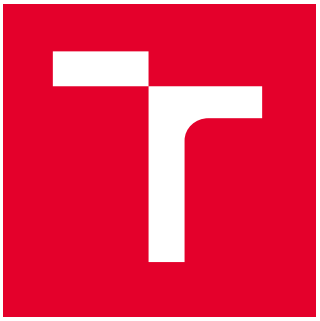


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

CENTRÁLNÍ ZPRACOVÁNÍ A VYHODNOCOVÁNÍ BEZPEČNOSTNÍCH UDÁLOSTÍ

CENTRAL PROCESSING AND EVALUATION OF SECURITY EVENTS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE — **Dominik Žáček**

AUTHOR

VEDOUCÍ PRÁCE — **Ing. Daniel Paučo**

SUPERVISOR

BRNO 2022

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní program **Informační bezpečnost**

Ústav telekomunikací

Student: Dominik Žáček ***ID:*** 221587

Ročník: 3 ***Akademický rok:*** 2021/22

NÁZEV TÉMATU:

Centrální zpracování a vyhodnocování bezpečnostních událostí

POKYNY PRO VYPRACOVÁNÍ:

Práce má za úkol vytvořit systém, na který budou odesílány bezpečnostní události z různých instancí systému pro detekci anomálií od firmy Flowmon Networks. Tyto události budou následně korelované a bude počítána závažnost IP adres, které dané události zapříčinily. Závažné IP adresy pak bude možné zpětně propagovat do systémů pro detekci anomálií. Výstupem je software, který bude přijímat události ze systémů Flowmon ADS ve formátu JSON, zároveň dokáže korelovat události pomocí IP adres a doménových jmen, pro které bude vypočítána závažnost. Systém bude v pravidelných intervalech nejzávažnější IP adresy a domény publikovat v podobě jednoduchého CSV formátu.

DOPORUČENÁ LITERATURA:

[1] Bartos, V., Zadnik, M., Habib, S.M. and Vasilomanolakis, E., 2019. Network entity characterization and attack prediction. Future Generation Computer Systems, 97, pp.674-686.

[2] Uživatelská příručka 11.3 - Anomaly Detection System [online]. Flowmon Networks: ©2021 [cit. 16.9.2021]. Dostupné z: https://demo.flowmon.com/doc/adsplug/locale/cz/

Termín zadání: 7.2.2022 ***Termín odevzdání:*** 31.5.2022

Vedoucí práce: Ing. Daniel Paučo

doc. Ing. Jan Hajný, Ph.D.
předseda rady studijního programu

ABSTRAKT

Práce pojednává o tématu zlepšení bezpečnosti sítí informačních technologií. Jsou odhaleny nedostatky některých současných řešení a je poukázáno na vybrané skutečnosti, kterých je možné využít pro jejich vylepšení. Hlavním tématem i cílem bylo obecně zlepšit bezpečnost sítí zákazníků Flowmon a.s. díky sdílení informací o pachatelích bezpečnostních událostí detekovaných systémem Flowmon ADS. Mezi zákazníky firmy patří například nemocnice, které se mohou stát jedna po druhé obětí stejného útočníka či stejného útoku. Implementací mechanismu, kterým by bylo možné tyto informace mezi zákazníky sdílet, by bylo možné napadením předejít. Byl navržen a implementován systém aby bylo tohoto cíle dosaženo. Nejprve vznikla jedna aplikace odesílající bezpečnostní události k centrálnímu zpracování. Poté byla vytvořena aplikace vystupující jako centrální server, který události přijímá. Byl vytvořen mechanismus normalizace přijatých dat na základě kterých je vytvořeno číslo udávající závažnost události. Tento mechanismus lze pro jednotlivé typy událostí konfigurovat konfiguračním souborem. Nakonec jsou tyto informace vyhodnoceny v jeden jediný údaj takzvané Future Misbehavior Probability score. Každý útočník je tedy ohodnocen skórem od 0 do 1, kdy 1 značí nejzávažnější útočníky. Útočníci jsou poté seskupeni podle skóre a mohou být nasdíleni zákazníkům. Zákazníci díky tomu mohou podniknout různá protiopatření jako například útočníky preventivně zablokovat.

KLÍČOVÁ SLOVA

FMP skóre, Flowmon ADS, Future Misbehavior Probability score, bezpečnost, predikce síťových útoků, prevence útoků, reputace, sdílení bezpečnostních hlášení, sítě informačních technologií

ABSTRACT

The work discusses the topic of improving the security of IT networks. The shortcomings of some of the current solutions are revealed and selected facts are highlighted that can be used to improve the security. The main theme and objective was generally to improve the security of Flowmon customers' networks by sharing information about the perpetrators of security incidents detected by Flowmon ADS. The firm's customers include hospitals, for example, which may fall victim one after another to the same attacker or attack. By implementing a mechanism to share this information between customers, the attack could be avoided. A system has been designed and implemented to achieve this goal. At the beginning, there was one application sending security events for central processing. An application acting as a central server was then created to receive these events. A mechanism has been established to normalize the data received, based on which a number is created indicating the severity of the event. This mechanism can be configured with a configuration file for individual event types. Finally, this information is evaluated in one single piece of data, the so-called Future Misbehavior Probability score. Each attacker is therefore rated between 0 and 1, with 1 indicating the most serious attackers. Attackers are then grouped by score and can be shared with customers. This allows customers to take various countermeasures, such as pre-emptively blocking the attackers.

KEYWORDS

FMP score, Flowmon ADS, Future Misbehavior Probability score, attack prevention, information technology networks, network attack prediction, reputation, security, security alert sharing

ŽÁČEK, Dominik. *Centrální zpracování a vyhodnocování bezpečnostních událostí*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2022, 42 s. Bakalářská práce. Vedoucí práce: Ing. Daniel Paučo

Prohlášení autora o původnosti díla

Jméno a příjmení autora:	Dominik Žáček
VUT ID autora:	221587
Typ práce:	Bakalářská práce
Akademický rok:	2021/22
Téma závěrečné práce:	Centrální zpracování a vyhodnocování bezpečnostních událostí

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora*

*Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Rád bych poděkoval konzultantovi bakalářské práce panu Ing. Martinu Holkoviči a vedoucímu bakalářské práce panu Ing. Danielu Paučovi za jejich odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Obsah

Seznam obrázků

Úvod

V dnešní době internetu má skoro každá společnost nějaký server, který ale může být napaden a útočník může způsobit obrovské škody. Proto je potřeba se před těmito útoky chránit. Je zde několik zaběhlých možností jak se bránit. Příkladem může být firewall, antivirus, Intrusion Detection Systemem (dále jen IDS) nebo blacklist. A právě výstupem této práce lze velice zefektivnit blacklistování konkrétních škodlivých adres. Před útokem bychom byli samozřejmě nejvíce chráněni, pokud bychom se úplně odpojily od internetu nebo zablokovali veškerou komunikaci a povolili ji pouze explicitně tzv. whitelist. To ovšem nejde ve všech případech například u webového serveru, který z principu musí komunikovat se všemi, případně emailového serveru z obdobných důvodů. Proto někdy nezbývá nic jiného než zakazovat komunikaci s útočníky.

Začátek práce je věnován úvodu do problematiky síťové bezpečnosti. Jsou předestřeny současné metody obrany proti síťovým útokům a jejich nevýhody zároveň s teorií na základě čeho by bylo možné současné metody zlepšit. Následně jsou představena již existující související řešení s touto prací a jaké jsou požadavky takových řešení. Kapitola Flowmon ADS pojednává o poduktu Flowmon Networks a.s. se kterým tato práce spolupracuje tj. přijímá z něj bezpečnostní hlášení ze sítě. Poslední kapitola je věnována samotnému řešení. Je představeno schéma komunikace systému, jakým způsobem se přijaté události normalizují, vyhodnocují, jaká je závislost skóre na čase a uložení událostí a skóre do databáze. Konec kapitoly je věnován módům spuštění a konfiguraci aplikace.

1 Úvod do problematiky

Problematika síťové bezpečnosti je rychle se vyvíjející a obsáhlé téma na které již bylo napsáno nespočet prací. Následující podkapitoly jsou věnovány hlavně možnostem prevence síťových útoků na základě předchozích incidentů útočníka. Je odpovězeno na otázku proč je výhodné centrálně zpracovávat události o bezpečnostních událostech a na základě čeho je možné další chování útočníka predikovat.

1.1 Proč potřebujeme centrální zpracování

Problémem v predikci síťových útoků je schopnost identifikovat útočníka. Pokud budeme blacklistovat příliš mnoho aktérů, tak se legitimní uživatel nedostane k webové službě. Na svůj blacklist můžeme dávat nejškodlivější útočníky, na které jsme v naší síti narazily, takové řešení se nazývá Local worst offender list (dále jen LWOL). Zde je ovšem míra předvídání útoku od útočníka, který se v naší síti ještě neobjevil nulová. Dále se na internetu vyskytují různé blacklisty zveřejňované organizacemi zabývající se bezpečností. Útočníci na těchto listech jsou většinou reprezentováni IP adresami. Tyto listy se nazývají Global worst offender list (dále jen GWOL) a do jisté míry dokáží zamezit nejhorším útočníkům, ovšem pokud útočník změní svoji IP adresu tak je jejich záznam na listu k ničemu. Stejně tak je trendem, že útočník často útočí pouze na nějakou vybranou službu například webový server. Tím pádem některé cíle jsou úplně z hledáčku útočníka. Z tohoto pohledu je vidět, že z globálního blacklistu toho o útočníkovi mnoho nevyčteme. Pouze že byl někým nahlášen, ale nevíme kdy, proč a jak moc byl jeho delikt závažný vůči autorovi nahlášení. Tím pádem je dobré sbírat velká data o útočnících a útocích. Efektivně toho dosáhneme, pokud budeme od obětí data sbírat a centrálně je vyhodnocovat. Na základě vyhodnocení určíme skóre škodlivosti. [1]

1.2 Na základě čeho budeme vyhodnocovat

Bylo například zjištěno že, 1 % nejaktivnějších útočníků je zodpovědné až za 82 % útoků. Dále jak by se dalo předpokládat, tak se potvrdilo, že jakmile je nějaký uživatel registrován jako útočník tak má mnohem větší pravděpodobnost zaútočit znovu než legitimní uživatel. Přibližně 90 % útoků je do 5 dní zopakováno. Po uplynutí této doby je častým jevem, že útočník změní svoji identitu (IP adresu). Na tento jev je také potřeba brát ohled, protože použitá IP adresa může být po útočníkovi přidělena legitimnímu uživateli a pokud bychom nechali záznam na blacklistu na vždy, tak by byl uživateli zabráněn přístup do sítě. [1] V průběhu let se zjistilo, že některé

časti internetu ať už na základě nějakého síťového prefixu, autonomního systému [3], nebo územní oblasti mají mnohem větší počet útočníků než jiné. V odborné literatuře se pro tyto celky ujal pojem Bad Neighbourhoods. Do češtiny bychom tento výraz mohli přeložit jako špatná sousedství. Jejich vznik se přisuzuje špatným nastavením bezpečnosti v konkrétní síti. Toho je následně využito útočníky. Ti kompromitují často velkou část zařízení v takové síti, ze kterých udělají takzvané zombie nebo boty, kteří plní příkazy botmastera. Následně Botmaster využívá svoje boty k šíření nejrůznějších škodlivých aktivit v síti, od rozesílání spamu přes phishing až po DDoS útoky.[2, 5]

2 Stávající řešení

V současné době existuje několik druhů řešení pro síťovou bezpečnost. Obecně počítačová bezpečnost je rychle se rozvíjející odvětví a způsobů řešení je nepřeberné množství. Vhodnost vybraného řešení vždy závisí na konkrétních požadavcích na bezpečnost dané sítě, jako je například velikost investice do bezpečnosti, velikost chráněných aktiv, zda-li je vhodné aby služba preferovala spíše falešné pozitivní nebo falešné negativní hlášení atp. Tato práce se dále zaobírá zejména platformou pro sběr dat o škodlivých aktivitách útočníků a jejich sdílením s ostatními uživateli platformy. I tyto platformy jsou již běžně rozšířené[4]. V České republice stojí za zmínku například systém Warden od zájmového sdružení vysokých škol a Akademie věd České republiky CESNET, který umožňuje právě sdílení a následné využívání kolektivních informací [8].

2.1 Network Entity Reputation Database System

Aby bylo možné informace o útocích efektivně sdílet je potřeba je ukládat do databáze a nějak hodnotit. Takový systém se nazývá Network Entity Reputation Database System zkráceně NERD. Hlavní funkcí je obecně ukládání různých bezpečnostních událostí. Ukládat takové informace poskytuje mnoho výhod. Lze je sdílet a ochránit tak před podobným útokem další potenciální oběti. Dále je také samozřejmě více než vhodné využít tyto informace pro lepší zabezpečení samotné sítě, která událost zaznamenala. Může tak buď s útočníkem úplně zablokovat komunikaci, lépe zabezpečit cílenou službu nebo nainstalovat obranný mechanismus proti danému útoku. Hlavním problémem v takových systémech bývá návrh vhodné datové struktury. Datová struktura musí být jednotná, aby byla vhodná pro další případné automatické zpracování. Naproti tomu musí být ale také dostatečně flexibilní aby byla schopna pojmout co nejvíce informací o úplně jiných typech útocích od jiných detektorů atp. V situaci, kdy se záznamy z NERD plánují sdílet s ostatními je také požadavek na anonymitu oběti, aby nebylo možné záznamu o útoku nějakým způsobem zneužít.[7, 10]

2.2 Malware Information Sharing Platform

Jedna síť má pouze velice omezené informace co se děje v celém internetu. V podstatě ví jen o tom co se děje v síti případně na hranicích sítě. Z toho plyne, že sama nemůže žádným způsobem předpovídat trendy útoků ani útoky samotné aniž by na síť bylo takovým způsobem již zaútočeno. Což je samozřejmě velice neefektivní. V tomto scénáři by byla sítí chráněná aktiva nejprve poškozena nebo ukradena a až na

druhý útok stejného typu nebo od stejné síťové entity by byla síť schopna reagovat. Z tohoto důvodu je ve společnosti velká iniciativa pro používání platformy sdílející bezpečnostní události. [11] Na základě této iniciativy vznikl open source projekt zvaný Malware Information Sharing Platform zkráceně MISP. Uživatelé této platformy tedy sdílejí a dostávají bezpečnostní hlášení od jiných uživatelů. Platforma má několik komunit. Komunita napojena na Computer Incident Response Center Luxembourg zkráceně CIRCL je nyní oficiálně využívána více než 800 společnostmi, ovšem reálně bude číslo daleko větší protože projekt je možné používat privátně. Platforma dokáže provádět automatické korelace. Podporuje standard Structured Threat Information eXpression zkráceně STIX a s tím tedy možnost sdílet bezpečnostní incidenty s ostatními platformami využívající tento standard. Bezpečnost platformy je řešena pomocí Pretty Good Privacy zkráceně PGP. Díky tomu, že je open source lze doprogramovávat svoje vlastní moduly v jazyce Python. Dále je možné data libovolně exportovat a importovat. [9]

2.3 Další z oblasti

Jak bylo již zmíněno je další množství systému poskytujících obecně služby v oblasti odborně zvané Threat Intelligence Sharing Platforms. Vzniklo i několik standardů, aby bylo možné informace sdílet mezi různými platformami. Jedněmi z nejvýznamnějších standardů jsou např. Cyber Observable eXpression zkráceně CybOX nebo STIX. Příkladem nejznámějších platforem jsou Microsoft Interflow, Open Threat Exchange, McAfee Threat Intelligence Exchange, Facebook Threat Exchange, IBM X-Force Exchange, Collective Intelligence Framework a také již zmiňovaný MISP. Platformy jsou k dispozici jak komerční, open source tak i plně zdarma, byť tedy většina zmíněných a obecně používaných jsou právě komerční s uzavřeným zdrojovým kódem. [12]

2.4 Funkce a požadavky

Hlavní funkcí všech platforem pro sdílení bezpečnostních událostí je jak již název napovídá umožnit sdílení bezpečnostních událostí mezi jednotlivými síťovými entitami za účelem vyšší bezpečnosti sítě uživatelů platformy. Uživatelé těchto platforem mohou být od jednotlivců přes soukromé firmy až po státní organizace. Vzhledem k množství a frekvenci incidentů je nutná také automatizace sdílení a vyhodnocování těchto hlášení. Sdílené informace by měli být vždy relevantní a výstižné aby byly snadno a efektivně využitelné v obraně vůči stejnému útoku. V žádném případě by neměli zahlcovat uživatele platforem. Požaduje se také bezpečnost platformy, protože

se sama může lehce stát cílem útočníků, vzhledem k tomu že obsahuje pro útočníky velice cenné informace. Z toho důvodu je také požadována od platforem jistá míra anonymity a také důvěryhodnosti. Další funkce se mohou velice lišit závisle na platformě, protože výklad definice platforma pro sdílení bezpečnostních událostí si může každý výrobce vykládat jinak. Také není nikde přesně vymezeno co se za takovou platformu může a nemůže považovat. Některé platformy tedy umí pouze například sdílet bezpečnostní události jiné je zvládnou i vyhodnotit. [12, 13]

3 Flowmon ADS

Flowmon ADS je jedním z modulů produktu Flowmon společnosti Progress. Produkt je Security Information and Event Management systém zkráceně SIEM, který analyzuje síťové toky. Mezi odběratele tohoto produktu patří například společnosti Seznam.cz, Raiffeisenbank, a.s., Siemens, s.r.o. nebo Vodafone. Produkt je dostupný buď jako on-premise řešení nebo jako cloudová služba. Další moduly jsou Dashboard and Reports, Monitoring Center, Application Performance Monitoring, Application Performance Monitoring TG, Packet Investigator, Flowmon DDoS Defender. [15] Tato práce se dále zabývá zejména modulem Flowmon ADS. Ten se řadí do kategorie Anomaly Detection System zkráceně ADS. Podporuje několik standardů jako například Netflow v5 a Netflow v9, implementuje standart RFC 5103. Lze do velké míry automatizovat například posíláním na email, do syslog nebo vytvářením vlastních scriptů. Události je možné také automatizovaně mazat. Do modulu je také integrován framework MITRE ATT&CK. V uživatelském rozhrání modulu jsou záložky *Analýza*, *Události*, *Reporty*, *Nastavení*, *O aplikaci*. [14]

3.1 Záložka událostí

Pro tuto práci je nejdůležitější záložka událostí obsahující tabulku s bezpečnostními hlášeními. Jednoduchý seznam je možné filtrovat podle několika kritérií. Prvním z nich je datum, kdy při zvolení tohoto filtru budou zobrazeny pouze události z vybraného časového období. Další filtr perspektiva umožnuje výběr z možností Bittorrent Download, Cryptocurrency Mining, Massive Spamming, Ransomware, Security issues, Teamviewer a Webshare Download. Dále je možné filtrovat podle zdrojové IP, která způsobila incident a cílů. Použití pokročilejších filtrů je také možné po kliknutí na tlačítko s popisem Více filtrů. Tabulka základně obsahuje sloupce *pořadí události*, *ID události*, *čas detekce události*, *priorita*, *typ události*, *podtyp události*, *zdroj*, *detail*, *cíle*, *zdroj dat*, *akce*. Další sloupce je možné dodatečně přidat. Po kliknutí na tlačítko ve sloupci *akce* je vyvoláno dialogové okno s možnostmi zvolit *Detail události*, *Záznam události*, *Související události IDS*, *Vizualizovat událost*. Data lze exportovat do CSV souboru. [14]

3.2 Záložka analýza

Záložka analýza analyzuje události z výše zmiňované záložky událostí. Je možné analyzované události filtrovat podle data, perspektivy, zdroje dat a nebo zdrojové IP. Filtr perspektiv má stejné možnosti jako v záložce událostí. Prvním úsekem na záložce od shora je graf s analýzou toků v průběhu času. Pod ním je úsek analyzující

události v průběhu času, které je možné seskupit buď podle priority nebo metody. Posledním úsekem jsou události podle priority. Zde je zobrazen celkový počet událostí. Události jsou roztříděny podle priority do skupin jako je například *SCANS*, *UPLOAD* nebo *DICTATTACK*. Skupiny je možné rozkliknout, přičemž se zobrazí zanořená tabulka se sloupci Zdrojová IP adresa, Filtry zdrojové IP, Počet událostí a Akce. Po rozkliknutí libovolné položky v tabulce se nám zobrazí další zanořená tabulka velice podobná tabulce událostí, kde jsou vyfiltrované události pro konkrétní prioritu útoku a zdrojovou IP adresu. I zde je možné kliknout na tlačítko akce, které vyvolá stejné okno jako na řádku jednotlivé události v záložce událostí.[14]

3.3 Způsoby detekce

Detekované metody se dělí do kategorií síťové anomálie, vzdálený přístup, detekce útoků, detekce nechtěných služeb. Do kategorie detekované anomálie spadá pokud sítí protékají velká data, určitá služba přestane být dostupná, dlouhodobé chování sítě se vychýlí z normálu, byla detekována heterogenní nebo přímá internetová komunikace, v síti byla zjištěna možná přítomnost parazitních zařízení. Do kategorie vzdáleného přístupu spadá zejména detekce telnet protokolu. Kategorie útoků je detekována při slovníkových útocích na služby HTTP, IMAP, FTP, VNC, POP3, RDP, SMTP, SSH, Telnet, Samba dále při detekci DoS útoků, scanování sítě pomocí TCP a detekce odchozího spamu. Mezi nechtěné služby potom patří detekce BitTorrent P2P sítě, používání TeamVieweru, VOIP protokolů, připojení k VPN a tunelování. Různé detekční metody je možné v systému jednotlivě i hromadně zapínat případně vypínat. Většina metod má podmetody jako například SCAN může mít různé druhy jako například ARP scan nebo UDP scan. V podkapitolách této sekce jsou rozebrány podrobně vybrané detekční metody, podmetody a jejich parametry.[14]

3.3.1 BITTORRENT

V této detekční metodě se využívá heuristiky a monitorování toku sítě. Lze nastavit parametry detekční metody. LANFilter umožňuje nastavením IP adresy ignorovat například místní síť, aby nedocházelo k planým poplachům. *MinSeeds* je minimální počet zařízení od kterých musí být stahováno aby byla vytvořena událost. *MinHighPorts* určuje minimální počet připojení na portech vyšších než 10240. Dále je možné nastavit parametr *MinimalProbability*, která určuje minimální procento získané z heuristik pro vytvoření události.

3.3.2 BLACKLIST

Slouží k detekci komunikace s blacklistovanými síťovými entitami. Blacklist může být buď standardně firmy Flowmon nebo vlastní. Podmetody jsou *Host*, *Service*, *Web*, *Domain*. Detekční metoda má několik parametrů. *IgnoreUnreachable*, *IgnoreUnsuccExt* a *IgnoreUnsuccInt* nastavují, že nebude vytvořena událost pokud blacklistovaná entita není nějakým způsobem dostupná. *IgnorePorts* nastavuje vynechané porty při detekování. *ActiveBlacklists* je seznam používaných blacklistů. Pokud je vytvořena událost touto metodou často to může znamenat kompromitované zařízení v síti snažící se komunikovat s útočníkem.

3.4 Parametry události

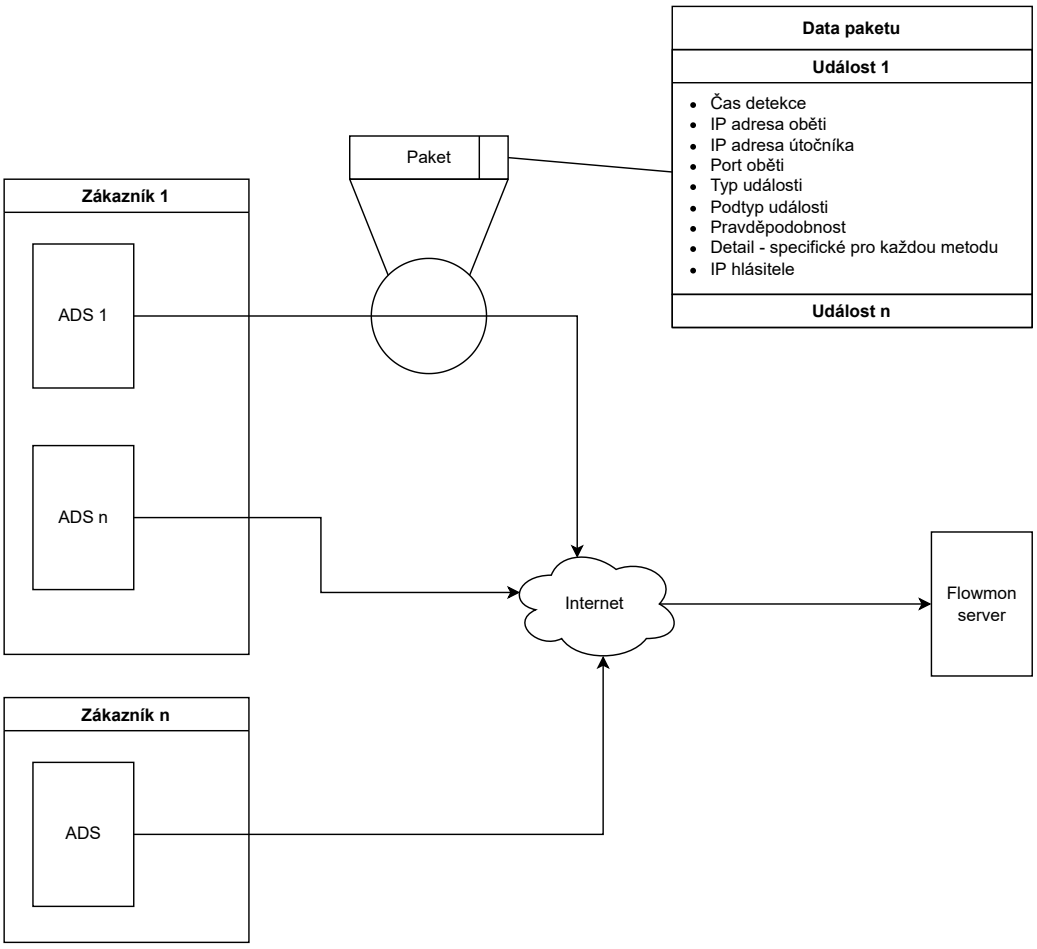
Kliknutím na tlačíko akce a výběrem možnosti Detail události se zobrazí dialogové okno s kompletním detailem konkrétní události. Nadpisem okna je Událost č.XXX, kde XXX je ID konkrétní události. Prvním odstavcem je typ události, může to být například již zmiňovaná kategorie *DICTATTACK*. Dalším odstavcem je podtyp události, kdy podtyp je závislý na typu události. Tedy u zmíněné kategorie *DICTATTACK* může být podtyp například *RDPProtocol*. Třetím odstavcem je Detail události, který se skládá z textu který popisuje událost například *Slovníkový útok na RDP, pokusy: 571, porty: 3389, délka trvání útoku: 3 m 21 s 209 ms, průměrný čas mezi pokusy: 352 ms.* Pod detailem je odstavec MITRE ATT&CK, kde je klasifikována použitá technika a taktika podle frameworku MITRE ATT&CK. V další sekci okna jsou informace závislé na typu útoku rozdělené do tří vertikálních částí. Příkladem takových informací mohou být Čas detekce, Naposledy aktualizováno, První tok, Původce události, Zachycené jméno původce, MAC adresa, Identita uživatele, Pravděpodobnost, False positive, Detekováno instancí, Zdroj dat. Poslední sekcí okna je několik záložek také závislých na typu události například Cíle, Komentáře, Kategorie, Atributy, Záznam události, Související události IDS, Záznamy provozu.[14]

4 Návrh a implementace

Byla vytvořena infrastruktura pro odesílání událostí z Flowmon ADS na centrální úložiště, kde jsou události vyhodnocovány a pro každou entitu, v našem případě IP adresu, vypočteno takzvané Future Misbehavior Probability score dále jen FMP skóre. IP adresy je poté možné rozdělit do kategorií podle škodlivosti tj. velikosti FMP skóre a ty následně publikovat do souborů, které je možné snadno distribuovat k zákazníkům. Infrastruktura je tedy tvořena dvěma spolu komunikujícími aplikacemi. Aplikace odesílající událostí z Flowmon ADS na straně zákazníka je ve zdrojových kódech nazvána Client. Druhá aplikace přijímající a vyhodnocující události na straně Flowmon je ve zdrojových kódech nazvána FMPCalculator. Obě aplikace se ovládají pomocí argumentů z příkazové řádky, odkud se aplikace spouští. Pro práci s argumenty je využito knihovny Python3 *argparse*.

4.1 Odesílání událostí z Flowmon ADS

Bezpečnostní události jsou přijímány z existujícího řešení Flowmon ADS ve formátu JavaScript Object Notation dále jen JSON. Pro testovací účely byla vytvořena možnost odeslat JSON soubor s bezpečnostními událostmi uložený v zařízení. Přílohou této práce jsou všechny JSON soubory použité při jejím vytváření a testování. Pro použití souboru se aplikace Client spouští s argumentem příkazové řádky *-f CESTA*, případně *–file-path CESTA*, kde *CESTA* je cesta k souboru na disku. Dalším argumentem příkazové řádky je *-i ADRESA*, nebo také *–ip ADRESA*, kde *ADRESA* je IPv4 adresa serveru na který se bezpečnostní události odesílají. Argument *-p PORT* nebo také *–port PORT* nastavuje síťový port na který mají být bezpečnostní události serveru odeslány. Po spuštění jsou bezpečnostní události deserializovány pomocí standardní knihovny *json* programovacího jazyka Python. Deserializování vytvoří objekt typu slovník. Slovník je poté převeden na vlastní třídu *Packet*. Třída *Packet* obsahuje pole objektů *SecurityEvent*. Každá událost z Flowmon ADS je převedena v paměti na objekt typu *SecurityEvent*. Ten má vlastnosti využívané k vyhodnocení bezpečnostní události viz Obr. 4.1. Všechna data jsou získána z JSONu datového modelu Flowmon ADS kromě IP hlásitele. Čas detekce je datový typ data a času získán z JSONu jako datetime. IP adresa oběti je datový typ string a je získány z JSONu jako hodnota *ip* v prvním elementu pole *targets*. IP adresa útočníka je také datový typ string a je získána z *ip* v objektu *source*. Port oběti je číselný datový typ získaný z první hodnoty *ports* v objektu *attributes*. Podtyp události je datový typ string získaný ze *subType* v JSONu. Pravděpodobnost je číselný datový typ s plovoucí desetinnou čárkou jehož hodnota může být od 0 do 1. Typ události je datový typ string získaný z type v JSONu. Detail je datový typ string získaný z



Obr. 4.1: Odeslání bezpečnostní události.

detail v JSONu. V modelu každé události se vyskytují pouze poslední dvě vlastnosti tj. Typ a Detail. Ostatní vlastnosti jsou závislé na typu události a tedy v modelu být mohou ale nemusí. IP hlásitele je vyplněno IP adresou zařízení, které události nahlásilo. Pokud je program spuštěn v testovacím módu tj. s argumentem příkazové řádky *-t* nebo *–testing* tak je vybráno ze tří náhodných IP adres. Následně je objekt *Packet* převeden na text opět pomocí základní knihovny *json* jazyka Python. Text je zakódován na bity v UTF-8. Bity jsou nakonec odeslány na specifikovanou IP adresu a port k dalšímu zpracování.

4.2 Přenos událostí

Aplikace spolu komunikují přes Transmission Control Protocol dále jen TCP. Tento protokol má velkou výhodu zajištění doručení paketů. Dále pokud jsou pakety odeslány a po trase se změní jejich pořadí na straně příjemce paketů jsou zpět poskládány ve správném pořadí ve kterém byli odeslány. Nevýhodou oproti User Datagram Protocol dále jen UDP je zejména to, že Flowmon používá na svých serverech techniku zvanou load balancing, kvůli které není možné udržovat více než 65535 otevřených spojení. Tato limitace ovšem v této fázi produktu zatím nepřináší žádná reálná omezení. Pokud by se produkt začal využívat vysokým počtem zákazníků a maximální počet otevřených spojení by přestal dostačovat, aplikace by musely být předělány na UDP. V takovém případě by musely být některé funkce TCP implementovány na aplikační vrstvě. Ta by tedy musela navíc implementovat číslování paketů a způsob který by zajistil jejich spolehlivé doručení. Zajistit spolehlivé doručení by šlo například odesláním potvrzovací zprávy o úspěšném doručení, pokud by tedy byl odeslán paket který by se ztratil a odesílatel by neobdržel zprávu o úspěšném odeslání paketu, paket by byl odeslán znovu.

4.3 Přijetí událostí serverem

V případě spuštění aplikace FMPCalculator bez parametrů příkazové řádky se aplikace spustí v roli serveru, ve schématu komunikace na Obr. 4.1. zaznačeno jako Flowmon Server, a celou dobu naslouchá pro příchozí připojení. Jakmile je ze zákazníkovy strany odesláno hlášení na tento server, tak se vytvoří nové vlákno které dále zpracovává data. Vytvořením nového vlákna je umožněno pracovat paralelně na zpracování několika událostí zároveň bez nutnosti blokovat hlavní vlákno. Hlavní vlákno tedy nemusí čekat na vstupně výstupní operace databáze ani samotnou deserializaci JSONu a místo toho se věnuje přijímání nových připojení. Přijaté bity se ve vláknu které je přijalo nejprve dekódují formátem UTF-8 do běžného textu. Běžný text je dále deserializován z formátu JSON na Pythonový objekt typu slovník. Každý klíč slovníku je opět převeden na třídu Packet, která obsahuje pole typu *SecurityEvent* jehož definice je cíleně naprosto stejná jako tomu bylo u odesílání události.

4.4 Normalizace událostí

Za účelem lepší strojové práce s událostmi se některé jejich vlastnosti přetvářejí. Přetvoření probíhá zavoláním konstruktoru objektu SecurityEventModel, který přijímá vstupní parametr typu *SecurityEvent*. Proměnná type objektu *SecurityEventModel*

je získána z vstupního objektu převedením textového řetězce označujícího typ události na výčtový typ *SecurityEventType*. Proměnná *sub_type* je získána obdobným způsobem převodu textového řetězce na výčtový typ *SecurityEventSubType*. Rozsah, vlastnost *volume* typu číslo s plovoucí desetinnou čárkou objektu *SecurityEventModel*, je získán z textového řetězce detail objektu *SecurityEvent*. Rozsah nabývá hodnot od 0 do 1, kdy 1 indikuje maximální možný rozsah bezpečnostní události pro danou trojici parametr detailu, typ a podtyp bezpečnostní události nastavený v konfiguračním souboru *MaxParameterValues.csv*. Proměnná *weight* objektu *SecurityEventModel* je desetinné číslo udávající váhu bezpečnostní události nabývající hodnot od 0 do 1. Váha je získána ze stejného konfiguračního souboru jako rozsah pro danou dvojici typ a podtyp bezpečnostní události. Ostatní proměnné objektu *SecurityEventModel* jako *attacker_ip*, *detection_date_time*, *victim_ip*, *victim_port* a *reporterId* jsou namapovány na proměnné obdobného názvu ze vstupního parametru typu *SecurityEvent*. Jakmile je událost v této podobě je připravena k vyhodnocení.

4.5 Vyhodnocení FMP skóre

Objekt *SecurityEventModel* obsahuje všechny doporučené atributy aby bylo možné síťovou entitu, která událost způsobila ohodnotit nějakým FMP skórem. Obsahuje tedy atributy jak bylo doporučeno v ostatních vědeckých pracích čas detekce, unikátní identifikátor škodlivé entity, typ události, kategorie události, rozsah události a identifikátor detektoru. [4, 1] Proces vyhodnocení začne vyhledáním síťové entity *EntityModel* v tabulce *Entities*, která událost způsobila dle jejího unikátního identifikátoru například IP adresou. Díky tomu jsou nalezeny i ostatní bezpečnostní události, které síťová entita způsobila. Pokud ještě nebyla síťová entita zaznamenána tj. není v tabulce nalezena, tak bude vytvořena. Samotné FMP skóre je vyhodnocováno pro dvě různá časová období. Obě období jsou vyhodnocena ručně vytvořenými binárními stromy. Výstupem ohodnocení každého období je tedy hodnota od 0 do 1, kdy 1 je maximální možná škodlivost pro dané období. Finální FMP skóre S, které je poté přiděleno síťové entitě je vypočteno jako $S = \frac{2K+D}{3}$, kde K je vyhodnocené skóre pro krátkodobé období a D je vyhodnocené skóre pro dlouhodobé období.

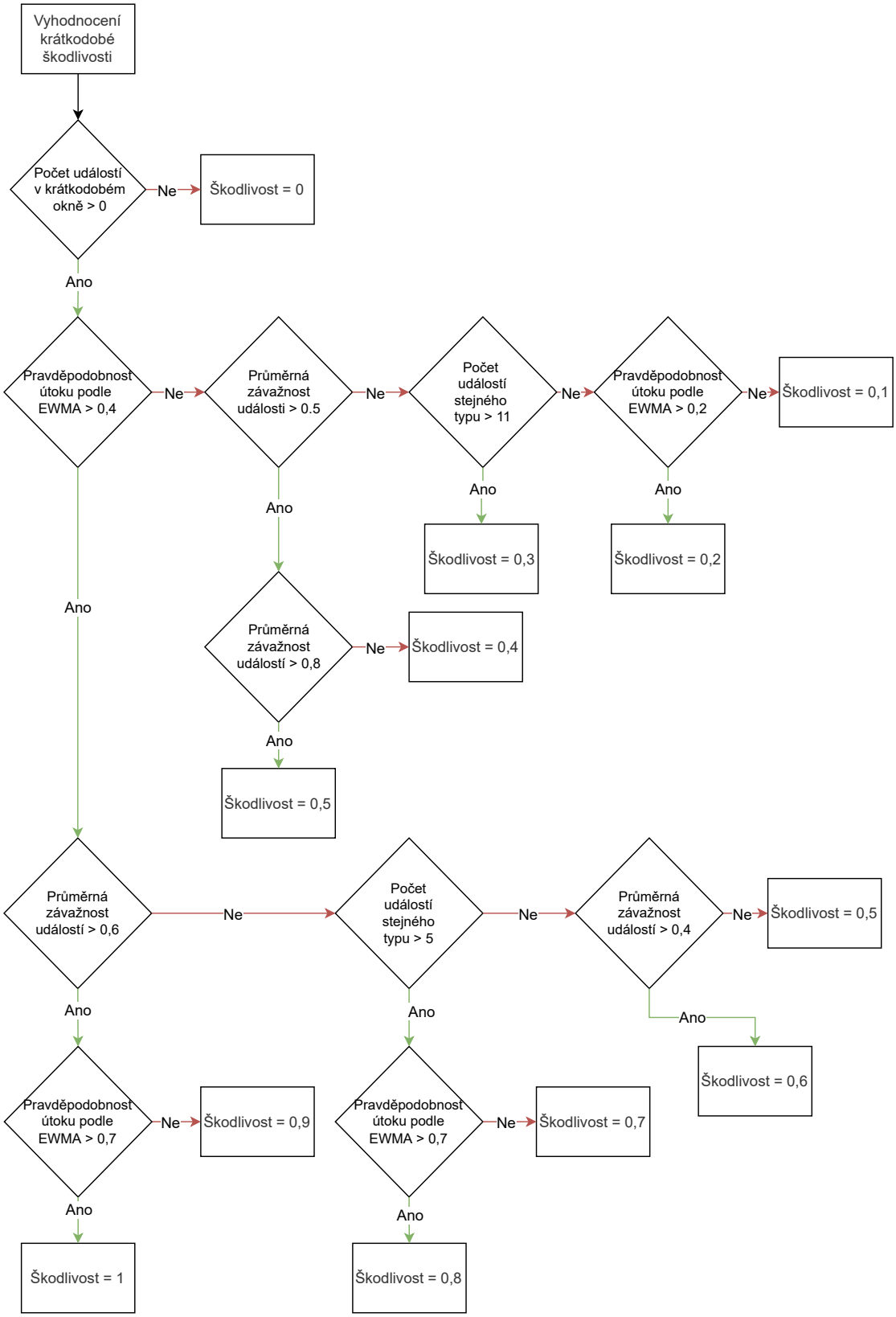
4.5.1 Krátkodobé FMP skóre

Pro vyhodnocení krátkodobého skóre je využito všech událostí způsobených entitou, které spadají do krátkodobého časového okna nastavitelného z příkazové řádky. Pokud není nalezena žádná událost, která by do krátkodobého okna náležela bude vyhodnoceno skóre 0. První rozhodnutí stromu závisí na pravděpodobnosti útoku

podle exponenciálně váženého klouzavého průměru anglicky Exponentially Weighted Moving Average dále jen EWMA. Bylo rozhodnuto, že pro krátkodobou škodlivost je tento identifikátor nejdůležitější a proto byl umístěn do kořene stromu kde je v podstatě rozhodnuto jestli škodlivost síťové entity bude v rozmezí od 0,1 do 0,5 tj. nízká až střední škodlivost nebo od 0,5 do 1 tj. střední až vysoká škodlivost. Pro výpočet EWMA byla implementována vlastní funkce využívající vzorce EWMA.

$$y_t = \frac{\sum_{i=0}^{t} w_i x_{t-i}}{\sum_{i=0}^{t} w_i}$$

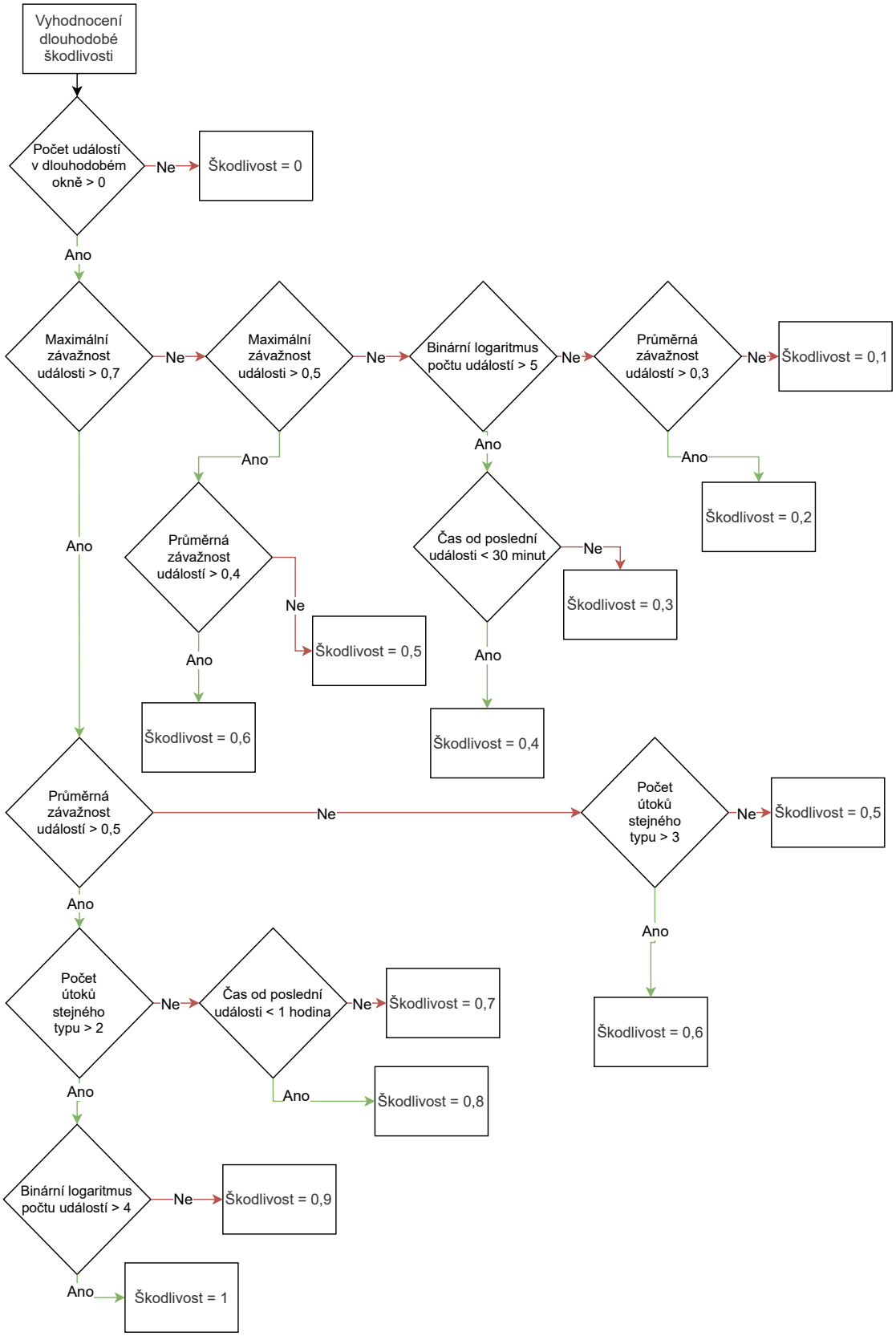
Kde y_t je výsledná pravděpodobnost útoku podle EWMA, t je velikost krátkodobého časového okna, i je hodina krátkodobého okna, x_{t-i} nabývá hodnoty 1 pokud v hodině $t - i$ byla entitou způsobena alespoň jedna událost v ostatních případech je hodnota 0, w_i je váha. Váha se počítá jako $w_i = (1 - \alpha)^i$. Neznámá α je takzvaný uhlazovací faktor vypočten jako $\alpha = \frac{2}{t+1}$. Další rozhodnutí je učiněno na základě velikosti průměrné závažnosti. Průměrná závažnost se získá vypočtením aritmetického průměru závažností $z = weight \times volume$ událostí spadajících do krátkodobého okna, přičemž *weight* je váha události a *volume* je rozsah události. Tohle rozhodování již není v obou větvích vyvážené. Větev směřující k vyššímu skóre má zpravidla vyšší nároky. Využitím tohoto mechanismu lze zmírnit důležitost předchozího uzlu v tomto konkrétním případě kořene stromu. V určitých případech je tedy dokonce možné dosáhnout skóre 0,5 nezávisle na kořeni stromu. Binární strom byl konstruován také s ohledem na kontext situace ve které se rozhodování nachází. Například tedy pokud závažnost události je nízká typicky například scanování sítě tak pro vyšší ohodnocení entity je potřeba těchto událostí více než kdyby události měli vyšší závažnost, nebo také pokud je pravděpodobnost útoku podle EWMA velmi vysoká a události jsou závažné je bez dalších podmínek přiděleno vysoké skóre. Detailní algoritmus vyhodnocení je na Obr. 4.2.



Obr. 4.2: Vyhodnocení krátkodobé škodlivosti.

4.5.2 Dlouhodobé FMP skóre

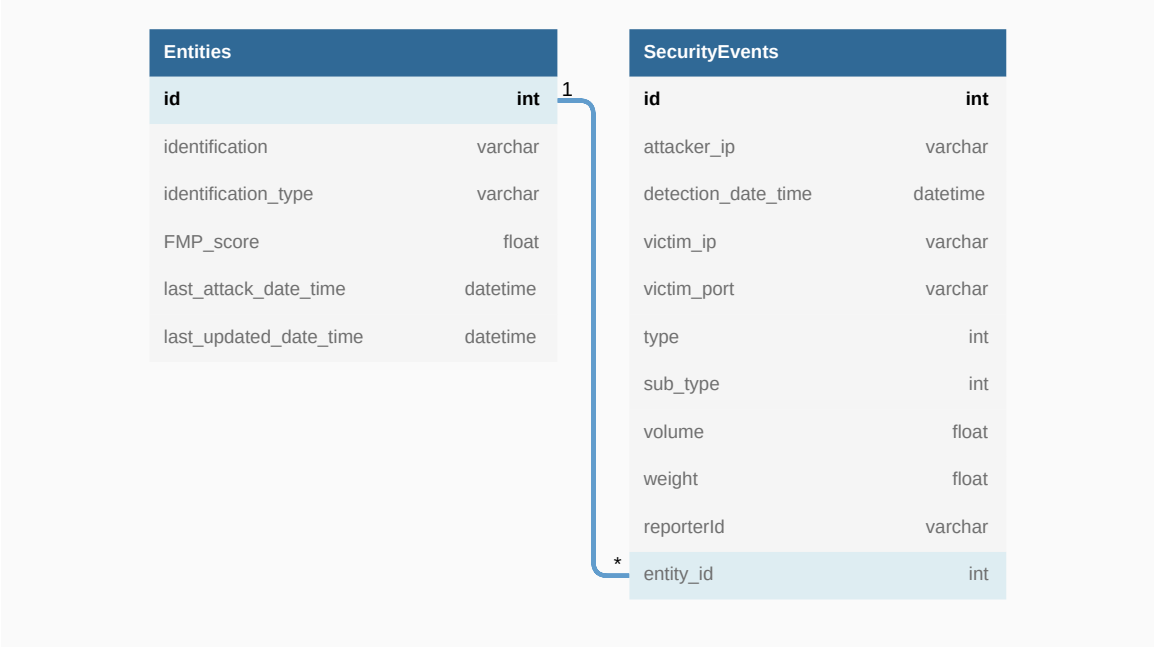
Pro vyhodnocení dlouhodobého skóre je využito všech událostí způsobených entitou, které spadají do dlouhodobého časového okna nastavitelného z příkazové řádky. Pokud žádná událost nespadá do dlouhodobého okna tak nejen dlouhodobé FMP skóre je vyhodnoceno jako 0, ale také výsledné FMP skóre je vyhodnoceno jako 0. Platí totiž že $d \geq k$, kde d je dlouhodobé okno a k je krátkodobé okno tím pádem také platí, že pokud nejsou žádné události způsobené entitou v dlouhodobém okně nemohou být ani v krátkodobém okně. Dlouhodobé FMP skóre na rozdíl od krátkodobého mnohem méně závisí na časových údajích. Čas od posledního útoku je jediným parametrem stromu využívajícím čas a je vždy využit pouze v poslední větvi stromu takže jeho vliv na finální skóre je malý. Obecně by toto skóre mělo spíše poskytovat informaci o nějakém dlouhodobě agregovaném chování pachatele. Kořenem stromu byla zvolena maximální závažnost události, což je jednoduše největší hodnota závažnosti ze všech událostí způsobených síťovou entitou spadajících do dlouhodobého okna. Kořen stromu rozhodne jestli bude skóre buď v rozmezí 0,1 - 0,6 nebo 0,5 - 1. Skóre 0,5 a 0,6 může být tedy získáno nezávisle na výsledku rozhodnutí kořene stromu. To má za výhodu, že pokud nebude například těsně splněna podmínka kořene stromu stejně bude vyhodnoceno vyšší skóre. Stejně jako u krátkodobého skóre byli podmínky tohoto stromu implementovány s ohledem na kontext předchozího rozhodování, například tedy pokud je zjištěno, že události nejsou příliš závažné tak následuje podmínka závislá na jejich počtu, aby bylo i pro viníky méně závažných ale přesto četných událostí vyhodnoceno alespoň středně závažné skóre. Parametry tohoto stromu průměrná závažnost útoku a počet útoků stejného typu jsou získány stejně jako u vyhodnocení krátkodobého okna akorát jsou zahrnuty všechny události až po dlouhodobé okno. Posledním nezmíněným parametrem je binární logaritmus počtu událostí $b = log_2 p$, kde p je počet událostí v dlouhodobém okně. Detailní algoritmus vyhodnocení je na Obr. 4.3.



Obr. 4.3: Vyhodnocení dlouhodobé škodlivosti.

4.6 Uložení událostí

Po vyhodnocení FMP skóre entity a vyplnění časového razítka poslední změny aktuálním časovým razítkem jsou entita a příchozí události uloženy do SQLite databáze. Aplikace využívá k práci s databází Objektově relační zobrazení anglicky Object–relational mapping zkráceně ORM, díky čemuž není potřeba v aplikaci psát přímo Structured Query Language, zkráceně SQL, dotazy či příkazy. Databáze obsahuje tabulky SecurityEvents a Entities. Tabulka SecurityEvents obsahuje sloupec *id* který slouží jako primární klíč, dále obsahuje sloupce dle datové struktury objektu SecurityEventModel. Tabulka také obsahuje Foreign Key, dále jen FK, do tabulky *Entities*. Tento FK slouží pro definování relace 1:N mezi tabulkami *Entities* a *SecurityEvents*. Znamená to tedy, že jedna síťová entita může mít k sobě přiřazených několik bezpečnostních událostí, které způsobila. Tabulka Entities má také sloupec *id* sloužící jako primární klíč, všechny sloupce tabulky jsou schodné s datovou strukturou objektu *EntityModel*. V momentální verzi se *identificator* využívá pouze pro ukládání IP adres. Pozdější rozšíření systému mohou ovšem využít sloupce *identificatorType* a následně možnosti využívat i jiné identifikátory než jen IP, například doménová jména. Schéma databáze je na Obr. 4.4.



Obr. 4.4: Schéma databáze.

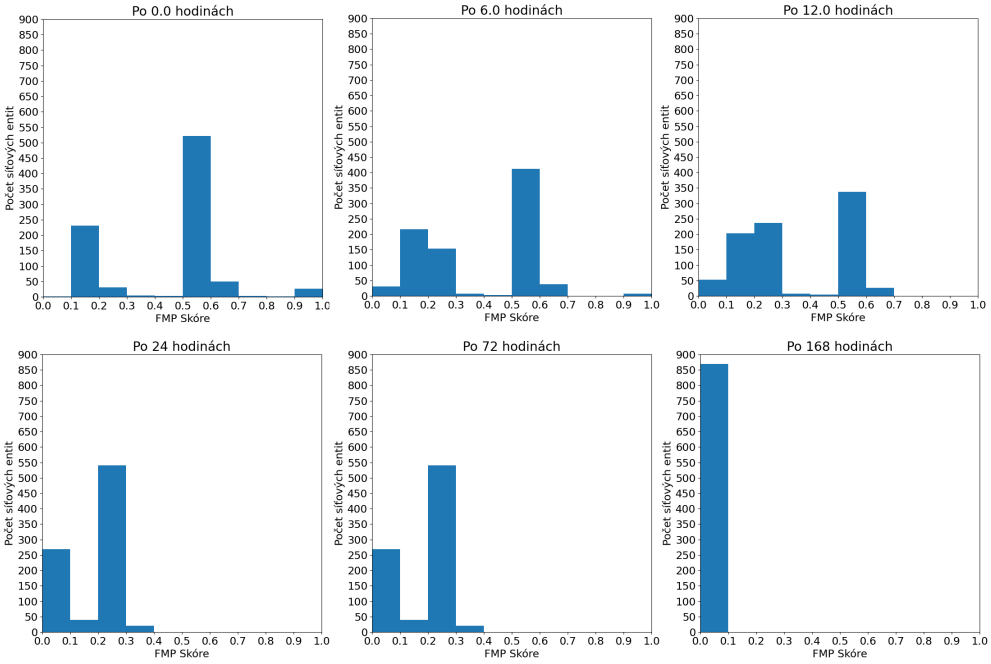
4.7 Opětovné vyhodnocení událostí

Vzhledem k tomu, že FMP skóre je veličina závislá na čase tak je potřeba tuto veličinu kontinuálně znovu v průběhu času vyhodnocovat. V opačném případě by jednou škodlivá entita byla zaznamenána navždy, její skóre by nikdy nekleslo a místo dynamického skóre měnícího se v čase bychom měli v podstatě statický blacklist. Takže bylo nutné do aplikace FMPCalculator funkci znovu vyhodnocení implementovat. Pro spuštění aplikace v tomto módu je potřeba ji spustit s argumenty příkazové řádky *-r LAST_UPDATED* nebo *–reevaluation LAST_UPDATED*, kde *LAST_UPDATED* nastavuje minimální počet hodin uplynutých od časového razítka poslední změny entity aby byla entita znovu vyhodnocena. Aplikaci je tedy možné s těmito parametry jednoduše spouštět kontinuálně například z programu Cron na operačních systémech Linux nebo pomocí programu Task Scheduler na operačním systému Windows. Po spuštění jsou nalezeny všechny entity kde počet hodin od data poslední změny je větší než *LAST_UPDATED*. Pro každou takovou entitu je znovu vyhodnoceno FMP skóre stejným způsobem jaký byl popsán v kapitole 4.5 a změněno datum poslední změny na aktuální časové razítko. Po znovu vyhodnocení všech nalezených entit jsou smazány všechny entity a jimi způsobené události, které mají výsledné FMP skóre rovné 0. Databáze tak nikdy nenaroste do neúnosných rozměrů ani při velkém vytížení.

4.7.1 Vývoj skóre v čase

Celý algoritmus vyhodnocení a opětovného vyhodnocení v čase byl otestován na smíšených reálných a umělých datech. Během testů bylo vyhodnoceno přes 5,5 tisíce různých IP adres a okolo 6,5 tisíc událostí jimi způsobených. Z celkového počtu IP adres okolo 900 způsobilo událost která se vyhodnocuje. Mezi události které se nevyhodnocují patří například hlášení o použití Peer-to-peer sítí nebo IP tunelu. Na Obr. 4.5 jsou vyobrazeny histogramy vývoje FMP skóre v čase. Snímek s nadpisem Po 0.0 hodinách byl pořízen ihned po odeslání a vyhodnocení všech zmíněných dat. Vyhodnocování bylo provedeno s krátkodobým oknem 24 hodin a dlouhodobým oknem 168 hodin. Na tomto snímku je možné vidět jasné rozdělení IP adres do mírně škodlivé, středně škodlivé a vysoce škodlivé skupiny. Po 6 hodinách je velký úbytek FMP skóre hlavně ve vysoce škodlivé skupině a znatelný úbytek ve středně škodlivé skupině hlavně z důvodu jejich závislosti na EWMA, kdy s přibývajícím časem není možné získat příliš vysoké skóre bez nově příchozích událostí. Po 12 hodinách je tento trend ještě více znatelný, vysoce škodlivá skupina se v podstatě celá přesune do středně škodlivé skupiny a původní středně škodlivá skupina se začne ještě více přesouvat do mírně škodlivé skupiny například z důvodu vypadnutí z krátkodobého

okna. Po 24 hodinách již všechny události vypadli z krátkodobého okna a maximální možné FMP skóre, které mohou obdržet je 0,33 dle vzorce popsaného v kapitole 4.5. Skóre je tedy od tohoto okamžiku čistě závislé pouze na dlouhodobé škodlivosti. Po 72 hodinách se histogram skoro nezmění z důvodu velice nízké závislosti dlouhodobé škodlivosti na čase. Po 168 hodinách všechny IP adresy vypadli i z dlouhodobého okna a všechny tedy měli FMP skóre 0.



Obr. 4.5: Vývoj skóre v čase.

4.8 Další módy aplikace

Kromě již zmíněných módů serveru a znovu vyhodnocení je možné aplikaci FMPCalculator spustit v dalších dvou módech pomocí různých argumentů příkazové řádky.

4.8.1 Vykreslování histogramů

Při spuštění aplikace s argumentem příkazové řádky *-p* nebo *–plot* bude výstupem aplikace histogram FMP skóre entit v databázi. V tomto módu je využito známé a volně dostupné knihovny matplotlib pro vykreslení histogramu. Tento mód aplikace byl využit pro vytvoření histogramů popisovaných v kapitole 4.7.1.

4.8.2 Publikování útočníků

Posledním módem ve kterém je možné aplikaci spustit je publikování útočníků do souborů. Pro publikování útočníků se aplikace spouští s parametry příkazové řádky *-u LOW MID HIGH* nebo *–publish LOW MID HIGH*. Po spuštění jsou vytvořeny tři textové soubory *low.txt*, *medium.txt* a *high.txt*. Do souboru *low.txt* jsou uloženy všechny IP adresy, které mají FMP skóre s takové že $LOW \leq s < MID$. Do souboru *medium.txt* jsou uloženy všechny IP adresy, které mají FMP skóre s takové že $MID \leq s < HIGH$. Do souboru *high.txt* jsou uloženy všechny IP adresy, které mají FMP skóre s takové že $HIGH \leq s$.

4.9 Konfigurace

Aplikace dovoluje konfigurovat časová okna a normalizaci dat. Časová okna se nastavují pomocí příkazové řádky. Normalizace dat je konfigurována pomocí souboru hodnot oddělených čárkami anglicky Comma-separated values zkráceně tedy CSV.

4.9.1 Časová okna

Nastavit krátkodobé časové okno lze pomocí parametru příkazové řádky *-s HOURS* nebo *–short-window HOURS*, kde *HOURS* je počet hodin krátkodobého časového okna. Pokud tento parametr není při spuštění vyplněn je standardně použita hodnota 24. Nastavit dlouhodobé časové okno lze pomocí parametrů *-l HOURS* nebo *–long-window HOURS*, kde *HOURS* je počet hodin dlouhodobého okna. Při nezadání parametru dlouhodobého okna bude použita hodnota 168.

4.9.2 Normalizace

V konfiguračním souboru *MaxParametrValues.csv* umístěném ve složce *Configuration* je možné nastavit jak se budou události normalizovat. První dva sloupce v souboru, *Typ* a *Podtyp*, slouží k identifikaci druhů událostí pro které chceme nastavení použít. Sloupec *Parametr* slouží k definici parametru detailu události ze kterého budeme odvozovat rozsah události. Například detail události typu *DICTATTACK* podtypu *SSHProtocol* vypadá následovně: *SSH dictionary attack, attempts: 278, port(s): 22, attack duration: 1 h 8 min 37 s 989 ms, average time between attempts: 14 s 812 ms.*. V příkladném detailu máme parametry *attempts*, *attack duration*, *average time between attempts*. Do sloupce *Maximální referenční hodnota* se zapisuje nejvyšší hodnota parametru, při dosažení či překročení této hodnoty bude rozsah roven 1 tj. maximální rozsah události. V případě nedosažení maximální hodnoty bude rozsah r vypočten jako $r = \frac{s}{m}$, kde m je nastavený maximální rozsah události

a s je skutečný rozsah události. Do sloupce *váha* je vloženo číslo od 0 do 1. Závažnost události se získá jako $rozsah \times váha$ události. Pokud nechceme nějaký typ a podtyp události vyhodnocovat stačí tedy nastavit váhu události v tomto souboru na 0. Výhodou tohoto řešení je snadná nastavitelnost parametrů bez nutnosti editovat či znát zdrojový kód.

Závěr

Cílem bakalářské práce bylo vytvořit systém pro sdílení informací mezi zákazníky Flowmon Networks a.s. o škodlivých událostech vzniklých v síti za účelem vzájemného posílení síťové bezpečnosti. Byli vytvořeny dvě spolu komunikující aplikace. První aplikace odesílá hlášení o bezpečnostních událostech získaných z Flowmon ADS. Druhá aplikace nejprve tato hlášení přijme. Z důvodu veliké rozmanitosti různých hlášení je nutné data aplikací normalizovat za účelem zisku nějakého jednotného ukazatele závažnosti události, na základě čehož je již možné data vyhodnocovat. Některá data tedy tímto procesem odpadnout protože bude zjištěno, že nemají být vyhodnocována, příkladem může být třeba hlášení o použití VPN v síti. Zbylá data jsou vyhodnocena navrženým algoritmem binárních stromů. Během vyhodnocování jsou použity jak statistiky závislé na čase jako EWMA nebo čas od posledního útoku tak zároveň různé agregáty normalizovaných závažností událostí jako například průměrná závažnosti událostí. Po vyhodnocení je záznam uložen do databáze. Záznamy v databázi je možné publikovat do souborů. Soubory mohou být snadno distribuovány zákazníkům, kteří díky tomu mohou pokusy útočníků zablokovat či hlásit.

Hlavním přínosem této práce je tedy zlepšení ochrany zákazníků Flowmon Networks a.s. a v důsledku toho zlepšení bezpečnosti jejich sítě. Všechno škodlivé chování útočníka na jakéhokoliv zákazníka je centrálně ukládáno, vyhodnocováno a poté sdíleno mezi zákazníky. Pokud útočník zaútočí na jednoho zákazníka tak další zákazníci mají již o útočníkovi informace a mohou na tuto informaci adekvátně reagovat, například zablokováním veškeré potenciální komunikace s útočníkem.

Pokračování práce by mohlo přinést rychlejší zpracování událostí zejména přepsáním do výkonnějších programovacích jazyků například jazyka C. S většími testovacími daty by také bylo možné místo ruční implementace rozhodovacích stromů použít metody strojového učení.

Literatura

[1] BARTOŠ, Václav. *Reputace zdrojů škodlivého provozu* [online]. Brno, 2018 [cit. 2021-12-09]. Dostupné z: <`https://www.fit.vut.cz/study/phd-thesis-file/758/758.pdf`>. Disertační práce. VUT Brno. Vedoucí práce Doc. Ing. JAN KOŘENEK, Ph.D.

[2] G. C. M. Moura, R. Sadre and A. Pras *Bad neighborhoods on the internet* in IEEE Communications Magazine, vol. 52, no. 7, pp. 132-139, July 2014, doi: 10.1109/MCOM.2014.6852094.

[3] *BGP Ranking* [online]. [cit. 2021-12-09]. Dostupné z: <`https://bgpranking.circl.lu/`>

[4] BARTOS, Vaclav, et al. *Network entity characterization and attack prediction.* Future Generation Computer Systems, 2019, 97: 674-686.

[5] COOKE, Evan; JAHANIAN, Farnam; MCPHERSON, Danny. *Zombie Roundup: Understanding, Detecting, and Disrupting Botnets.* SRUTI, 2005, 5: 6-6.

[6] ZHANG, Jian; PORRAS, Phillip A.; ULLRICH, Johannes. *Highly Predictive Blacklisting.* In: USENIX Security Symposium. 2008. p. 107-122.

[7] BARTOŠ, Václav. NERD: *Network Entity Reputation Database.* In: Proceedings of the 14th International Conference on Availability, Reliability and Security. 2019. p. 1-7.

[8] CESNET WARDEN [online]. [cit. 2021-12-09]. Dostupné z: <`https://www.cesnet.cz/sluzby/warden/`>

[9] MISP [online]. [cit. 2021-12-09]. Dostupné z: <`https://www.misp-project.org/`>

[10] ANDROULAKI, Elli, et al. *Reputation systems for anonymous networks. In: International Symposium on Privacy Enhancing Technologies Symposium.* Springer, Berlin, Heidelberg, 2008. p. 202-218.

[11] WAGNER, Cynthia, et al. *Misp: The design and implementation of a collaborative threat intelligence sharing platform.* In: Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security. 2016. p. 49-56.

[12] SAUERWEIN, Clemens, et al. *Threat intelligence sharing platforms: An exploratory study of software vendors and research perspectives.* 2017.

[13] ABU, Md Sahrom, et al. *Cyber threat intelligence–issue and challenges.* Indonesian Journal of Electrical Engineering and Computer Science, 2018, 10.1: 371-379.

[14] Uživatelská příručka 11.3 - Anomaly Detection System [online]. Flowmon Networks: ©2021 [cit. 16.9.2021]. Dostupné z: <`https://demo.flowmon.com/doc/adsplug/locale/cz/`>

[15] Flowmon [online]. [cit. 2021-12-09]. Dostupné z: <`https://www.flowmon.com/`>