viacvrstvová.** Kombinácia štatistických parametrov, prahovej detekcie a predikčných tunelov poskytuje dostatočnú robustnosť na prvotné odhalenie anomálií, ktoré môžu indikovať útok.
- **Štatistická detekcia je nenáročná a vhodná pre real-time monitoring.** Výpočet sledovaných parametrov nie je výpočtovo náročný, čo umožňuje ich nasadenie v reálnom čase aj vo viacklientskom (multitenantnom) SOCaaS prostredí, kde sú nároky na škálovanie kritické.
- **Tieto parametre sú vhodné ako „early warning“ vrstva pred ML modelmi.** SOC riešenie môže využívať kombináciu:
  - prvotná štatistická detekcia ⇒ rýchla identifikácia náhlej zmeny,
  - následné ML/AI modely ⇒ presnejšia klasifikácia, zníženie falošných poplachov.

  Štatistické koeficienty tak predstavujú lacný a rýchly trigger pre hlbšiu analýzu.
- **Zohľadňujú multitenantné správanie.** Keďže jednotlivé tenantské siete majú odlišný normál, predikčný tunel umožňuje učiť „baseline“ pre každého zákazníka nezávisle. To je zásadný predpoklad pre akékoľvek centralizované SOC riešenie.
- **Limitácie pri nízkofrekvenčných (Low-Rate) útokoch ostávajú.** Niektoré útoky sú

zámerne navrhnuté tak, aby napodobnili legitímnu prevádzku. V týchto prípadoch je potrebné doplniť ďalšie vrstvy detekcie – tokové analýzy, behaviorálne modely alebo korelácie udalostí.

Celkovo možno konštatovať, že štatistické parametre IP prevádzky predstavujú robustný základ pre rýchlu detekciu DoS/DDoS útokov v SOCaaS ekosystéme. Ich kombinácia s predikčnými metódami poskytuje viacúrovňový systém včasného varovania, ktorý je ľahko škálovateľný, má nízke nároky na spracovanie dát a umožňuje dôveryhodné rozpoznávanie aj pri vysokej variabilite sieťovej prevádzky. Tieto poznatky sú prínosné pre návrh autonómnych detekčných modulov v modernom SOC, ktoré potrebujú rýchlo a spoľahlivo reagovať na zmeny v správaní siete bez zásahu operátora.

3 DETEKCIA PHISHINGOVÝCH E-MAILOV A SPOOFINGU

Phishing je klamlivá kybernetická taktika, ktorú útočníci používajú na získanie citlivých informácií od nič netušiacich obetí. Táto metóda zvyčajne zahŕňa posielanie podvodných e-mailov alebo správ, ktoré napodobňujú legitímne zdroje, ako sú banky, vládne agentúry alebo známe spoločnosti. Cieľom je oklamať príjemcov, aby prezradili osobné údaje, ako sú prihlasovacie údaje, čísla kreditných kariet alebo iné citlivé údaje ako napr. číslo sociálneho zabezpečenia.

Phishing predstavuje typický príklad útoku sociálneho inžinierstva (angl. social engineering). Útoky sociálneho inžinierstva sú rôzne techniky, ktoré zneužívajú ľudskú psychológiu na škodlivé účely. Phishing e-mailov je asi najznámejší príklad, zahŕňa útočníkov posielajúcich e-maily, ktoré sa zdajú byť z dôveryhodných zdrojov, aby oklamali príjemcov a prinútili ich poskytnúť citlivé informácie alebo kliknúť na škodlivé odkazy. Tieto e-maily často napodobňujú vzhľad a jazyk legitímnej korešpondencie od bánk, poskytovateľov služieb alebo dokonca priateľov a kolegov, čo ich robí ťažko rozlíšiteľnými od skutočných komunikácií.

Phishing patrí dlhodobo medzi najčastejšie a najúspešnejšie kybernetické útoky. Podľa pravidelných bezpečnostných reportov (napr. Verizon DBIR, ENISA Threat Landscape) je phishing každoročne jedným z hlavných počiatočných vektorov kompromitácie a stojí za väčšinou incidentov, pri ktorých útočníci získajú prvotný prístup k účtom, systémom alebo interným sieťam. Úspešnosť phishingových kampaní je vysoká najmä preto, že útočníci cielia na ľudský faktor, ktorý je najslabším článkom bezpečnostného reťazca.

Phishing sa rozširuje aj na textové správy, známe ako SMS phishing alebo ***smishing***. Pri týchto útokoch páchateľ posiela textové správy, ktoré lákajú príjemcu, aby klikol na škodlivý odkaz alebo poskytol osobné údaje. Tieto správy môžu pôsobiť ako správy od dôveryhodnej organizácie a často vytvárajú pocit naliehavosti na okamžitú reakciu na prijatú správu. ***Vishing***, alebo hlasový phishing (angl. voice phishing), je ďalší variant, kde útočníci využívajú telefonáty na získanie citlivých informácií. V týchto situáciách sa útočník často vydáva za legitímneho zástupcu organizácie, aby získal informácie alebo priamu akciu od obete útoku.

Moderná forma hlasového vishingu je ***deepfake vishing***, pri ktorej útočník využíva syntézu hlasu založenú na umelej inteligencii, aby napodobnil hlas konkrétnej osoby – napríklad manažéra, kolegu alebo zákazníka. Vďaka pokročilým modelom hlasového klonovania stačí útočníkovi len niekoľko sekundová nahrávka, aby vytvoril presvedčivý falošný hlas. Takýto hovor môže obeť presvedčiť, aby vykonala vysokorizikové akcie, ako sú prevody peňazí, zmena hesiel alebo poskytnutie citlivých údajov. Deepfake vishing je mimoriadne nebezpečný, pretože navonok pôsobí ako legitímny telefonát a obchádza tradičné technické detekčné mechanizmy.

Vydávanie sa za niekoho a pretextovanie sú sofistikované taktiky používané pri útokoch sociálneho inžinierstva, pri ktorých útočník predstiera určitú autoritu alebo vytvára presvedčivý scenár na zneužitie dôvery obete útoku. Prijatím identity autoritatívnej osoby ako je policajt, IT technik alebo firemný manažér, môžu útočníci manipulovať cieľmi útokov, aby prezradili dôverné informácie alebo vykonali kroky ohrozujúce bezpečnosť. Vytváranie presvedčivých scenárov si vyžaduje podrobné poznanie obete alebo jej organizácie, čo robí tieto útoky obzvlášť zákernými a účinnými.

Baiting (angl. baiting = vábiť, lákať) a ***tailgating*** (angl. tailgate = zadné vrátka) sú ďalšie rozšírené metódy sociálneho inžinierstva. *Baiting* útoky zahŕňajú lákanie obetí útoku sľubom tovaru alebo služieb, aby ukradli ich osobné údaje alebo infikovali ich systémy škodlivým softvérom. Môže to byť vo forme stiahnutia bezplatného softvéru alebo prístupu k exkluzívnemu obsahu. *Tailgating* je naopak taktika fyzického narušenia bezpečnosti, pri ktorej

neoprávnená osoba získa prístup do zakázanej oblasti tým, že sleduje niekoho oprávneného. Tento útok využíva spoločenskú normu zdvorilosti a prirodzenú tendenciu pomáhať druhým, čo útočníkom umožňuje preniknúť do zabezpečených prostredí.

Vyvolávanie a manipulácia sú jemné, no silné techniky sociálneho inžinierstva. Tieto metódy zahŕňajú získavanie informácií prostredníctvom zdanlivo nevinného rozhovoru, kde útočník zapája cieľ obeť do diskusie a nenápadne ho usmerňuje k získaniu požadovaných informácií. Útočníci často emocionálne manipulujú cieľmi, využívajúc ľudské sklony, ako je túžba byť obľúbený alebo strach z konfrontácie. Získanie dôvery prostredníctvom vzťahu a empatie je v týchto interakciách kľúčové, pretože znižuje obranu cieľa útoku a robí ho náchylnejším k prezradeniu citlivých informácií.

Každý útok sociálneho inžinierstva ukazuje zložitosť a rozmanitosť taktík, ktoré útočníci používajú na zneužitie ľudskej psychológie. Zdôrazňujú potrebu neustálej ostražitosti v osobnom aj pracovnom prostredí, aby sa zabránilo týmto manipulatívnym technikám. Pochopenie povahy týchto útokov je prvým krokom k rozvoju účinných stratégií na ich zvládnutie, pričom sa zdôrazňuje dôležitosť povedomia a vzdelávania v prebiehajúcom boji proti kybernetickým hrozbám.

Na boj proti phishingu musia byť jednotlivci a organizácie ostražití a skeptickí voči nevyžiadaným komunikáciám. Kľúčové preventívne opatrenia zahŕňajú neklikanie na odkazy ani sťahovanie príloh z neznámych zdrojov, overovanie pravosti žiadostí o osobné údaje a používanie anti-phishingových nástrojov dostupných vo väčšine moderných webových prehliadačov a e-mailových služieb. Vzdelávanie a školenia v identifikácii phishingových pokusov sú kľúčové pre vybudovanie prvej línie obrany proti týmto kybernetickým útokom.

3.1 Experimenty a výsledky

Cieľom pilotného testovania bolo overiť a porovnať vhodnosť metód K-Means a HDBSCAN na identifikáciu phishingu v dátach vo forme logov. Testovanie malo zistiť, či tieto metódy dokážu:

- identifikovať opakujúce sa vzorce a husté oblasti logov
- odhaliť anomálie a outlier-y, ktoré môžu indikovať phishingové alebo podozrivé udalosti
- porovnať rozdiely v prístupe oboch algoritmov, najmä schopnosť detekcie outlier-ov a štruktúru klastrov
- poskytnúť zrozumiteľnú interpretáciu klastrov.

Pre pilotné testovanie sme pracovali so súborom 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. Detailný popis experimentov možno nájsť v podkladovom dokumente Analýza ML metód a algoritmov pre detekciu phishing.docx. Na základe vykonaných experimentov možno vyvodiť nasledovný záver.

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

Algoritmus *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 mimo-ležiace 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.

3.2 Popis vybraného algoritmu

HDBSCAN je hierarchický klastrovací algoritmus, ktorý je založený na dobre známom a v praxi overenom algoritme DBSCAN. DBSCAN je algoritmus klastrovania založený na hustote, ktorý nevyžaduje vopred určený počet klastrov. DBSCAN závisí od dvoch parametrov: minimálny počet bodov potrebných na vytvorenie zhluku zo skupiny bodov a ε (ktorý určuje minimálnu vzdialenosť medzi dvoma bodmi, aby sa mohli nazývať susedné body).

Interne algoritmus klasifikuje každý dátový bod ako jeden z nasledujúcich troch kategórií:

- stredové body - sú tie, ktoré majú aspoň minimálny počet bodov v okolí definovaný okruhom polomeru ε
- hraničné body – sú také body, ktoré nie sú stredovými bodmi, ale nachádzajú sa v susednej oblasti alebo klastra stredového bodu opísaného vyššie
- body anomálie – sú také body, ktoré nie sú ani stredovým bodom, ani z neho nie sú dosiahnuteľné (teda ani hraničný bod)

Stručne zhrnuté, proces klastrovania funguje nasledovne:

1. Nastavenie hodnoty parametrov, minimálny počet bodov a ε.
2. Náhodné vybratie počiatočného bodu (povedzme A) a nájdenie všetkých bodov vo vzdialenosti ε alebo menšej od tohto bodu.
3. Ak počet bodov spĺňa prahovú hodnotu pre minimálny počet bodov, potom je A stredovým bodom a okolo neho sa môže vytvoriť zhluk alebo klaster. Všetky body vo vzdialenosti ε alebo menšej sú hraničné body a pridajú sa do zhluku so stredom v bode A.

4. Iteratívne opakovanie týchto krokov pre každý bod. Ak sa ukáže, že bod (povedzme B) pridaný do zhluku je tiež stredovým bodom, najprv vytvoríme jeho vlastný zhluk a potom ho zlúčime s pôvodným zhlukom okolo A.

DBSCAN je teda schopný identifikovať oblasti s vysokou hustotou v príznakovom priestore a zoskupiť body do klastrov. Bod, ktorý nespadá do zhluku, je určený ako anomálny alebo odľahlý.

Algoritmus DBSCAN očakáva 2 vstupy definované používateľom: prahovú hodnotu pre hustotu ε a m_{pts}, ktorá sa správa ako vyhladzovací faktor pre odhad hustoty jednotlivých objektov. Pôvodný algoritmus DBSCAN zaviedol pojem tzv. *jadrových objektov*, čo sú objekty s aspoň m_{pts} susedmi v rámci definovanej vzdialenosti ε. Títo susedia vytvárajú susedstvo N_{ϵ} objektu ε. Klaster C je definovaný ako neprázdna maximálna podmnožina vstupnej dátovej množiny taká, že každý pár objektov v C je spojený na základe hustoty. 2 jadrové objekty x_p a x_q sú spojené na základe hustoty, pokiaľ sú priamo alebo tranzitívne dosiahnuteľné v rozsahu prahovej hodnoty ε, pričom priama dosiahnuteľnosť znamená, že $x_p \in N_{\varepsilon}(x_q)$ a $x_q \in N_{\varepsilon}(x_p)$. V prípade algoritmu DBSCAN je globálna prahová hodnota hustoty ťažko nastaviteľný, ak existujú zhluky rôznej hustoty, pretože každý klaster musí prekročiť stanovenú úroveň hustoty. Algoritmus HDBSCAN tento problém rieši vytvorením hierarchie zhlukov s rôznou hustotou. Aby sa obmedzila hĺbka hierarchie, bol zavedený voliteľný parameter m_{cl} ako minimálny počet objektov v rámci klastra. V literatúre sa odporúča nastaviť $m_{cl} = m_{pts}$ a ponechať m_{pts} ako jediný používateľom definovaný parameter.

Hierarchia HDBSCAN sa vytvára nasledujúcimi krokmi:

1. Vytvorenie minimálneho stromu rozpätia MST (Minimum Spanning Tree) grafu vzájomnej dosiahnuteľnosti, ktorý pozostáva zo všetkých objektov ako vrcholov a vzájomných dosiahnuteľných vzdialeností pre každý pár objektov ako váhy hrán. Vzájomná dosiahnuteľnosť je definovaná ako:

   $d_{mreach}(x_p, x_q) = max\{d_{core}(x_p), d_{core}(x_q), d(x_p, x_q)\}$,

   kde $d_{core}(x)$ je vzdialenosť objektu x od jeho najbližších susedov m_{pts} a $d(x_p, x_q)$ je vzdialenosť medzi objektami x_p a x_q.

2. Celá dátová sada je obsiahnutá v klastri, ktorý tvorí koreň hierarchie. Následné zhluky a náhodné alebo šumové prvky vznikajú odstraňovaním hrán zo stromu MST v klesajúcom poradí jednotlivých váh. Odstránenie hrany vedie buď k dvom novým zhlukom s viacerými objektmi ako m_{cl} v každom klastri, zmenšenému klastru s minimálne m_{cl} objektmi a náhodnými elementami, alebo len šumovými prvkami vtedy, ak klaster zmizne.
3. Z celej hierarchie zhlukovania obsahujúcej všetky rozdelenia sa vytvorí kompaktná hierarchia obsahujúca iba najdôležitejšie úrovne, konkrétne keď sa zhluky objavia prvýkrát alebo úplne zmiznú.

V špeciálnom prípade nastavenie parametra $m_{pts} = 1$ vedie k hierarchii jedného prepojenia.

Vďaka rôznym prahom hustoty reprezentovaným hierarchiou (čím bližšie ku koreňu stromu, tým menej hustá je prahová hodnota hustoty), používateľ už nemusí zadávať parameter ε. Pôvodný algoritmus DBSCAN vytvára ploché partície, ktorých klastre alebo zhluky sú maximálne množiny prepojených bodov s hustotou bodov v ich okolí nad danou prahovou hodnotou hustoty, nastavenou parametrom ε. Výsledok zodpovedá extrakcii klastrov z hierarchie HDBSCAN na úrovni hustoty ε, čo predstavuje horizontálny rez hierarchiou.

Aby sa z takejto hierarchie extrahovali zhluky s variabilnými hustotami (rôznych úrovní), bola

zavedená miera stability S (Stability). Stabilita S klastra meria, ako stabilný v hierarchii daný zhluk je. To závisí od rozsahu hustoty, v ktorom klaster existuje, t. j. od minimálnej hustoty, pri ktorej klaster začína existovať, a maximálnej hustoty, pri ktorej sa klaster buď rozdelí alebo zmizne, ako aj od počtu objektov, ktoré zhluk obsahuje v porovnaní s jeho potomkami a rodičovským klastrom. Hodnoty stability je možné vypočítať pri vytváraní hierarchie.

Algoritmus na extrakciu plochej partície z takejto hierarchie je iteratívny postup a začína na úrovni listu ako aktuálneho zhlukovacieho riešenia. Pre každý pár rodič-potomok v aktuálnom riešení: Ak je rodič C_p stabilnejší ako jeho potomkovia, povedzme C_{d1} a C_{d2}, t.j. $S_{Cp} > S_{Cd1} + S_{Cd2}$, C_{d1} a C_{d2} sú v aktuálnom riešení nahradené C_{d1} a C_{d2}. Inak je rodič C_p priradený k C_{d1} a C_{d2} ako nový rodičovský klaster, pričom C_p sa odteraz prakticky ignoruje. Tento postup sa opakuje, kým koreň hierarchie nie je rodičom všetkých klastrov v aktuálnom riešení.

Extrahovanie zhlukovacieho riešenia týmto spôsobom je efektívne, pretože v každej fáze sa prijímajú lokálne rozhodnutia, t. j. porovnanie iba rodičov a potomkov (a žiadnych iných klastrov) stačí na získanie globálne optimálneho riešenia týkajúceho sa S, čím sa vráti najstabilnejšia množina klastrov alebo zhlukov.

Viac informácií k popisu spomínaných algoritmov pre zhlukovanie alebo klastrovanie možno nájsť v podkladovom dokumente Clustering.pdf.

3.3 Záver

Táto kapitola sa zameriavala na problematiku phishingových útokov a príbuzných techník sociálneho inžinierstva, ktoré dlhodobo patria medzi najčastejšie a najúspešnejšie vektory kybernetických útokov. Vstupné dáta pre použité algoritmy strojového učenia sú tvorené bezpečnostnými logmi získanými zo SIEM nástrojov, ktoré agregujú a normalizujú udalosti z mailových serverov, bezpečnostných brán a sieťových senzorov. Tieto logy obsahujú štruktúrované príznaky opisujúce technické, behaviorálne a reputačné charakteristiky e-mailovej komunikácie, na základe ktorých je možné identifikovať anomálie a podozrivé vzory správania bez potreby explicitného označenia dát. Analýza ukázala, že phishing, vrátane jeho moderných foriem ako smishing, vishing a deepfake vishing, predstavuje významnú bezpečnostnú hrozbu najmä vďaka zneužívaniu ľudského faktora a schopnosti obchádzať tradičné technické ochranné mechanizmy.

V praktickej časti kapitoly boli overované možnosti využitia metód strojového učenia na identifikáciu phishingu v dátach vo forme bezpečnostných logov. Pilotné testovanie porovnávalo klastrovacie algoritmy K-Means a HDBSCAN s cieľom posúdiť ich schopnosť identifikovať opakujúce sa vzorce správania, odhaliť anomálie a poskytnúť interpretovateľné výsledky vhodné pre analytické využitie v prostredí SOC. Výsledky experimentov potvrdili, že zatiaľ čo algoritmus K-Means umožňuje rýchly a deterministický rozklad dát, jeho použitie je limitované predpokladom rovnomernej hustoty klastrov a neschopnosťou explicitne identifikovať odľahlé hodnoty.

Naopak, algoritmus HDBSCAN sa v rámci pilotného testovania ukázal ako vhodnejší nástroj pre analýzu heterogénnych bezpečnostných logov. Jeho schopnosť pracovať s klastrami rôznej hustoty, automaticky určovať ich počet a identifikovať outliery predstavuje významnú výhodu pri detekcii phishingových a iných neštandardných udalostí. Tieto vlastnosti umožňujú efektívnejšie zachytiť jemné rozdiely v správaní, ktoré môžu indikovať podozrivé alebo škodlivé aktivity, a zároveň redukovať šum v dátach.

Na základe uvedených zistení možno konštatovať, že metódy klastrovania založené na hustote, reprezentované algoritmom HDBSCAN, predstavujú perspektívny prístup k analýze logov v kontexte detekcie phishingu a spoofingu, najmä v prípadoch, kde nie sú k dispozícii presne označené tréningové dáta. Táto kapitola tak vytvára metodický základ pre ďalšie časti práce,

ktoré sa zameriavajú na aplikáciu strojového učenia pri detekcii ďalších kategórií bezpečnostných hrozieb v prostredí SOC.

4 SPAM/HAM KLASIFIKÁCIA

Väčšina bezpečnostných hrozieb používa e-mail ako vektor útoku. Keďže objem takto prenášanej prevádzky je obzvlášť veľký, je potrebné použiť automatizované postupy detekcie, ktoré využívajú algoritmy strojového učenia (ML) a umelej inteligencie (AI). Aby sme plne pochopili silu algoritmov ML/AI aplikovaných na detekciu spamu, je potrebné si najprv ujasniť predstavy, na akých úlohách by sa malo vykonávať filtrovanie mailových správ typu spam.

Pre lepšie pochopenie úloh, vykonávaných spamovým filtrom, možno uviesť nasledovný príklad. Predstavte si, že oddeľujete e-maily, ktoré dostávame, a kategorizujete ich na základe prítomnosti alebo neprítomnosti konkrétnych kľúčových slov, ktoré sa vyskytujú v texte e-mailov s určitou frekvenciou. Na tento účel možno mohli uviesť všetky správy, ktoré dostávame do doručenej pošty, v tabuľke. Ako však postupovať pri klasifikácii dostávaných správ do kategórií HAM (legitímna správa) alebo SPAM?

Ako už bolo spomenuté, bude sa hľadať počet výskytov podozrivých kľúčových slov v texte e-mailových správ. Jednotlivým správam identifikovaným ako spam sa potom pridelí skóre na základe počtu výskytov identifikovaných kľúčových slov. Toto skóre ďalej poskytne referenciu pre klasifikáciu následných e-mailových správ.

Určí sa prahová hodnotu, vďaka ktorej bude možné oddeliť emailové správy typu spam. Ak vypočítané skóre prekročí stanovenú prahovú hodnotu, e-mail bude automaticky klasifikovaný ako spam - v opačnom prípade bude správa prijatá ako legitímna a teda klasifikovaná ako ham. Táto prahová hodnota (ako aj pridelené skóre) sa bude neustále určovať, aby sa zohľadnila nová séria spamových správ, s ktorými sa v budúcnosti možno stretnúť.

Už z abstraktného popisu algoritmu detekcie spamu si možno všimnúť niektoré dôležité vlastnosti, ktoré je potrebné mať na pamäti: je nutné pristúpiť k identifikácii určitého počtu podozrivých kľúčových slov, ktoré umožňujú klasifikovať správy ako potenciálne spamové e-maily, pričom každému e-mailu sa priradí skóre na základe počtu výskytov identifikovaných kľúčových slov.

Je potrebné nastaviť určitú prahovú hodnotu pre skóre priradené jednotlivým e-mailom, nad ktorou budú e-maily automaticky klasifikované ako spam. Takisto je dôležité správne zvážiť význam kľúčových slov prítomných v texte e-mailov, aby bolo možné adekvátne reprezentovať mieru pravdepodobnosti, že správa, ktorá obsahuje dané kľúčové slová, predstavuje spam (kľúčové slová by v podstate mohli byť dokonca neškodné, ale spolu s väčšou pravdepodobnosťou predstavujú nevyžiadanú poštu).

Je nutné vziať do úvahy, že odosielatelia nevyžiadanej pošty si dobre uvedomujú snahu filtrovať nechcené správy, a preto sa budú snažiť čo najlepšie prijať nové stratégie, aby spamové filtre oklamali. To sa premieta do procesu nepretržitého a iteratívneho učenia, ktorý sa dá dobre implementovať pomocou algoritmov ML/AI.

Z uvedeného je jasné, že nie je náhoda, že detekcia spamu predstavuje prvý test pri zavádzaní ML/AI v oblasti kybernetickej bezpečnosti. Prvé riešenia na detekciu spamu v praxi využívali statické pravidlá, ktoré používali regulárne výrazy na identifikáciu preddefinovaných vzorov podozrivých slov v texte e-mailu.

Tieto statické pravidlá sa rýchlo ukázali ako neúčinné v dôsledku stále nových stratégií klamania implementovaných spamermi na oklamanie antispamových filtrov. Preto bolo potrebné prijať dynamický prístup, ktorý by umožnil spamovému filtru učiť sa na základe neustálych inovácií zavádzaných spamermi a zároveň využívať rozhodnutia používateľa pri

klasifikácii jeho e-mailov. Týmto spôsobom bolo možné efektívne zvládnuť explozívne šírenie fenoménu spamu.

4.1 Experimenty a výsledky

Experimenty prebehli na 2 najčastejšie používaných datasetoch v oblasti automatickej detekcie emailových správ typu SPAM – „SpamAssassin“ a „SMS Spam“, ktoré predstavujú „zlatý štandard“ v tejto oblasti. Bližšie podrobnosti k týmto databázam ako aj k samotným experimentom sú uvedené v podkladovom dokumente Analýza ML metód a algoritmov pre detekciu spam emailových správ.docx. Ako zhrnutie možno uviesť nasledovné. 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**.

4.2 Popis vybraného algoritmu

Algoritmus **XGBoost** (Extreme Gradient Boosting) je jednou z najefektívnejších implementácií rozhodovacích stromov s podporou gradientu a bol vybraný ako jeden z najlepších algoritmov strojového učenia používaných v súťažiach Kaggle. Algoritmus XGBoost, špeciálne navrhnutý na optimalizáciu využitia pamäte a využitie hardvérového výpočtového výkonu, skracuje čas vykonávania pri súčasnej zvýšenej efektívnosti v porovnaní s mnohými algoritmami strojového učenia a dokonca aj modelmi hlbokého učenia. Hlavnou myšlienkou boostingu je postupne budovať pod-stromy z pôvodného stromu tak, aby každý nasledujúci strom znížil chyby (reziduá) stromu predchádzajúceho. Týmto spôsobom budú nové pod-stromy aktualizovať predchádzajúce reziduá, aby sa znížila chybovosť účelovej funkcie (cost function).

Zoberme si súbor dát vyjadrený takto:

$$D = \{(x_i, y_i) \text{ kde } x_i \in R_n;\ y_i \in R)\},$$

kde n predstavuje dimenziu príznakov x_i a y_i je zodpovedajúca odpoveď alebo trieda. Pre i-ty prvok množiny dát definujeme predikovanú hodnotu $\hat{y}_i$ nasledovne:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), \quad f_k \in F,$$

kde f_k označuje nezávislý strom v priestore regresných stromov F a $f_k(x_i)$ vyjadruje predikované skóre pre i-tu vzorku a k-ty strom. Objektívna funkcia algoritmu XGBoost, označovaná ako L, je daná nasledovne:

$$\mathcal{L} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i\right) + \sum_{k=1}^{K} \Omega\left(f_k\right).$$

Minimalizáciou objektívnej funkcie L je možné natrénovať funkcie f_k regresného stromu. Chybová funkcia $l(y_i, \hat{y}_i)$ vyhodnocuje rozdiel medzi predikciou $\hat{y}_i$ a skutočnou hodnotou y_i. Člen Ω sa používa na zabránenie problému s pretrénovaním penalizáciou zložitosti modelu nasledovne:

$$\Omega\left(f_k\right) = \gamma T + \frac{1}{2}\lambda \left\|w\right\|^2 ,$$

kde T je počet listov v strome, w predstavuje skóre (váhu) listu (napr. predpokladaná hodnota pre vzorky, ktoré skončia v tomto liste), γ je regularizačný faktor, ktorý vyjadruje náklady na zložitosť, ktoré odrádzajú od vytvárania stromu s príliš veľkým počtom listov a λ je regularizačný výraz L2 na váhach listov, ktorý sa ich snaží zmenšovať tak, aby bol model konzervatívnejší. Priama minimalizácia objektívnej funkcie je zložitá. XGBoost používa Taylorovo rozšírenie druhého rádu na jeho aproximáciu. Toto je kľúčový dôvod jeho výkonu, pretože používa prvú aj druhú deriváciu (gradient a hessián). Nech $I_j = \{i|q(x_i) = j\}$ je inštancia množiny listov j s $q(x)$ fixnou štruktúrou. Optimálne váhy w_j listu j a zodpovedajúcu optimálnu hodnotu možno získať nasledujúcimi rovnicami:

$$w_j^* = -\frac{g_j}{h_j + \lambda},$$

$$\mathcal{L}^* = -\frac{1}{2}\sum_{j=1}^{T} \frac{\left(\sum_{i \in I_j} g_i\right)^2}{\sum_{i \in I_j} h_i + \lambda} + \lambda T ,$$

kde g_i a h_i sú prvá a druhá derivácia stratovej funkcie L. Stratovú funkciu L možno použiť ako skóre kvality stromovej štruktúry q. Čím je skóre menšie, tým je model lepší. Keďže nie je možné vyčísliť všetky stromové štruktúry, *greedy* algoritmus môže problém vyriešiť tak, že začne od jedného listu a iteratívne pridáva vetvy do stromu. Nech I_R a I_L sú množiny inštancií pravého a ľavého uzla po rozdelení. Za predpokladu $I = I_R \cup I_L$, potom sa po rozdelení redukcia straty vyjadrí nasledovne:

$$\mathcal{L}_{\text{split}} = \frac{1}{2}\left[\frac{\left(\sum_{i \in I_L} g_i\right)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{\left(\sum_{i \in I_R} g_i\right)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{\left(\sum_{i \in I} g_i\right)^2}{\sum_{i \in I} h_i + \lambda}\right] - \gamma.$$

Tento vzťah sa zvyčajne používa v praxi na hodnotenie kandidátov na rozdelenie. Model XGBoost používa veľa jednoduchých stromov a skóre listových uzlov počas rozdelenia. Prvé tri členy rovnice predstavujú skóre ľavého, pravého a pôvodného listu. Okrem toho člen γ predstavuje regularizáciu na dodatočnom liste a bude sa používať v procese trénovania.

4.3 Záver

Táto kapitola sa zameriavala na problematiku automatickej klasifikácie e-mailových správ typu spam a ham, ktorá predstavuje jednu z najdlhšie skúmaných a prakticky nasadzovaných oblastí využitia strojového učenia v kybernetickej bezpečnosti. Analýza poukázala na historický vývoj spamových filtrov od jednoduchých pravidlových a heuristických

prístupov až po moderné metódy strojového učenia, ktoré dokážu adaptívne reagovať na meniace sa stratégie odosielateľov nevyžiadanej pošty.

Experimentálna časť kapitoly bola zameraná na porovnanie vybraných klasifikačných algoritmov na referenčných datasetoch SpamAssassin a SMS Spam, ktoré predstavujú etablovaný štandard v oblasti hodnotenia účinnosti spamových filtrov. Dosiahnuté výsledky potvrdili, že algoritmy založené na súborovom učení, predovšetkým XGBoost, poskytujú veľmi vysokú presnosť klasifikácie pri zachovaní priaznivých časových nárokov na vyhodnocovanie. Vysoké hodnoty metrík Precision a Recall naznačujú, že zvolený prístup je vhodný nielen na detekciu nevyžiadanej pošty, ale aj na minimalizáciu falošne pozitívnych a falošne negatívnych klasifikácií, ktoré majú zásadný vplyv na použiteľnosť systému v reálnej prevádzke.

Porovnanie výsledkov na rôznych datasetoch zároveň ukázalo, že účinnosť klasifikačných modelov je výrazne ovplyvnená charakterom vstupných dát, ich štruktúrou a kvalitou. Pokles presnosti pri spracovaní datasetu SMS Spam poukazuje na potrebu prispôsobenia modelov konkrétnemu typu komunikácie a zdôrazňuje význam správnej voľby príznakov a tréningových dát. Z tohto dôvodu je nevyhnutné, aby klasifikačné modely v produkčnom prostredí prechádzali priebežnou aktualizáciou a adaptáciou na nové formy spamových kampaní.

Na základe vykonaných experimentov možno konštatovať, že algoritmus XGBoost ***predstavuje vhodné riešenie*** pre klasifikáciu spam/ham v kontexte bezpečnostných systémov, najmä vďaka svojej robustnosti, vysokej presnosti a schopnosti efektívne pracovať s rozsiahlymi a heterogénnymi dátami. Táto kapitola tak poskytuje pevný metodický základ pre využitie klasifikačných algoritmov strojového učenia pri ochrane e-mailovej komunikácie a vytvára východisko pre ďalšie kapitoly, ktoré sa venujú detekcii komplexnejších a sofistikovanejších bezpečnostných hrozieb.

5 DETEKCIA POKROČILÝCH APT TECHNÍK

Pokročilá perzistentná hrozba APT (Advanced Persistent Threat) je forma kybernetickej hrozby, ktorú predstavujú dobre financované organizácie, často s cieľom získať kľúčové informácie od cieľovej organizácie. APT hrozba sa líši od konvenčných a viac-stupňových útokov a je definovaná kombináciou troch slovíčok:

- Pokročilá – Advanced – Pokročilý aktér hrozieb je technicky zdatný, skúsený a schopný objavovať alebo zavádzať nové zraniteľnosti do systému a vyvíjať nové spôsoby, ako tieto zraniteľnosti zneužiť. Útočníci dokonca siahajú po rôznych útokoch ako napr. zero-day exploit, planted insiders, alebo supply chain exploitation, keď to obrana cieľa vyžaduje.
- Pretrvávajúca – Persistent – Pretrvávajúca hrozba je ktokoľvek, kto prejavuje odhodlanie útočiť až do dosiahnutia cieľa, je zručný a skúsený v prispôsobovaní sa nepriateľskému územiu, aby zostal neodhalený, a má zdroje na vývoj svojich TTP na porazenie obranných strategických alebo taktických protiopatrení. Tieto zdroje zahŕňajú rôzne aktíva a nie sú špecificky obmedzené len na finančné prostriedky. Ľudský kapitál je napríklad veľmi relevantným zdrojom. Pretrvávajúca hrozba bude mať prístup k personálnym kapacitám pre každú operáciu, ktorú iniciuje. Pracujú prevažne v tímoch, keďže rozsah a škála ich činností sú dostatočne rozsiahle. Ďalšími aktívami sú infraštruktúra, čas a finančné zdroje.
- Hrozba – Threat – Hrozba označuje jednotlivcov (napr. osoby, skupiny a organizácie), ktorí majú motiváciu, schopnosti a príležitosť zapojiť sa do škodlivých činov. Kľúčovou definujúcou črtou hrozby je jej zlomyseľný úmysel, ktorý je hnacou silou jej motivácie.

Pri uvažovaní o APT je dôležité si uvedomiť, že sa jedná o ľudí, keďže tento pojem bol špeciálne navrhnutý tak, aby ich odlíšil od jednoduchého malvéru. Ako tvrdí Richard Bejtlich: "Hrozba znamená, že protivník nie je kus bezduchého kódu." Technicky neexistuje spôsob, ako zachytiť ľudskú kreativitu alebo predstavivosť. Preto nemôžeme predpokladať, že vieme predpovedať, aký útok APT vykoná. Výskum často považuje APT útoky za cielené viacstupňové útoky. Avšak obmedzenie, že viacstupňový útok musí pozostávať aspoň z dvoch atomických alebo drobných udalostí, nie je pre APT útok povinné. Ak útočník potrebuje na dosiahnutie cieľa len jednu akciu, použije ju. Cesta najmenšieho odporu, ako časť pojmu Pokročilá v definícii, opisuje ich schopnosť využívať celé spektrum narušení. Napríklad, ak existuje známa zraniteľnosť s príslušným zneužívaným materiálom (exploit material), APT ju môže a bude využívať na dosiahnutie svojich cieľov.

Viacstupňový útok môže tiež zasiahnuť jeden konkrétny cieľ. Preto majú detekčné systémy problém rozlíšiť hrozby stojace za útokom.

APT sa snažia zostať neodhalené a preto používajú techniky skrývania (concealment), pričom jednou z najčastejšie používaných je technika living-off-the-land. Tieto techniky zahŕňajú maskovanie škodlivej činnosti za zdanlivo neškodné programy, ktoré využívajú nástroje už na počítačoch používateľov. Tieto nástroje sú často súčasťou operačného systému alebo iného softvéru inštalovaného používateľom. Tento typ útoku je obzvlášť náročné odhaliť, pretože útočníci zvyčajne nevytvárajú nové škodlivé súbory na cieľových počítačoch a bežné antivírusové skeny ich nemusia identifikovať.

Tieto techniky sa v konvenčnom spravodajstve kybernetických hrozieb CTI (Cyber Threat Iintelligence) nezohľadňujú, pretože vyžadujú vyššiu úroveň sofistikovanosti od útočníka a nemožno ich porovnávať so známymi postupmi. Tento fakt zdôrazňuje dôležitosť

zohľadnenia týchto pokročilých taktík v bezpečnostných opatreniach a neustáleho aktualizovania CTI, aby sme boli o krok vpred s meniacim sa prostredím hrozieb.

Napriek tomu APT môže a bude používať známe vzory útokov, ako sú opakované útoky (replay attacks) alebo vloženie kódu (injection), prostredníctvom známych zraniteľností. Zavedené systémy založené na pravidlách a dátami riadené prístupy pre konkrétne TTP sú čiastočným riešením.

V najhoršom prejave možno APT opísať ako komplexného nepriateľa so širokým spektrom útokov charakterizovaného DoD (Department of Defense) a niekedy nazývaného integrované hrozby. Táto vlastnosť znamená, že nevyhľadávajú príležitosti na úder - na umožnenie svojich operácií využívajú politické, vojenské, spravodajské alebo ekonomické zdroje.

Historické príklady útokov APT, ktoré sú obzvlášť zaujímavé, sú útok Stuxnet, ktorý bol použitý na zničenie centrifúg na obohacovanie uránu v Natanzi v Iráne, Industroyer, ktorý spôsobil výpadok elektriny na Ukrajine v roku 2016, a TRITON, ktorý zaútočil na bezpečnostné prístrojové systémy ropného priemyslu. Medzi ďalšími útokmi APT predstavuje významný pilier vo vývoji APT prechod od prevažne špionáže k deštruktívnym útokom.

Okrem kradnutia informácií a manipulácie či dokonca ničenia cieľov môže APT tiež usilovať o inštaláciu backdoor softvéru, aby získal jednoduchý prístup v strategicky výhodných časoch. Okrem využití rôznych taktík, techník a procedúr TTP (Tactics, Techniques, and Procedures) sa typy útokov líšia aj podľa oblastí útočníka, cieľa, účelu a prístupu.

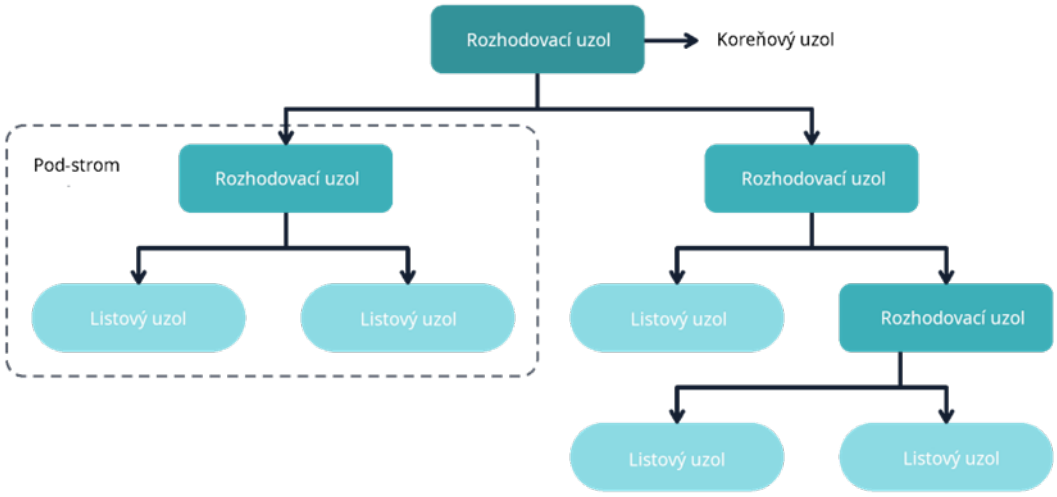
5.1 Experimenty a výsledky

Na základe vykonaných experimentov možno zhodnotiť, že najvhodnejšia metóda pre detekciu pokročilých APT útokov je 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. Bližšie podrobnosti k samotným experimentom sú uvedené v podkladovom dokumente Analýza ML metód a algoritmov pre detekciu APT.docx.

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

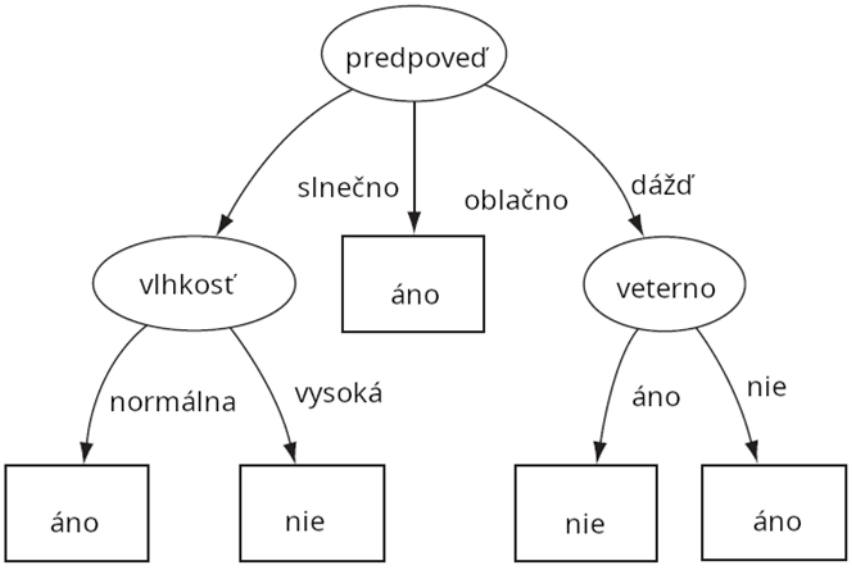
5.2 Popis vybraného algoritmu

Rozhodovací strom je neparametrický algoritmus, ktorý sa používa na klasifikačné aj regresné úlohy. Má hierarchickú stromovú štruktúru, ktorá pozostáva z koreňového uzla, vetiev, vnútorných uzlov a listových uzlov. Ako možno vidieť z nižšie uvedeného diagramu, rozhodovací strom začína koreňovým uzlom, ktorý nemá žiadne prichádzajúce vetvy.



Obr. 20: Rozhodovací strom

Odchádzajúce vetvy z koreňového uzla sa potom privádzajú do interných uzlov, známych aj ako rozhodovacie uzly. Na základe dostupných funkcií oba typy uzlov vykonávajú hodnotenia s cieľom vytvoriť homogénne podmnožiny, ktoré sú označené listovými uzlami alebo koncovými uzlami. Listové uzly predstavujú všetky možné výsledky v rámci súboru údajov. Ako príklad si možno predstaviť, že ste sa snažíme posúdiť, či by sme mali alebo nemali ísť hrať golf, môžeme na výber použiť nasledujúce pravidlá rozhodovania:



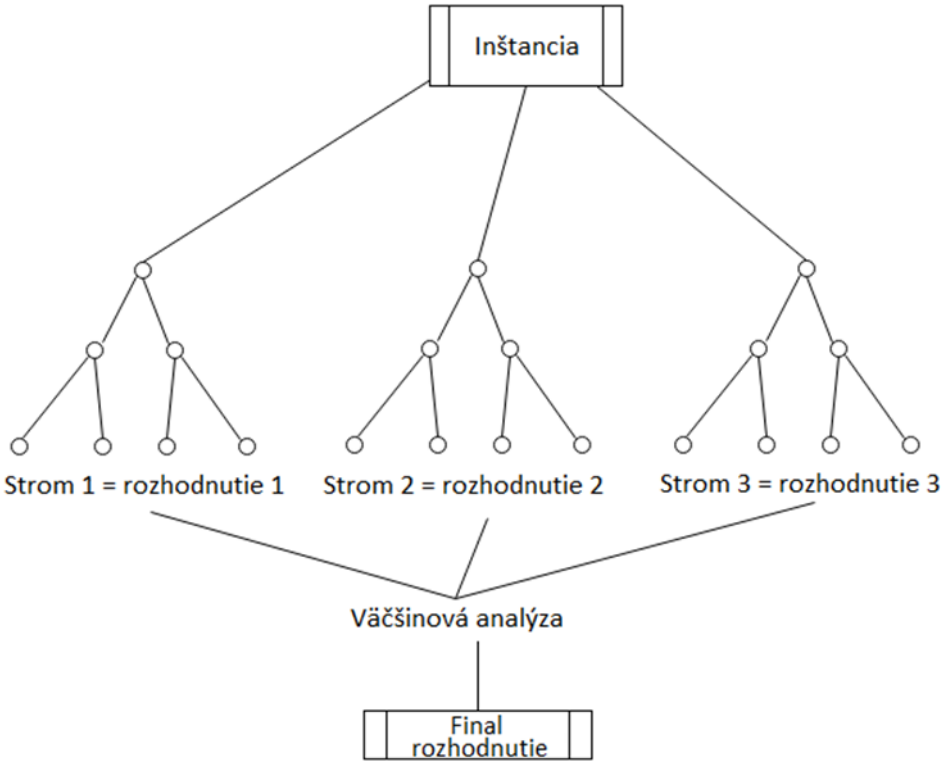
Obr. 21: Rozhodovací strom pre dataset Golf

Tento typ štruktúry vývojového diagramu tiež vytvára ľahko stráviteľnú reprezentáciu rozhodovania, čo umožňuje rôznym skupinám v celej organizácii lepšie pochopiť, prečo bolo rozhodnutie prijaté. Učenie rozhodovacieho stromu využíva stratégiu rozdeľuj a panuj tým, že vykonáva chamtivé vyhľadávanie s cieľom identifikovať optimálne body rozdelenia v strome.

Tento proces rozdelenia sa potom opakuje zhora nadol, rekurzívnym spôsobom, kým všetky alebo väčšina záznamov nie sú klasifikované pod špecifickými štítkami tried. To, či sú

všetky dátové body klasifikované ako homogénne množiny, do značnej miery závisí od zložitosti rozhodovacieho stromu. Menšie stromy sú ľahšie schopné dosiahnuť čisté listové uzly – t. j. dátové body v jednej triede. Ako však strom rastie, je čoraz ťažšie udržať túto čistotu a zvyčajne to vedie k tomu, že do daného pod-stromu spadá príliš málo údajov. Keď k tomu dôjde, je tento jav známy ako fragmentácia údajov a často to môže viesť k nadmernému prispôsobeniu. Výsledkom je, že rozhodovacie stromy uprednostňujú malé stromy, čo je v súlade so zásadou úspornosti podľa princípu „Occamova britva“, čo znamená, že "entity by sa nemali množiť pokiaľ to nie je nevyhnutné". Inak povedané, rozhodovacie stromy by mali zvyšovať svoju zložitosť iba v prípade potreby, pretože najjednoduchšie vysvetlenie je často najlepšie. Aby sa znížila zložitosť a zabránilo sa nadmernému prispôsobeniu, zvyčajne sa používa prerezávanie; Toto je proces, ktorý odstraňuje vetvy, ktoré sa rozdeľujú na prvkoch s nízkou dôležitosťou. Prispôsobenie modelu sa potom môže vyhodnotiť prostredníctvom procesu krížovej validácie.

Random forest (náhodný les) je metóda súborového učenia (ensamble learning), ktorá sa používajú buď na klasifikáciu, alebo na regresné úlohy. Náhodné lesy sa skladajú z viacerých rozhodovacích stromov, ktoré sa kombinujú do jednomyseľného rozhodnutia alebo klasifikácie (obrázok nižšie).



Obr. 22: Random forest – princíp činnosti

Náhodné lesy sú lepšie ako bežné rozhodovacie stromy, pretože nespôsobujú nadmerné prispôsobovanie dát a následné pretrénovanie modelu. Toto sa dosiahnuť vďaka použitiu metód, ktoré využívajú náhodné podpriestory (napríklad *bagging*) na zníženie korelácie medzi jednotlivými stromami. Elementárne algoritmy/stromy sú trénované na rôznych podmnožinách príznakového priestoru, ktoré sú tiež náhodne vyberané. Súbor modelov rozhodovacích stromov pomocou metódy náhodného podpriestoru možno zostaviť pomocou nasledujúceho algoritmu.

Pre N trénovacích objektov a počet príznakov D , postupujeme nasledovne:

1. Zvolí sa parameter L ako počet jednotlivých stromov v súbore (*ensemble*).
2. Pre každý jednotlivý l-tý strom sa vyberie $d_l < D$ ako počet príznakov pre l. Typicky sa pre všetky stromy používa len jedna hodnota.
3. Pre každý strom sa vytvorí tréningovú podmnožinu X_n pomocou bootstrap metódy.

Vytvorenie rozhodovacích stromov z X_n vzoriek sa vykoná nasledovne:

1. Vyberú sa náhodné príznaky d_l zo zdroja, potom optimálne rozdelenie trénovacej množiny obmedzí svoje vyhľadávanie na ne.
2. Podľa daného kritéria sa vyberie najlepší atribút a podľa neho sa urobí rozdelenie v strome.
3. Strom sa vytvára, kým v každom liste nezostane viac ako n_{min} objektov, kým sa nedosiahne určitú výšku stromu, alebo kým nie je vyčerpaná tréningová sada.

Aby bolo možné aplikovať súbor modelu (ensemble) na nový dátový objekt, je potrebné skombinovať výsledky jednotlivých modelov väčšinovým hlasovaním alebo kombinovaním *a posteriori* pravdepodobností. Príklad finálneho klasifikátora je nasledovný:

$$a(x) = \frac{1}{L}\sum_{i=1}^{L} l_i(x)$$

Medzi základné parametre metódy *random forest* patria:

- počet stromov
- počet príznakov pre selekciu rozdelenia stromu
- maximálna hĺbka stromu
- kritérium nečistoty

5.3 Záver

Táto kapitola sa venovala problematike detekcie pokročilých perzistentných hrozieb (APT), ktoré predstavujú jednu z najkomplexnejších a najnáročnejších kategórií kybernetických útokov. APT útoky sa vyznačujú vysokou mierou sofistikovanosti, dlhodobou perzistenciou a schopnosťou adaptácie na obranné mechanizmy cieľovej organizácie, pričom často využívajú kombináciu technických zraniteľností, legitímnych nástrojov a ľudského faktora. Tieto vlastnosti výrazne znižujú účinnosť tradičných detekčných mechanizmov založených výlučne na signatúrach alebo statických pravidlách.

V experimentálnej časti kapitoly boli analyzované možnosti využitia algoritmov strojového učenia na detekciu APT útokov na základe dát zo súboru DAPT2020, ktorý patrí medzi najrozsiahlejšie a najčastejšie používané datasety v tejto oblasti. Porovnanie vybraných klasifikačných metód ukázalo, že algoritmus Random Forest dosahuje najlepšie výsledky z hľadiska presnosti detekcie, pričom dosiahnuté hodnoty metrík Precision a Recall potvrdzujú jeho schopnosť spoľahlivo identifikovať APT aktivity pri súčasnom zachovaní nízkej miery falošných hlásení. Významným faktorom je aj jeho výpočtová efektívnosť, ktorá umožňuje jeho použitie v prostredí blízkom reálnemu času.

Porovnanie s ďalšími algoritmami, ako sú SVM a logistická regresia, poukázalo na kompromis medzi presnosťou a časovou náročnosťou spracovania. Zatiaľ čo SVM dosahuje relatívne vysokú úspešnosť detekcie, jeho výrazne vyššie časové nároky obmedzujú praktickú použiteľnosť v operatívnom prostredí SOC. Naopak, Random Forest poskytuje vyvážený pomer medzi kvalitou detekcie, robustnosťou voči pretrénovaniu a výpočtovou náročnosťou, čo z neho robí vhodného kandidáta pre nasadenie v produkčných bezpečnostných systémoch.

Na základe dosiahnutých výsledkov možno konštatovať, že metódy reprezentované algoritmom Random Forest, predstavujú efektívny nástroj na podporu detekcie pokročilých APT techník, najmä v prostrediach s veľkým objemom heterogénnych bezpečnostných dát. Táto kapitola zároveň poukazuje na potrebu kombinácie dátovo riadených prístupov so znalosťami o taktikách, technikách a procedúrach útočníkov, keďže detekcia APT útokov si vyžaduje viacvrstvový prístup presahujúci možnosti jednotlivých detekčných mechanizmov.

6 KLASIFIKÁCIA A IDENTIFIKÁCIA MALWARE VZOROV

Malvér, známy aj ako škodlivý softvér, je dodnes jednou z hlavných hrozieb týkajúcich sa kybernetickej bezpečnosti. Útočníci často inštalujú škodlivý softvér do systému obete tým, že oklamú používateľov, aby vykonali určité akcie, napríklad kliknutím na nejakú URL adresu, z ktorej sa malware sťahuje, otvorením príloh e-mailu alebo stiahnutím niektorých súborov. Aby si malvér získal oporu v systéme obete, môže tiež využiť zraniteľnosti obete, ako sú chyby, otvorené porty a pluginy. Po stiahnutí sa malvér často rozšíri do iných systémov/sietí.

Na ochranu pred malvérom (angl. malware) je dôležitou bezpečnostnou funkciou určiť, či je vzorka softvéru škodlivá alebo nie, a či je vzorka softvéru škodlivá, určiť jeho typ. Preto boli techniky detekcie malvéru neustále zdokonaľované, aby zvládli rýchlu evolúciu malvéru.

Mnohé navrhované prístupy sú založené na klasifikačných technikách založených na ML. Klasifikácia môže byť binárna, teda rozdeľuje softvérové vzorky do dvoch tried, a to škodlivé a neškodlivé (neškodné/normálne), alebo multi-triedna, teda klasifikuje vzorku podľa rôznych kategórií malvéru, ako sú backdoor, trójsky kôn alebo vírus.

Schopnosť správne klasifikovať malvér pomáha získať poznatky o jeho správaní a tým lepšie sa proti nemu brániť. Detekcia malvéru je však náročná kvôli technikám zahmlievania (angl. obfuscation) a modifikácie správania, ktoré autori malvéru používajú.

Existuje niekoľko rodín malvéru, ktoré sa zvyčajne líšia podľa svojho cieľa a prístupu:

- Adware
- Malware bez súborov
- Vírusy
- Červy
- Trójske kone
- Botnety
- Ransomware
- Downloader

Analýza malvéru určuje dopad, pôvod a cieľ malvéru. Takáto analýza môže mať rôzne ciele, konkrétne:

- Detekcia malvéru
- Analýza podobnosti malvéru
- Detekcia kategórie malvéru: na detekciu konkrétneho typu malvéru vzhľadom na danú taxonómiu malvéru.

Analýza malvéru sa zvyčajne delí na statickú, dynamickú alebo hybridnú. Statická analýza sa zameriava na spustiteľný kód a štruktúru malvéru bez jeho spustenia. Statická analýza je užitočná na identifikáciu škodlivých softvérových vzoriek a ich odlíšenie od neškodných tým, že generuje odtlačky prstov (fingerprints) škodlivých vzoriek. Odtlačky prstov škodlivých vzoriek pomáhajú jednoznačne identifikovať známy malvér. Príznaky analyzované počas statickej analýzy zahŕňajú reťazce, hash-e súborov, importy DLL, opcodes, API volania a metadáta spojené s prenosným spustiteľným súborom PE (Portable Executable). Statická analýza však nie je veľmi účinná, keď autori malvéru používajú dynamicky sťahované dáta, balenie malvéru, zahmlievanie (obfuscation) zdrojov a metódy proti disassemblácii.

Na rozdiel od statickej analýzy, dynamická analýza monitoruje správanie malvéru počas jej vykonávania. Zvyčajne sa vykonáva v kontrolovanom virtuálnom prostredí, napríklad v sandboxe. Akcie vykonávané malvérom sa dynamicky sledujú a rozširujú na sledovanie prevádzky, API volaní, úprav v registroch, analýzy pamäte a úprav systémových súborov.

Hybridná analýza kombinuje statickú a dynamickú analýzu. Spočiatku sa zdrojový kód analyzuje bez jeho spustenia a neskôr sa analýza vykonáva spustením kódu v kontrolovanom virtuálnom prostredí. Inými slovami, hybridná analýza sa zaoberá analýzou signatúr malvéru a zlepšovaním analýzy kombinovaním s dynamickými parametrami. Čo sa týka detekcie, existujú dve hlavné techniky a to technika založená na signatúrach a technika založená na správaní. Detekcia založená na signatúrach je založená na odtlačkoch prstov, ktoré jedinečne charakterizujú rôzne typy malvéru. Má nízky počet falošných hlásení, ale dá sa obísť úpravami kódu malvéru. Jeho hlavné obmedzenie je však v prípade neznámeho malvéru. Detekcia založená na správaní dynamicky monitoruje akcie malvéru a je účinná pri detekcii neznámeho malvéru. Avšak je náchylná na vysoký počet falošne pozitívnych výsledkov. Okrem techník založených na signatúrach a správaní sa existujú techniky založené na špecifikáciách, ktoré ukladajú určité pravidlá špecifikujúce platné správanie, aby sa odlíšilo od škodlivého správania. Pravidlá sú založené na extrahovaných parametroch alebo príznakoch a môžu byť použiteľné pre siete alebo systémy v závislosti od miesta detekcie škodlivého softvéru. Avšak, keď malvér používa techniky zahmlievania kódu, pravidlá je potrebné často aktualizovať, čo zvyšuje zložitosť systémov monitorujúcich škodlivé správanie. Na odstránenie nedostatkov existujúcich techník boli navrhnuté techniky založené na strojovom učení pre zlepšenie detekcie a klasifikácie malvéru a tiež na spracovanie veľkých dát generovaných komerčnými spravodajskými kanálmi hrozieb. Detailnejšie informácie ohľadom malvéru, jednotlivých experimentoch a výsledkoch možno nájsť v podkladovom dokumente Analýza ML metód a algoritmov pre detekciu malware.docx.

6.1 Experimenty a výsledky

Zo širšieho spektra škodlivého softvéru sme sa v našich experimentoch zmerali na malvér typu **Trójsky kôň** (angl. trojan horse). Pre experimenty bol použitý dataset Trojan detection, ktorý obsahuje záznamy o tokoch v sieťovej prevádzke obsahujúci záznamy o neškodnej a škodlivej prevádzke. Viac informácií možno nájsť v podkladovom dokumente Analýza ML metód a algoritmov pre detekciu malware 2.docx. Ako zhrnutie možno uviesť nasledovné. Na základe vykonaných experimentov možno zhodnotiť, že najvhodnejšie metódy pre detekciu malvéru typu „trojský kôň“ z prevádzky v sieti sú nasledovné:

- Random Forest: úspešnosť 99,99%, čas vyhodnocovania ~9 sek.
- AdaBoost: úspešnosť 99,96%, čas vyhodnocovania ~19 sek.
- DL FFNN: úspešnosť 98,39%, čas vyhodnocovania ~18 sek.

Z uvedených algoritmov vykonal Random Forest výpočty dvakrát rýchlejšie. Metóda DL FFNN (Deep Learning Feed Forward Neural Network) má dlhší čas trénovania modelu oproti ostatným metódam, čo je však štandardná vlastnosť neurónových sietí (obzvlášť hlboké siete). Najrýchlejšou metódou bol Gaussian Naive Bayes klasifikátor – avšak dosiahla najhoršie výsledky (91.49 %).

Za najvhodnejšou metódou na detekciu malvéru typu Trojský kôň analyzovaním sieťovej prevádzky považujeme algoritmus **Random Forest**.

6.2 Popis vybraného algoritmu

Vzhľadom k tomu, že popis nášho vybraného algoritmu sa nachádza v časti 5.2, duplicitne ho tu nebudeme uvádzať.

6.3 Záver

Táto kapitola sa zameriavala na problematiku detekcie, klasifikácie a identifikácie malvéru, ktorý aj napriek dlhodobej existencii bezpečnostných mechanizmov zostáva jednou z

najvýznamnejších hrozieb v oblasti kybernetickej bezpečnosti. Analýza poukázala na rozmanitosť malvérových rodín, techník šírenia a spôsobov maskovania, ktoré útočníci využívajú na obchádzanie tradičných ochranných mechanizmov založených na signatúrach a statických pravidlách.

V teoretickej časti kapitoly boli porovnané základné prístupy k analýze malvéru, konkrétne statická, dynamická a hybridná analýza, spolu s ich výhodami a obmedzeniami. Zároveň bolo zdôraznené, že moderné techniky zahmlievania a modifikácie správania výrazne znižujú účinnosť izolovaných analytických prístupov, čo vytvára priestor pre využitie dátovo riadených metód strojového učenia, schopných pracovať s veľkým objemom heterogénnych údajov.

Experimentálna časť kapitoly bola zameraná na detekciu malvéru typu trójsky kôň na základe analýzy sieťovej prevádzky. Výsledky experimentov preukázali, že algoritmy súborového učenia dosahujú veľmi vysokú úspešnosť detekcie, pričom algoritmus Random Forest vykázal najlepší pomer medzi presnosťou klasifikácie a časovou náročnosťou spracovania. Dosiahnuté hodnoty úspešnosti potvrdzujú schopnosť tohto prístupu efektívne rozlišovať medzi škodlivou a legitímnou prevádzkou aj v prostredí s vysokou mierou dátovej variability.

Porovnanie s ďalšími metódami, ako sú AdaBoost a hlboké neurónové siete typu FFNN, poukázalo na kompromisy medzi presnosťou, výpočtovou náročnosťou a praktickou použiteľnosťou v reálnom prostredí. Zatiaľ čo hlboké neurónové siete poskytujú vysokú presnosť, ich vyššie nároky na tréning a interpretáciu modelu môžu predstavovať obmedzenie v operatívnych bezpečnostných systémoch. Random Forest sa v tomto kontexte ukázal ako robustné a dobre interpretovateľné riešenie vhodné pre nasadenie v prostredí SOC.

Na základe dosiahnutých výsledkov možno konštatovať, že klasifikačné algoritmy strojového učenia, najmä metódy súborového učenia, predstavujú efektívny nástroj na podporu detekcie a identifikácie malvéru na úrovni sieťovej prevádzky. Táto kapitola tak dopĺňa predchádzajúce časti práce o ďalší dôležitý aspekt využitia ML v kybernetickej bezpečnosti a poskytuje metodický základ pre integráciu malvérových detekčných mechanizmov do komplexného SOC riešenia.

7 ZÁVER – NÁVRH

Tento dokument analyzuje vybrané kategórie kybernetických hrozieb a na základe tejto analýzy navrhuje prakticky realizovateľné riešenia využívajúce algoritmy strojového učenia a umelej inteligencie v prostredí bezpečnostného operačného centra (SOC). Dokument sa zameriava na hrozby, ktoré v reálnej prevádzke predstavujú významnú záťaž pre SOC analytikov, a to najmä z hľadiska objemu dát, časovej náročnosti spracovania a potreby rýchlej a presnej reakcie.

Na základe teoretickej analýzy a experimentálneho overenia bolo navrhnuté využitie viacerých tried ML algoritmov, ktoré sú vhodné pre rôzne fázy spracovania bezpečnostných incidentov, nakoľko použitie len jednej je nemožné. V oblasti detekcie phishingových útokov a sociálneho inžinierstva sa ako perspektívne ukázali metódy učenia bez učiteľa, predovšetkým algoritmus HDBSCAN, schopný identifikovať neštandardné a neznáme vzorce správania v bezpečnostných logoch. Pre úlohy masovej klasifikácie, ako je rozlišovanie spam/ham komunikácie, boli navrhnuté klasifikačné algoritmy typu XGBoost, ktoré poskytujú vysokú presnosť pri zachovaní nízkych časových nárokov. V prípade detekcie pokročilých APT techník a malvéru v sieťovej prevádzke boli ako najvhodnejšie identifikované metódy súborového učenia, predovšetkým Random Forest, ktoré ponúkajú vyvážený pomer medzi presnosťou, robustnosťou a interpretovateľnosťou výsledkov.

Navrhované riešenia sú koncipované ako analytická vrstva integrovaná do existujúceho SOC workflow, ktorá nadväzuje na tradičné bezpečnostné mechanizmy, ako sú SIEM, IDS/IPS alebo EDR systémy. Algoritmy strojového učenia v tomto kontexte neposkytujú autonómne rozhodovanie, ale slúžia ako podpora detekcie, triedenia a prioritizácie bezpečnostných udalostí, čím výrazne znižujú manuálnu záťaž analytikov a skracujú čas potrebný na identifikáciu relevantných incidentov.

Výsledkom navrhovaného prístupu je ***AI/ML-powered SOC***, v ktorom sú dátovo riadené analytické metódy využité na systematické spracovanie veľkých objemov bezpečnostných dát a na podporu rozhodovania bezpečnostných tímov. Dokument zároveň ukazuje, že efektívne nasadenie ML v SOC si vyžaduje kombináciu technických opatrení, kvalitných vstupných dát a jasne definovaných procesov, pričom ľudský faktor zostáva neoddeliteľnou súčasťou bezpečnostných operácií.

Navrhnuté riešenia vytvárajú metodický a technologický základ pre ďalší rozvoj SOC smerom k vyššej úrovni automatizácie a analytickej vyspelosti, pričom zachovávajú kontrolu nad rozhodovacími procesmi a rešpektujú praktické obmedzenia reálnej prevádzky. Tento prístup umožňuje postupnú transformáciu SOC na moderné, dátovo orientované a AI/ML-podporované bezpečnostné prostredie schopné efektívne reagovať na súčasné aj budúce kybernetické hrozby.

PRÍLOHY

Súbor: V7 Príloha (KPB3) – Analýza ML metód a algoritmov pre detekciu spam emailových správ.docx

Podkladové dokumenty:

Analýza ML metód a algoritmov pre detekciu APT.docx

Analýza ML metód a algoritmov pre detekciu malware 2.docx

Analýza ML metód a algoritmov pre detekciu phishing.docx

Analýza ML metód a algoritmov pre detekciu sieťových anomálií.docx

Analýza ML metód a algoritmov pre detekciu spam.docx

Clustering.pdf

DAPT_at_MLHat2020.pdf

DECISION TREES - CLASSIFICATION AND REGRESSION TREES - CART.pdf

Ensamble metody obecně Bagging - náhodné lesy - Boost.pdf

KREDITNÝ SKÓRING POMOCOU NÁHODNÝCH LESOV.pdf

Machine Recognition of DDoS Attacks Using Statistical Parameters.pdf

podklad pre V7 v rámci KPB3_Návrh ML algoritmov.docx