VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ
DEPARTMENT OF COMPUTER SYSTEMS

REPUTACE ZDROJŮ ŠKODLIVÉHO PROVOZU

REPUTATION OF MALICIOUS TRAFFIC SOURCES

DISERTAČNÍ PRÁCE
PHD THESIS

AUTOR PRÁCE — **Ing. VÁCLAV BARTOŠ**
AUTHOR

ŠKOLITEL — **doc. Ing. JAN KOŘENEK, Ph.D.**
SUPERVISOR

BRNO 2018

Abstrakt

Při zajišťování bezpečnosti počítačových sítí je mimo jiné nezbytné získávat a zpracovávat informace o existujících hrozbách, ať už odvozené z hlášení vlastních detekčních nástrojů či pocházející od třetích stran. Mezi takové informace patří i seznamy síťových entit (IP adres, doménových jmen, URL apod.), které byly identifikovány jako škodlivé. V mnoha případech však prostá binární informace, zda je daná entita škodlivá či nikoliv, nestačí. Je vhodné mít ke každé entitě i další data popisující jí prováděné škodlivé aktivity a také shrnující skóre, které její reputaci vyjádří číselně. To umožní jednak rychlé zhodnocení míry hrozby, kterou určitá entita představuje, a zároveň umožní entity porovnávat a řadit. Tato práce se zabývá návrhem právě takového reputačního skóre. Navržené skóre, nazvané *Future Maliciousness Probability* (FMP skóre), je hodnota mezi 0 a 1 přiřazená každé podezřelé síťové entitě a vyjadřující pravděpodobnost, že bude daná entita v nejbližší době (znovu) provádět určitou škodlivou činnost. Výpočet tohoto skóre je tedy založen na předpovědi budoucích útoků. Tato předpověď vychází z historie přijatých hlášení o bezpečnostních událostech a z dalších relevantních dat týkajících se dané entity a je založena na pokročilých metodách strojového učení. Metoda výpočtu skóre je v práci nejprve popsána obecně, pro libovolný typ entity a vstupní data, a poté je přizpůsobena pro konkrétní případ – hodnocení IPv4 adres na základě hlášení ze systému pro sdílení bezpečnostních událostí a doplňujících dat z reputační databáze. Tato varianta pak byla vyhodnocena na reálných datech. Kvůli potřebě získat dostatečně velkou a kvalitní datovou sadu pro toto vyhodnocení se část práce věnuje i oblasti detekce bezpečnostních událostí, konkrétně vývoji frameworku pro analýzu dat o síťových tocích NEMEA a návrhu několika nových detekčních metod. Dále je popsán návrh a implementace otevřené reputační databáze NERD, která slouží k udržování profilů nahlášených IP adres. Data z těchto systémů pak byla využita jak pro vyhodnocení přesnosti predikce, tak pro vyhodnocení vybraných případů použití výsledného FMP skóre.

Klíčová slova

síťová bezpečnost, reputace, reputační skóre, reputační databáze, predikce útoků, strojové učení, analýza síťového provozu

Citace

BARTOŠ, Václav. *Reputace zdrojů škodlivého provozu*. Brno, 2018. Disertační práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Školitel doc. Ing. Jan Kořenek, Ph.D.

Abstract

An important part of maintaining network security is collecting and processing information about cyber threats, both from network operator's own detection tools and from third parties. A commonly used type of such information are lists of network entities (IP addresses, domains, URLs, etc.) which were identified as malicious. However, in many cases, the simple binary distinction between malicious and non-malicious entities is not sufficient. It is beneficial to keep other supplementary information for each entity, which describes its malicious activities, and also a summarizing score, which evaluates its reputation numerically. Such a score allows for quick comprehension of the level of threat the entity poses and allows to compare and sort entities. The goal of this work is to design a method for such summarization. The resulting score, called *Future Maliciousness Probability* (FMP score), is a value between 0 and 1, assigned to each suspicious network entity, expressing the probability that the entity will do some kind of malicious activity in a near future. Therefore, the scoring is based of prediction of future attacks. Advanced machine learning methods are used to perform the prediction. Their input is formed by previously received alerts about security events and other relevant data related to the entity. The method of computing the score is first described in a general way, usable for any kind of entity and input data. Then a more concrete version is presented for scoring IPv4 address by utilizing alerts from an alert sharing system and supplementary data from a reputation database. This variant is then evaluated on a real world dataset. In order to get enough amount and quality of data for this dataset, a part of the work is also dedicated to the area of security analysis of network data. A framework for analysis of flow data, NEMEA, and several new detection methods are designed and implemented. An open reputation database, NERD, is also implemented and described in this work. Data from these systems are then used to evaluate precision of the predictor as well as to evaluate selected use cases of the scoring method.

Keywords

network security, reputation, reputation score, reputation database, attack prediction, machine learning, network traffic analysis

Reputace zdrojů škodlivého provozu

Prohlášení

Prohlašuji, že jsem tuto disertační práci vypracoval samostatně pod vedením doc. Ing. Jana Kořenka, PhD. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Václav Bartoš
18. prosince 2018

Poděkování

Chtěl bych poděkovat svému školiteli doc. Ing. Janu Kořenkovi, PhD. a také Ing. Martinu Žádníkovi, PhD. za jejich odborné vedení a množství dobrých rad v průběhu celého doktorského studia. Dále děkuji zástupcům organizace CESNET za poskytnutí zázemí pro můj výzkum a za přístup k potřebným datům. Zvláštní dík patří členům skupiny věnující se analýze síťového provozu, vedené Ing. Tomášem Čejkou, PhD., v rámci níž byly implementovány a uvedeny do praxe myšlenky prezentované v této práci. V neposlední řadě pak děkuji své nejbližší rodině za podporu v průběhu celého studia.

Obsah

Kapitola 1

Úvod

Počítačové sítě a jejich prostřednictvím poskytované služby jsou dnes zcela běžnou součástí života většiny z nás. Je proto důležité zajistit nejen maximální spolehlivost těchto sítí a služeb, ale také jejich zabezpečení. Na všechny počítačové systémy připojené k internetu, od běžných domácích počítačů a mobilních telefonů přes servery velkých firem či veřejných institucí až po systémy řídicí kritickou infrastrukturu, totiž prakticky neustále míří nejrůznější kybernetické hrozby [1, 2, 3]. Těmi mohou být například různé podvodné emaily a phishing, útoky typu odepření služby (DDoS), šíření malware či pokusy o ovládnutí cizích systémů útočníky za různými účely, jako je změna obsahu webových stránek, odposlech citlivých dat, průmyslová i státní špionáž či sabotáž, často jsou také napadené systémy jen využity pro vykonávání dalších útoků. Stále častější jsou také masivní úniky osobních dat [4, 5]. V posledních letech se navíc objevují nové hrozby, jako například ransomware, tedy malware, jenž nějakým způsobem (obvykle zašifrováním dat) znepřístupní počítačový systém a vyžaduje zaplacení výkupného [6, 7, 8], nebo tzv. cryptojacking, tedy neoprávněné využívaní cizích výpočetních prostředků pro těžbu kryptoměn [9, 10].

Proti kybernetickým hrozbám lze bojovat řadou způsobů. Samozřejmostí jsou dnes pravidelné aktualizace software, antivirové programy, na úrovni správy sítí také firewally a jiná bezpečností zařízení, ale také např. vzdělávání uživatelů. Neméně důležité je však také monitorování síťového provozu a spolehlivá detekce probíhajících útoků a pokusů o ně. Za tímto účelem bývají nasazována různá zařízení typu IDS[1], systémy pro monitorování síťových toků, behaviorální analýzu, detekci anomálií a podobně. Užitečné informace o potenciálních hrozbách mohou poskytnout také tzv. honeypoty, které simulují napadnutelné zařízení a zaznamenávají činnost útočníka. Souhrnně tyto bezpečnostní monitorovací systémy poskytují informace o útocích směřovaných proti síti, ve které jsou nasazeny, případně i o již napadených zařízeních v této síti a jimi prováděných nežádoucích aktivitách.

Kromě monitorování vlastní sítě je vhodné získávat i informace o potenciálních hrozbách od třetích stran, tzv. kyberbezpečnostní informace (angl. *Cyber Threat Intelligence*, CTI). Těmi mohou být například seznamy různých indikátorů hrozeb (blacklisty), informace o objevených zranitelnostech, existujícím malware, technikách používaných známými skupinami hackerů, o aktuálních phishingových kampaních apod. Někdy jsou tato data vytvářena a spravována konkrétním subjektem a poskytována buď zdarma či v rámci placené služby, existují ale i různé platformy pro sdílení takových informací přímo mezi zapojenými orga-

[1] *Intrusion Detection System*, doslova „systém pro detekci průniků“, obecně jakýkoli systém pro detekci škodlivého provozu v počítačové síti

nizacemi, které pak data zároveň přijímají i vytvářejí. Sdílena jsou někdy i přímo strojová hlášení z detekčních systémů.

I přes rozvoj platforem pro sdílení informací na vyšší úrovni abstrakce, jsou podle nedávného průzkumu [11] stále nejčastěji používaným zdrojem kyberbezpečnostních informací běžné blacklisty (také nazývané seznamy indikátorů kompromitace, *Indicators of Compromise, IoC*), především seznamy škodlivých IP adres a URL.

Tyto blacklisty lze chápat jako jednoduchou formu hodnocení reputace z hlediska bezpečnostních hrozeb. Toto hodnocení je však jen binární – určitá entita (IP adresa, URL či jiný identifikátor) na seznamu buď je, nebo není, je tedy označena za škodlivou, nebo za neškodnou. Přitom skutečnost bývá složitější, různé adresy mohou představovat různou míru a různé druhy rizika a zdroje dat, ze kterých se vychází, mohou být různě spolehlivé. Žádná míra škodlivosti, spolehlivosti ani jiné doplňující informace (např. čas přidání na seznam, počet detekovaných pokusů o útok) však v blacklistech obvykle uváděny nejsou.

Blacklisty jsou vždy vytvářeny na základě určitých pravidel, například kolikrát musela být daná IP adresa nahlášena jako zdroj útoku, než je na seznam přidána, či po jak dlouhé době nečinnosti je ze seznamu odebrána. Problém je, že tato pravidla jsou určena poskytovatelem blacklistu a jeho uživatel na ně nemá žádný vliv (navíc v mnoha případech ani nejsou dostatečně dokumentována). Pokud je pak poskytován jen výsledný seznam bez dalších detailů, není ani možné vybrat si podmnožinu seznamu podle vlastních kritérií. Nelze si tak například zvolit, zda mají být zahrnuty i adresy jen mírně podezřelé, či jen ty, u nichž je velká jistota, že jsou škodlivé. Podobně v případě technických omezení na maximální délku aplikovaného blacklistu (např. při jeho využití pro blokování provozu) neexistuje způsob, jak z poskytnutého blacklistu vybrat pouze určitý počet adres tak, aby to byly ty, které představují největší hrozbu. Jinak řečeno, v případě tradičních blacklistů má uživatel jen dvě možnosti – použít celý blacklist tak, jak je, nebo ho nepoužít vůbec.

Částečné řešení nabízejí různé komerční databáze kyberbezpečnostních informací[2], které v jednom systému udržují nejrůznější typy dat o bezpečnostních hrozbách, včetně podrobných informací o škodlivých adresách, doménových jménech apod. Rozhraní těchto databází je však obvykle navrženo primárně pro vyhledání informací o jedné konkrétní entitě, možnost vytváření seznamů entit splňujících určitá kritéria obvykle chybí. Ani číselné hodnocení škodlivosti entit, které by umožnilo je porovnat a seřadit, není samozřejmostí a pokud už je nějaké skóre zobrazováno, není jasný jeho přesný význam ani způsob výpočtu.

V akademické sféře se hodnocení škodlivosti či reputace síťových entit v minulosti věnovalo několik prací [12, 13, 14]. Vždy však šlo jen o hodnocení celých skupin IP adres – síťových prefixů či čísel autonomních systémů. Takové hodnocení je založeno na pozorování, že adresy ve stejné síti se často chovají podobně, pokud je tedy v jedné síti detekováno několik škodlivých adres, je u ostatních adres v této síti vyšší pravděpodobnost, že jsou nebo brzy budou také škodlivé. Existující metody hodnotí škodlivost sítě pouze na základě počtu známých škodlivých adres v dané síti, není tedy možné aplikovat stejný princip i na hodnocení jednotlivých IP adres (resp. vznikla by opět jen binární informace, jako u běžných blacklistů). Hodnocení celých sítí má však zřejmou nevýhodu v tom, že všechny adresy v dané síti jsou hodnoceny stejně, ačkoliv některé mohou představovat větši hrozbu než jiné – přinejmenším v tom, že některé již byly skutečně detekovány jako škodlivé (a to potenciálně s různou intenzitou), u jiných se pouze předpokládá tato možnost na základě jejich příslušnosti do stejné sítě.

[2]Příklady těch, které umožňují alespoň částečný otevřený přístup, jsou AlienVault OTX, https://otx.alienvault.com/ či IBM X-Force Exchange, https://exchange.xforce.ibmcloud.com/

Jako řešení výše uvedených nedostatků současného stavu autor této práce navrhuje shromažďovat data o detekovaných bezpečnostních událostech v otevřené reputační databázi, a to tak, aby mohly být dále poskytovány nejen hotové seznamy entit, které jsou dle určitých pravidel vyhodnoceny jako škodlivé, ale i všechna další dostupná data o těchto entitách. Ta by uživatelům poskytovala detailní informace o potenciálních hrozbách a umožňovala každému vytvořit si seznam škodlivých entit dle vlastních kritérií. Velmi důležitou součástí systému poskytujícího tato data by měla být i možnost ohodnotit jednotlivé entity pomocí skóre, které by číselně vyjádřilo jejich reputaci či míru hrozby, kterou představují. To by jednak usnadnilo rychlé zhodnocení podezřelé entity člověkem analytikem, zároveň by umožnilo entity porovnávat či řadit. Takové seřazení pak v důsledku umožní i vytváření optimálních blacklistů uživatelem definované velikosti pouhým výběrem prvních n záznamů s nejhorší reputací.

Hlavním cílem této disertační práce je návrh metody výpočtu právě takového reputačního skóre. To by mělo být přiřazeno každé jednotlivé podezřelé entitě (např. IP adrese). Mělo by shrnovat informace jak o předchozím chování této entity, tak i o chování blízkých či jinak podobných entit (např. ostatních adres ve stejné síti). Tím se spojí výhody obou výše uvedených přístupů k hodnocení reputace – vysoká granularita blacklistů (které rozlišují jednotlivé entity, avšak bez skóre) a využití korelací mezi blízkými entitami, používané dosud pouze při hodnocení celých sítí. Kromě toho by měla metoda výpočtu reputačního skóre umožňovat zahrnout i případná další dostupná data relevantní pro vyhodnocení míry hrozby, kterou entita představuje.

Obecně platí, že ačkoliv hodnocení reputace vždy vychází z informací z minulosti, jeho účelem je především podpora rozhodování v současnosti a blízké budoucnosti. Aby tedy bylo skóre dobře použitelné v rámci obrany a prevence budoucích útoků, je dále vhodné, aby místo pouhého shrnutí známých informací o předchozím chování bylo založeno spíše na explicitní predikci chování dané entity v nejbližší budoucnosti. To sice zpravidla vychází z chování minulého, může být ale ovlivněno i řadou dalších faktorů.

Reputační skóre splňující výše uvedená kritéria lze využít řadou způsobů. Jak již bylo zmíněno, může sloužit pro rychlé zhodnocení škodlivosti entity člověkem či pro vytváření blacklistů s uživatelem volenými parametry, jako je limit na počet záznamů či mezní hodnota míry hrozby, kterou musí entita překročit, aby byla na seznam přidána. Skóre může být dále využito i jako vstup jiných algoritmů, například jako jedno z kritérií pro prioritizaci incidentů v SIEM systémech, v rámci algoritmů rozlišujících škodlivý provoz od legitimního, např. při ochraně proti spamu nebo DDoS útokům, nebo pro řízení míry detailu monitorování provozu jednotlivých entit (např. zachycení kompletního vzorku paketů či aktivace IDS pouze pro provoz určitých IP adres [15]).

Kromě návrhu metody hodnocení reputace síťových entit se část práce věnuje také oblasti detekce bezpečnostních hrozeb. Cílem je především získání dostatečného množství dat pro hlavní část této práce, neboť hlášení o detekovaných škodlivých aktivitách jsou hlavním vstupem pro určení reputačního skóre, nově vyvinuté detekční metody a framework, v němž byly implementovány, jsou však zároveň samy o sobě významným přínosem ve své oblasti.

1.1 Přínosy práce

Jádrem a hlavním přínosem této disertační práce je:

- Návrh obecné metody hodnocení reputace síťových entit na základě předpovědi jejich budoucího chování (kap. 4).
- Ověření navržené metody nad reálnými daty o škodlivých IP adresách (kap. 8 a 9).

Aby mohlo být dosaženo dobrých výsledků v tomto hlavním tématu práce, bylo nutné získat velké množství kvalitních dat a analyzovat je. To vedlo k několika vedlejším přínosům této práce:

- Návrh systému pro analýzu síťového provozu (NEMEA) a několika nových metod pro detekci škodlivého síťového provozu na základě analýzy dat o síťových tocích (kap. 5).
- Návrh a implementace reputační databáze síťových entit (kap. 6).
- Analýza charakteristik bezpečnostních hlášení a zdrojů škodlivého provozu (kap. 7).

Podíl autora

Veškeré zde popisované výsledky, s výjimkou systému NEMEA, jsou dílem autora této práce, buď na nich pracoval sám, nebo byl po celou dobu vedoucím prací.

Systém NEMEA a související detekční metody jsou výsledkem práce většího týmu lidí, jehož byl autor této práce součástí. Konkrétně byl autor jedním ze dvou lidí, kteří navrhli koncept systému NEMEA, vytvořili první prototyp a implementovali řadu základních modulů. Dále autor pracoval na vývoji několika nových detekčních metod a přípravě souvisejících publikací.

1.2 Struktura práce

Následující kapitola obsahuje stručný úvod do problematiky bezpečnostního monitorování sítí a zpracování hlášení o detekovaných událostech. Kapitola 3 pak shrnuje existující literaturu týkající se ohodnocování škodlivosti síťových entit a predikce jejich chování. V kapitole 4 je popsána hlavní myšlenka práce, metoda výpočtu reputačního skóre v obecné podobě, tj. bez ohledu na typ entity, a je zde uvedena její formální definice. Další části práce se pak zabývají aplikací této metody na konkrétním typu dat a souvisejícími implementačními činnostmi. Nejprve je v kapitole 5 popsán přínos v oblasti analýzy síťového provozu a detekce útoků, především je kapitola věnována systému NEMEA. V kapitole 6 je pak představen systém reputační databáze NERD. Následuje popis a analýza dat použitých v této práci (kapitola 7). Na základě kontextu a analýzy dat z předchozích kapitol je pak v kapitole 8 obecná metoda výpočtu FMP skóre konkretizována pro případ IPv4 adres a dat ze systémů Warden a NERD. Vyhodnocení vlastností této metody podle různých kritérií a ukázky jejího možného využití v praxi jsou uvedeny v kapitole 9. Celou práci pak shrnuje a uzavírá kapitola 10.

Kapitola 2

Úvod do problematiky

Tato kapitola obsahuje stručný úvod do problematiky bezpečnostního monitorování sítě a správy informací o kybernetických hrozbách. Zaměřena je zejména na metody a přístupy související s dalšími částmi této práce, konkrétně je popsáno monitorování sítě pomocí záznamů o IP tocích, vybrané metody detekce bezpečnostních událostí a základní přístupy a existující systémy pro sdílení a zpracování informací o detekovaných událostech a hrozbách.

2.1 Monitorování síťového provozu

Způsoby monitorování síťového provozu lze rozdělit na několik kategorií podle úrovně detailu – jednoduché statistiky o objemu provozu sbírané např. pomocí protokolu SNMP (*Simple Network Management Protocol*), data o jednotlivých síťových spojeních, tzv. IP tocích či *flow*, a detailní analýza kompletního obsahu paketů.

Nejjednodušší způsob, tedy sběr dat pomocí SNMP, umožňuje získávat ze směrovačů a jiných síťových prvků základní statistiky o jimi zpracovaném provozu. Tento protokol je podporován většinou i velmi jednoduchých síťových zařízení a je tak snadné získat informace o všech částech sítě. Tyto informace jsou však velmi agregované, obvykle jsou poskytovány jen informace o celkovém objemu přenesených dat na daném rozhraní. To sice stačí pro základní přehled o zatížení sítě, pro řešení případných problémů to však nemusí být dostačující a jakákoliv bezpečnostní analýza je prakticky vyloučena.

Opačným extrémem je zachytávání všech paketů procházejících sítí. V tomto případě nedochází k žádné agregaci a jsou získány kompletní informace o síťovém provozu. Tento princip monitorování je využíván u klasických IDS systémů založených na detekci vzorů. Tyto systémy obvykle procházejí obsah každého paketu a vyhledávají v nich předdefinované řetězce či regulární výrazy popisující určité útoky, malware apod. (podrobněji viz kap. 2.2.2) Vyhledávání však probíhá v reálném čase, on-line, a data nejsou ukládána. Archivace veškerého síťového provozu, např. pro pozdější podrobnější analýzu či pro zpětné dohledání důkazů o nějakém incidentu, je vzhledem k množství dat přenášených po dnešních sítích prakticky nemožná. Navíc by takto detailní monitorování, při kterém by byl ukládán i obsah komunikace, bylo velmi problematické i z legislativního hlediska kvůli ochraně osobních dat.

Vhodným kompromisem z pohledu poměru množství užitečných informací v datech ku jejich velikosti je měření tzv. IP toků [16]. Tato metoda je dnes velmi rozšířená a je často používána i pro analýzu provozu za účelem detekce bezpečnostních hrozeb. Na měření a ana-

lýzu IP toků je zaměřena i část této práce (kap. 5), proto je zde tento způsob monitorování popsán podrobněji.

2.1.1 Monitorování síťových toků

IP tok (angl. *flow*) je obvykle definován jako množina paketů pozorovaných na určitém místě v síti, které mají stejnou zdrojovou a cílovou adresu, zdrojový a cílový port, číslo protokolu, síťové rozhraní, na kterém byl paket zachycen, a typ služby (položka *Type of Service* v IP hlavičce). Někdy se používá jen prvních pět položek.

Ke každému IP toku jsou obvykle měřeny základní statistiky o spojení, jako např. kolik paketů a kolik bytů bylo celkem přeneseno, kdy spojení začalo a jak dlouho trvalo, či v případě TCP spojení jaké příznaky byly v průběhu komunikace nastaveny.

Obvykle se rozlišuje směr komunikace, kdy jedno obousměrné spojení sestává ze dvou toků, jeden pro každý směr. Některá zařízení ale dokáží oba směry spárovat a statistiky z obou směrů počítat v jednom záznamu, tzv. *bi-flow*.

Informace o tocích jsou měřeny tzv. *flow exportéry*, které je ve formě záznamů odesílají na *kolektor*. Tam jsou data ukládána a mohou být různě analyzována. Kolektor je obvykle jeden, exportérů bývá více a jsou umístěny na různých místech v síti. Tradičně plní úlohu exportérů routery, které mají měření toků implementováno jako svou sekundární funkci, prosazují se však i samostatné sondy – zařízení navržená speciálně pro tuto úlohu. Některé sondy jsou navíc programovatelné a lze je snadno upravit podle aktuálních potřeb měření.

Data o IP tocích jsou z exportéru na kolektor obvykle přenášena protokolem NetFlow (nejčastěji verze 5 a 9), proprietárním protokolem společnosti Cisco, který se stal *de-facto* standardem pro sběr tohoto typu dat, případně jeho alternativami od jiných výrobců (např. J-Flow společnosti Juniper). Čím dál častěji je také podporován a používán protokol IPFIX [17, 18], otevřený standard IETF, který vychází z NetFlow v9 a dále rozšiřuje jeho flexibilitu co se týče možností přenášet různé typy dat.

Více informací o běžných způsobech měření IP toků, architektuře a protokolech lze nalézt v [16].

2.1.2 Záznamy rozšířené o data z aplikační vrstvy

Tradiční monitorování IP toků poskytuje informace o provozu na síťové a transportní vrstvě (L3 a L4 ISO/OSI modelu). Tato úroveň detailu sice pro mnoho aplikací, včetně některých bezpečnostních analýz, stačí (viz kap. 2.2.1), přesto je však někdy vhodné mít k dispozici i vybrané informace z aplikační vrstvy (L7). V posledních letech se proto rozvíjí rozšířená varianta monitorování toků, při které jsou na exportéru analyzovány i hlavičky vybraných aplikačních protokolů a informace o nich jsou přidávány do záznamů o tocích [19, 20, 21, 22, 23, 24]. Záznamy jsou tak obohacovány o data, jako např. *URL*, *User-Agent*, návratový kód či *Content-Type* z HTTP provozu, doménová jména a IP adresy z DNS provozu, emailové adresy z protokolu SMTP apod. Tato data pak poskytují nové možnosti analýzy provozu, včetně možnosti detekovat bezpečností hrozby, které by v tradičních datech o tocích nebylo možné rozpoznat. Pro tento způsob rozšířeného monitorování zatím není zcela ustálený název, nejčastěji se však lze setkat s pojmem *application-aware flow monitoring* [20, 22, 16], někdy také *(L7-)extended flow monitoring* [25, 23, 26].

2.2 Detekce bezpečnostních hrozeb

Hlavním důvodem pro nasazení monitorovacích technologií je často možnost takto získaná data automatizovaně analyzovat za účelem detekce různých bezpečnostních hrozeb a útoků. Tato podkapitola stručně shrnuje různé přístupy k takové detekci.

Pro rozpoznání velmi výrazných anomálií, např. DDoS útoků, často stačí základní statistiky o objemu provozu, získané např. pomocí SNMP monitorování. I v případě úspěšné detekce však tato data obvykle neposkytují dostatek detailů o charakteru útoku a není je tak možné využít pro efektivní obranu. Pro detekci bezpečnostních hrozeb se tedy zpravidla využívají data o síťových tocích, případně analýza obsahu paketů v klasických IDS systémech. Tyto metody jsou popsány v následujících podkapitolách. Kapitola 2.2.3 pak představuje jeden z dalších způsobů získávání informací o bezpečnostních hrozbách, a to honeypoty.

2.2.1 Analýza dat o síťových tocích

Data o síťových tocích lze využívat pro detekci mnoha různých typů útoků a jiného nežádoucího provozu. Tento přístup je hojně využíván existujícími komerčními i open-source systémy pro monitorování sítě. Zároveň jsou metody analýzy toků předmětem mnoha akademických prací (např. [27, 28, 29, 30] a další níže).

Detekce založená na analýze síťových toků zpravidla vychází z toho, že mnohé nežádoucí aktivity, jako například skenování portů, DDoS útoky nebo automatizované hádání hesel, se projevují specifickými charakteristikami provozu, které lze v datech o síťových tocích rozpoznat. Jiné metody analýzy pak spočívají spíše ve výpočtu různých statistik charakterizujících provoz, naučení se běžného stavu a následné detekci anomálií.

Protože tradiční záznamy o IP tocích obsahují data pouze ze síťové a transportní vrstvy, jsou obecně nejsnáze rozpoznatelné útoky probíhající právě na těchto vrstvách, tedy především volumetrické DDoS útoky a skenování portů. Konkrétní práce zaměřené na detekci tohoto typu útoku jsou např. [31, 32, 30, 33].

Lze ale detekovat i útoky využívající specifické aplikační protokoly. Příkladem je detekce DNS amplifikačních DDoS útoků na základě analýzy množství a velikostí toků směřujících na a z DNS serverů [34].

V některých případech pak mohou tradiční záznamy o IP tocích poskytnout dostatek informací i pro detekci útoků probíhajících přímo na aplikační vrstvě. Příkladem jsou metody detekce slovníkových útoků na SSH popsané v [35] a [36]. Obě metody jsou velmi podobné a využívají toho, že tyto útoky se projevují velkým množstvím spojení mezi dvojicí IP adres, kdy každé spojení (a tedy i odpovídající dvojice záznamů o IP tocích) má počet paketů a bytů v určitém úzkém rozsahu. Obdobnou metodu lze použít pro detekci slovníkových útoků i na jiné protokoly, např. HTTP(S) [37].

Data rozšířená o položky z aplikační vrstvy pak samozřejmě umožňují detekovat další typy nežádoucího chování na této vrstvě. Příkladem je analýza HTTP požadavků [26], která umožňuje na základě rozšířených záznamů o IP tocích detekovat události jako webové skenování[1], přístupy přes webové proxy servery (a tím např. odhalení nepovolených proxy serverů uvnitř vlastní sítě), web crawlery, nebo opět i slovníkové útoky – jejich detekce pomocí dat z aplikační vrstvy je však spolehlivější a poskytuje více detailů, například

[1] Vyhledávání konkrétních URL, obvykle k administračním rozhraním známých CMS systémů, jako je Wordpres či Joomla!, na různých serverech.

konkrétní URL, na kterém jsou hesla hádána. Rozšířená data o IP tocích využívají i některé detektory v rámci systému NEMEA, popisované dále v kapitole 5.3.2.

Mezi práce zabývající se detekcí obecných anomálií v síťovém provozu na základě dat o IP tocích patří například [38, 39, 40, 41]. Tyto metody se zpravidla nezaměřují na konkrétní typ útoku, ale jsou schopné detekovat různé neobvyklé události, a to nejen bezpečnostního charakteru. Kromě DDoS útoků, intenzivních skenování, či šíření síťových červů tedy může jít například i o neobvykle velké datové přenosy či výpadky části sítě.

V praxi, v komerčním prostředí, se metody založené na analýze IP toků obvykle označují jako *Network Behavior Analysis* (NBA) (význam tohoto pojmu však není zcela jednoznačný a někdy jsou pod něj zahrnovány i jiné přístupy). Mezi komerční řešení schopná detekovat síťové útoky na základě analýzy dat o IP tocích patří například FlowMon ADS[2], FlowTrack[3], Cisco Stealthwatch[4] (dříve Lancope) či Arbor Sightline[5].

2.2.2 IDS systémy

Intrusion Detection Systems (IDS, systémy pro detekci průniků) jsou obecně jakékoliv systémy pro detekci kybernetických bezpečnostních hrozeb. V kontextu monitorování síťového provozu se tímto pojmem tradičně myslí systémy analyzující jednotlivé pakety a detekující útoky na základě databáze známých vzorů (tzv. signatur). Přístup, při kterém je detailně analyzován obsah paketů, je také nazýván *Deep Packet Inspection* (DPI).

Tyto IDS systémy lze rozdělit podle toho, zda jsou zaměřené převážně na vyhledávání určitých řetězců či regulárních výrazů v obsahu jednotlivých paketů, nebo s pakety pracují jako s tzv. událostmi a umožňují analyzovat i posloupnosti událostí a různé vztahy mezi nimi [42]. Výhodou prvního přístupu, vyhledávání regulárních výrazů, je jeho jednoduchost a z toho vyplývající rychlost zpracování. Typickými zástupci tohoto přístupu jsou systémy Snort[6] [43] a Suricata[7]. Analýza založená na událostech je sice obvykle výpočetně náročnější, ale umožňuje detekovat i složitější útoky sestávající např. z posloupnosti několika kroků. Typickým zástupcem tohoto přístupu je systém Bro[8] [44]. Celkově je výhodou všech IDS systémů založených na databázi vzorů poměrně vysoká spolehlivost detekce známých typů útoků či známého malware, nevýhodou je pak nemožnost detekovat útoky nové, pro které není v použité databázi odpovídající vzor.

Alternativou jsou systémy založené na detekci anomálií. Při té se IDS systém učí normální charakteristiky provozu a hlásí všechny podstatné odchylky od tohoto normálu – anomálie. Ty mohou, ale také nemusí, znamenat narušení bezpečnosti. Zásadní výhodou tohoto přístupu oproti databázi vzorů je schopnost detekovat i dosud neznámé útoky, nevýhodou je pak větší množství falešných poplachů. Pro detekci anomálií se dříve používaly různé statistické metody, dnes jsou stále častěji používány pokročilé metody strojového učení [45]. Tento přístup je dnes nejčastěji aplikován jako součást různých komerčních bezpečnostních řešení. Pravděpodobně jediným open-source IDS systémem pracujícím na principu detekce anomálií je Hogzilla IDS[9].

[2] https://www.flowmon.com/en/products/flowmon/anomaly-detection-system
[3] https://www.flowtraq.com/
[4] https://www.cisco.com/c/en/us/products/security/stealthwatch/index.html
[5] https://www.netscout.com/product/arbor-sightline
[6] https://www.snort.org/
[7] https://suricata-ids.org/
[8] https://www.bro.org/
[9] http://ids-hogzilla.org/

2.2.3 Honeypoty

Honeypoty jsou jedním z dalších možných zdrojů dat o síťových bezpečnostních hrozbách. Honeypot je definován jako počítačový systém, jehož účelem je být skenován, napadán a kompromitován útočníky [46, 47]. Honeypot zpravidla simuluje nějaký reálný systém či službu a detailně zaznamenává jakoukoliv interakci s ním. Cílem je nalákat útočníka, aby zaútočil na honeypot místo reálného systému, a pomocí záznamu jeho aktivit pak odhalit používané techniky či například získat vzorky škodlivého kódu.

Honeypoty se rozdělují do tří kategorií podle míry interakce s útočníkem – nízké, střední a vysoké [48]. Honeypoty s nízkou mírou interakce (*low interaction honeypots*) simulují jen základní vlastnosti určité služby (např. SSH, HTTP, databázové servery). Jsou schopny odpovědět na některé požadavky útočníka, např. vrací statické webové stránky v případě simulace HTTP serveru či umožňují přihlášení a simulují některé jednoduché příkazy v případě SSH připojení, ale nejde o reálný systém a množina simulovaných funkcí bývá značně omezená. Výhodou jsou malé nároky takového systému na výkon i na údržbu, nevýhodou je omezené množství informací o metodách útočníka, které lze tímto způsobem získat. Navíc pro útočníka nebývá těžké zjistit, že komunikuje s honeypotem a ne se skutečnou službou.

Naproti tomu honeypoty s vysokou mírou interakce (*high interaction honeypots*) se snaží být co nejvíce realistické, jsou tedy obvykle založeny na serveru s běžným operačním systémem a skutečnými službami, pouze je systém upravený pro podrobné zaznamenávání aktivit útočníka. Výhodou takového přístupu je možnost získat mnohem více informací o aktivitách útočníka, pro kterého je navíc jen velmi obtížné zjistit, že komunikuje s honeypotem. Nevýhodou však je, že pokud honeypot dovoluje provádět jakékoliv operace stejně jako na skutečném serveru, je obtížné zabránit ve zneužití samotného honeypotu, např. pro provádění dalších útoků z něj [46].

Jako honeypoty se střední mírou interakce (*medium interaction honeypots*) se pak zpravidla označují ty, které sice neobsahují skutečný operační systém a službu pouze simulují, tato simulace je však velmi detailní a důvěryhodná.

Pomocí honeypotů lze získat různé typy dat o bezpečnostních hrozbách. Může jít jen o jednoduchá hlášení o pokusech o přístup k honeypotu či zneužití nějaké zranitelnosti, seznam zkoušených hesel při slovníkových útocích, ale třeba i vzorky malware stažené útočníkem do „napadeného“ zařízení nebo kompletní záznamy požadavků či příkazů útočníka.

Hlášení o útočících IP adresách pak mohou být využita např. v IDS/IPS systémech[10] pro automatické blokování škodlivých adres [48], či jinak zpracována, podobně jako hlášení z klasických IDS systémů či z analýzy dat o IP tocích (o zpracování pojednávají další podkapitoly). Podrobná analýza zachyceného malware či záznamů aktivit na honeypotu pak může odhalit techniky používané útočníky, včetně zcela nových útočných vektorů [47]. I přes možnost částečné automatizace takové analýzy však objevení skutečně nových a zajímavých informací obvykle vyžaduje značné množství práce zkušeného experta.

2.3 Sdílení a zpracování kyberbezpečnostních informací

Při zajišťování bezpečnosti sítě a vyhodnocování rizik jsou využívány různé typy informací z různých zdrojů. Může jít o hlášení o detekovaných událostech z vlastních monitorovacích nástrojů (viz předchozí podkapitoly), o hlášení získaná z jiných sítí v rámci sdílení informací, různé seznamy a databáze škodlivých IP adres či domén, informace o zranitelnostech

[10] *Intrusion Detection and Prevention System* – systém schopný nejen detekovat probíhající pokus o útok, ale pomocí filtrování provozu mu i zabránit.

software, probíhajících kampaních a různém malware, nebo třeba souhrnné výroční zprávy shrnující nejvážnější hrozby v určité oblasti. Tyto informace mohou být dále korelovány, agregovány, obohacovány o další související data apod. Výsledné informace o hrozbách či incidentech je pak často vhodné prioritizovat dle důležitosti.

Následující podkapitoly se stručně věnují vybraným aspektům získávání a zpracování těchto informací, zejména těm souvisejícím s dalšími částmi této práce.

2.3.1 Klasifikace kyberbezpečnostních informací

Existuje široká škála druhů bezpečnostních informací a v praxi používaná terminologie je značně nepřehledná. Souhrnně se informace o kybernetických bezpečnostních hrozbách obvykle nazývají *Cyber Threat Intelligence*[11] (CTI). Stejný pojem ale může označovat i proces získávání takových informací. Navíc jsou pod pojmem CTI často chápána pouze pečlivě zpracovaná a kontextualizovaná data, určená pro zpracování člověkem a sloužící pro podporuju taktických či strategických rozhodování, někdy se ale jako CTI označují i více surová či nízkoúrovňová data typu hlášení z IDS systémů nebo seznamy škodlivých IP adres či domén.

Existuje však několik klasifikací typů kyberbezpečnostních dat, které umožňují se v problematice lépe orientovat. Základní, často používané dělení poskytuje zpráva [49] od Evropské agentury pro bezpečnost sítí a informací (ENISA):

- Nízkoúrovňová data (*low-level*) – Záznamy paketů či síťových toků, aplikační logy, záznamy z IDS, vzorky souborů či emailových zpráv.
- Indikátory pro detekci (*detection indicators*) – IP adresy, doménová jména, hashe souborů a další identifikátory spojené s konkrétními hrozbami či útoky.
- Poradní zprávy, výstrahy (*advisories*) – Informace o objevených zranitelnostech, vydaných opravách, novém malware, či obecně o konkrétních technikách používaných útočníky.
- Strategické zprávy (*strategic reports*) – Shrnutí výsledků analýz, sepsané jako souvislý text, jehož cílem je poskytnout přehled o konkrétních situacích či existujících hrozbách.

Jiné, podrobnější dělení nabízí např. dokument společnosti Microsoft [50], v němž je rozlišováno 7 typů informací: indikátory, hrozby, zranitelnosti, obranná opatření, situační přehled, doporučené postupy a strategické analýzy. Tento dokument patří mezi ty, které uvažují jen informace na vyšších úrovních, a surová nízkoúrovňová data tak v klasifikaci vůbec nejsou.

Za další dělení typů bezpečnostních informací lze považovat i datový model formátu STIX[12], který obsahuje objekty pro různé typy informací od velmi nízkoúrovňových dat, jako jsou síťové toky či IP adresy, až po vysokoúrovňové informace, jako jsou popisy konkrétních útočníků či kampaní.

Tato práce je zaměřena na zpracování dat na nižších úrovních abstrakce, tedy „nízkoúrovňových dat“ a „indiktátorů pro detekci“ dle klasifikace ENISA. Konkrétně jde především o strojově generovaná, nízkoúrovňová hlášení o detekovaných bezpečnostních událostech, jako jsou např. hlášení z IDS systémů či honeypotů. Jsou to jednoduché strukturované

[11] Lze přeložit jako „zpravodajství kybernetických hrozeb“, žádný ustálený český ekvivalent však neexistuje a i v českých textech se obvykle používá původní anglické *Cyber Threat Intelligence*.

[12] https://oasis-open.github.io/cti-documentation/stix/intro

zprávy nesoucí informaci o konkrétní škodlivé či podezřelé aktivitě mezi konkrétními zdroji a cíli v určitý čas. V dalším textu jsou tato data označována jako „hlášení o bezpečnostních událostech“ či jednoduše jen „hlášení“.

2.3.2 Sdílení

Kyberbezpečnostní informace lze získávat z vlastních bezpečnostních monitorovacích nástrojů, případně jinými nástroji a analýzami ve vlastní síti (např. analýzou zachyceného malware, vyhledáváním zranitelností apod.). Je však velmi vhodné tato data doplňovat o informace z dalších zdrojů.

Těmi jsou často data poskytovaná komerčními firmami, buď v rámci širších kyberbezpečnostních řešení, někdy i jako samostatná služba. V posledních letech se také stále více rozvíjejí různé platformy pro sdílení kyberbezpečnostních informací přímo mezi organizacemi. Některé takové platformy jsou provozovány velkými firmami, jiné vznikly jako nezávislé otevřené projekty.

Sdílení kyberbezpečnostních dat je také aktivní oblastí výzkumu a vývoje, což dokazují mimo jiné různé projekty zabývající se touto problematikou, jako např. aktuálně řešený evropský projekt PROTECTIVE[13] nebo český projekt SABU[14]. Tématu se věnuje také řada odborných publikací.

Například v pracech [51, 52], z let 2013 a 2014, autoři formálně navrhli koncept platformy pro sdílení kyberbezpečnostních dat a stanovili základní požadavky, které by takové platformy měly splňovat.

V práci [53] autoři analyzují, do jaké míry mohou informace o škodlivých IP adresách sdílené a agregované v rámci sdílecích platforem pomoci v boji proti různým typům škodlivých aktivit a kde jsou limity tohoto přístupu.

Další práce se pak často zaměřují spíš na mapování aktuálního stavu. Například zpráva [54] od agentury ENISA poskytuje detailní přehled o existujících standardech, formátech a nástrojích pro sdílení a zpracování kyberbezpečnostních informací (pochází však z roku 2014 a neobsahuje tak všechny dnes používané nástroje). Shrnutí datových formátů a protokolů uvádí také práce [55]. Poměrně aktuální seznam existujících sdílecích platforem a jejich vlastností je pak uveden v [56].

Práce [57] mapuje stav nasazení různých metod sdílení dat mezi poskytovateli internetového připojení a jak tito poskytovatelé využívají sdílená data pro obranu proti síťovým útokům.

Další oblastí, kterou je třeba se při sdílení kyberbezpečnostních dat zabývat, jsou legislativní aspekty v souvislosti s ochranou osobních údajů. Touto problematikou se zabývají například články [58, 59].

Většina v současnosti používaných sdílecích platforem začala vznikat především mezi lety 2012–2015 [53]. Mezi významné platformy provozované komerčními firmami patří například AlienVault Open Threat Exchange[15] (AV OTX), IBM X-Force Exchange[16], či Facebook ThreatExchange[17].

Příklady nezávislých, open-source platforem jsou MISP, vyvinutý organizací CIRCL[18], či Warden od organizace CESNET. Lze sem zařadit i platformu DShield.

[13] https://protective-h2020.eu/
[14] https://sabu.cesnet.cz/
[15] https://otx.alienvault.com
[16] https://exchange.xforce.ibmcloud.com/
[17] https://developers.facebook.com/programs/threatexchange/
[18] Computer Incident Response Center Luxembourg, https://www.circl.lu/

MISP[19,20] [60] je v současnosti velmi rozšířená platforma pro sdílení informací o kybernetických hrozbách. Podle oficiálních webových stránek ji v roce 2018 používá asi 6000 organizací. V systému MISP lze ukládat, korelovat a hlavně sdílet různé informace, především tzv. indikátory, jako jsou například doménová jména a IP adresy představující hrozbu, kontrolní součty (tzv. *hashe*) škodlivých souborů, ale třeba i telefonní čísla nebo čísla bankovních účtů spojená s kybernetickou kriminalitou. Data jsou strukturovaná a indikátory jsou vždy uvedeny jako součást určité „události“, ta může popisovat konkrétní incident, kampaň, vzorek malware, skupinu útočníků apod. Pro další obohacení informací lze ke každému datovému elementu přiřadit různé příznaky. Silnou stránkou platformy MISP je také pokročilý model sdílení informací. Existují stovky instancí systému, provozované různými organizacemi, které mohou vzájemně sdílet vybrané informace v modelu peer-to-peer. U každé události lze navíc podrobně specifikovat, jakým způsobem může být šířena dál, např. zda má zůstat jen v rámci dané instance, může se sdílet k dalším přímo připojeným instancím, či se může šířit kamkoliv v rámci celé peer-to-peer sítě.

Warden[21] [61, 62, 63] je systém pro sdílení bezpečnostních hlášení mezi zapojenými organizacemi. Hlavní instance je provozována sdružením CESNET, jde však o open-source software, který může nasadit i kdokoliv jiný. Sdílení probíhá přes jeden centrální server, na který tzv. odesílající klienti posílají data – hlášení z různých monitorovacích a detekčních nástrojů, jako jsou honeypoty, IDS, systémy pro analýzu flow dat apod. Server pak všechna tato hlášení poskytuje tzv. přijímacím klientům, kteří si je v pravidelných intervalech stahují a mohou je dále jakýkoliv způsobem zpracovávat – filtrovat, agregovat, vytvářet hlášení pro uživatele, počítat statistiky apod. Všechna hlášení jsou v systému Warden sdílena ve formátu IDEA[22] [64]. Tento formát je navržen speciálně pro reprezentaci nízkoúrovňových bezpečnostních hlášení. Data jsou strukturována, základem je formát JSON. Je definována množina položek s jasně daným významem, například kategorie události, čas začátku a konce události, čas detekce, zdrojová a cílová IP adresa, port a protokol, informace o detektoru apod. Většina položek je nepovinná a je možné vložit i jiné, nestandardní, položky, díky čemuž je formát velmi flexibilní a rozšiřitelný. Zároveň jsou však základní data vždy na jasně definovaných místech v záznamu, což umožňuje efektivní strojové zpracování.

DShield[23] je platforma určená pro sběr a analýzu hlášení o detekovaných příchozích škodlivých aktivitách od tisíců přispěvatelů. Jde o jednoduché zprávy z paketových filtrů, jako jsou firewally a IDS systémy, obsahující čas události, IP adresu a port zdroje a cíle a počet pokusů o spojení. Prakticky kdokoliv může výsledky svých detekčních systémů odesílat na servery DShield, kde jsou data agregována a analyzována. Výsledkem jsou pak nejrůznější statistiky dostupné na webových stránkách projektu a také blacklisty nejčastěji útočících IP adres a sítí. Je poskytováno i API pro strojový přístup ke všem datům. DShield je provozován skupinou dobrovolníků a sponzorován institutem SANS[24]. Kořeny této platformy sahají až do roku 2001.

[19] https://misp-project.org/

[20] Původně zkratka z *Malware Information Sharing Platform* protože již však platforma neslouží zdaleka jen ke sdílení informací o malware, používá se dnes pouze název MISP.

[21] https://warden.cesnet.cz/

[22] Intrusion Detection Extensible Alert, https://idea.cesnet.cz/

[23] https://www.dshield.org/

[24] https://www.sans.edu/

2.3.3 Agregace, korelace, obohacení a prioritizace hlášení

Konečným cílem generovaných, případně i sdílených, hlášení je jejich zpracování člověkem (bezpečnostním analytikem, členem CSIRT týmu, administrátorem sítě apod.), případně automatizovaným bezpečnostním nástrojem. Účelem takového zpracování může být obrana proti probíhajícím útokům, odhalení již úspěšných útoků a zhodnocení jejich následků, či prevence před možnými budoucími hrozbami.

Před tímto využitím je obvykle vhodné data předzpracovat. To může zahrnovat agregaci a korelaci hlášení, obohacování o doplňující data či prioritizaci událostí dle důležitosti. Tyto kroky jsou v následujícím textu stručně popsány.

Agregace a korelace

Agregace a korelace bezpečnostních hlášení je proces, při kterém je analyzováno větší množství hlášení z jednoho nebo více zdrojů, jsou mezi nimi vyhledávány podobnosti či souvislosti a cílem je snížit množství hlášení a zvýšit jejich informační hodnotu. Jinými slovy, cílem tohoto procesu je z nízkoúrovňových hlášení vytvořit záznamy komplexně popisující události na vyšší úrovni abstrakce (nazývané např. *meta-alerts* [65] nebo *hyper-alerts* [66]). Tedy například z několika hlášení o skenování sítě, podezřelých pokusech o přihlášení se na některé stroje a následném hlášení o abnormálním chování jednoho z těchto strojů vytvořit hlášení o pravděpodobné kompromitaci tohoto stroje, včetně detailů, jak k ní došlo. Také však může jít jen o jednoduchou agregaci podobných hlášení opakujících se v čase.

Různými způsoby agregace a korelace bezpečnostních hlášení se zabývá řada výzkumných prací již od devadesátých let minulého století. Shrnutí problematiky a přehled navržených metod poskytují například práce [67, 65, 66, 68]. Z hlediska této práce však tento krok zpracování není příliš podstatný a proto dále není podrobněji popisován.

Obohacování

Dalším možným krokem zpracování je obohacování – doplňování údajů v hlášeních o další informace získané z jiných zdrojů, zejména informace o hlášených IP adresách či jiných identifikátorech. Může jít například o zjišťování doménových jmen, geolokačních informací či přítomnosti na blacklistech pro nahlášené škodlivé adresy, určování důležitosti cílů příchozích útoků, zjištění kontaktů na správce dotčených zařízení apod. Mnohé z těchto údajů mohou být užitečné i jako vstup pro hodnocení škodlivosti síťových entit, resp. predikci jejich budoucího chování, proto je z hlediska této práce obohacování velmi důležité.

Je to však především praktická, implementační záležitost, v akademické sféře mu není věnována velká pozornost. Ve větší či menší míře je obohacování implementováno ve většině komerčních bezpečnostních řešení a SIEM[25] systémů. Mezi zástupce open-source systémů patří například Cortex[26], systém specializovaný na získávání informací o IP adresách a jiných identifikátorech, který je možné integrovat se systémy The Hive[27], nebo MISP. Získávání informací o IP adresách, i když ne přímo za účelem obohacování hlášení, je také klíčovou úlohou systému NERD, vyvinutého autorem této práce a popisovaného podrobně v kapitole 6.

[25] *Security Information and Event Management system*, systém pro správu bezpečnostních informací a událostí

[26] https://github.com/TheHive-Project/Cortex

[27] Open-source platforma pro podporu analýzy a reakce na bezpečnostní incidenty, https://thehive-project.org/

Prioritizace

Poslední fází zpracování hlášení před jejich prezentací člověku je jejich prioritizace, tedy seřazení dle závažnosti hlášené události [65, 69]. Toto seřazení je důležité, neboť operátoři jsou obvykle zavaleni velkým množstvím bezpečnostních hlášení a nemají dostatek kapacit věnovat se dostatečně všem. Automatické seřazení dle určitých kritérií tedy pomáhá věnovat se přednostně těm událostem, které představují největší hrozbu.

Například skenování sítí je dnes tak běžnou záležitostí, že hlášení o příchozím skenu zpravidla nikdo nevěnuje pozornost. Pokud však skenování pochází z pracovní stanice ve vnitřní síti, jde o událost, kterou je vhodné prozkoumat blíže, protože to může naznačovat, že je daná stanice nakažená malwarem. Pokud však zároveň přijde hlášení o DDoS útoku na kriticky důležitou službu, jde jistě o ještě závažnější událost, kterou je třeba řešit přednostně.

Pro prioritizaci hlášení na základě různých kritérií byla navržena řada metod. Jsou založeny například na fuzzy logice [70], na neuronových sítích, které se učí přiřazovat třídu závažnosti dle příkladů zadaných dříve člověkem [71], nebo třeba na aplikaci Dempster-Shaferovy teorie pro odhadnutí spolehlivosti hlášení, na němž je pak prioritizace také založena [72]. Dále jsou používány různé metody z oblasti vícekriteriální analýzy variant (angl. *Multiple Criteria Decision Analysis*, MCDA) [73], což je vědní obor zabývající se metodami rozhodování podle množství často konfliktních kritérií.

Metody prioritizace mají totiž zpravidla společné to, že závažnost hlášení je ovlivňována řadou různých kritérií. Kromě typu útoku patří mezi taková kritéria například ohodnocení důležitosti cíle ve vnitřní síti či data o známých souvisejících zranitelnostech – tedy parametry vycházející ze znalosti konkrétní sítě – dále například důvěryhodnost zdroje dat, informace, zda jde jen o potenciální hrozbu nebo hlášení o již proběhlém útoku, jestli byl útok úspěšný, před jakou dobou k němu došlo apod.

Kapitola 3

Přehled související literatury

Tato kapitola představuje přehled literatury v oblastech úzce souvisejících s hlavním tématem této práce. První podkapitola uvádí seznam existujících prací popisujících různé charakteristiky zdrojů škodlivého chování. Znalosti těchto charakteristik jsou využity při návrhu metody hodnocení reputace v této práci. Druhá podkapitola pak popisuje dříve navržené metody pro vyhodnocování reputace a práce zabývající se predikcí budoucích zdrojů škodlivého chování, včetně popisu jejich nedostatků.

3.1 Charakteristiky zdrojů škodlivého chování

Škodlivé aktivity na internetu lze zpravidla spojit s konkrétními identifikátory, jako jsou například IP adresy, z nichž přichází nežádoucí provoz, či doménová jména a URL, prostřednictvím nichž je distribuován malware či slouží např. k phishingu. Řada prací se v minulosti zabývala analýzou různých charakteristik těchto škodlivých entit, především IP adres, ať už jde o jejich rozložení v prostoru či např. změny chování v čase.

Jednou z prvních takových prací je [12] od Collinse a kol. z roku 2007. Autoři zde tvrdí, že stroje v některých sítích jsou více náchylné ke kompromitaci (např. nakažení malwarem) a vyčištění nakažených strojů v nich trvá v průměru déle, než v jiných sítích. Vlastnost těchto sítí nazývají *uncleanliness* a navrhují metodu umožňující tuto vlastnost kvantifikovat. Zjednodušeně řečeno jsou využity informace o známých škodlivých adresách (na základě sledování komunikačních kanálů botnetů a podle hlášení o detekovaném skenování, spamování a phishingu) a na základě zjištěných prostorových a časových korelací pak lze určit, ve kterých sítích lze pravděpodobně očekávat další zdroje škodlivého chování.

Do jisté míry podobná práce [13] se pak zabývá měřením počtu škodlivých IP adres v jednotlivých autonomních systémech (AS). Škodlivé adresy jsou zde rozpoznávány s využitím několika blacklistů různých typů a také vlastních nástrojů pro detekci spamu. Autoři ukazují, že některé AS vykazují výrazně vyšší míru zastoupení škodlivých adres než jiné. Zatímco ve většině případů je to pravděpodobně pouze výsledkem špatných bezpečnostních pravidel a opatření aplikovaných v dané síti, existují i AS, které mají na blacklistech více než 80 % jejich IP adres. Takové AS jsou prý nejspíš vytvořeny výhradně za účelem vykonávání škodlivých činností, jako je rozesílání spamu či poskytování phishingových stránek či stránek s malware. Dále jsou v práci zkoumány i vztahy mezi takto škodlivými a jim sousedícími AS a je například ukázáno, že existuje řada AS typu poskytovatel připojení, které sice samy o sobě škodlivé adresy neobsahují, ale poskytují připojení téměř výhradně autonomním systémům s velkým zastoupením škodlivých adres.

Různá míra zastoupení škodlivých IP adres v různých sítích byla dále podrobně zkoumána v sérii prací od Moury a dalších [14, 74, 75, 76, 77, 78, 79]. V těchto pracech autoři popisují tzv. špatná sousedství v internetu (*bad neighborhoods* či krátce jen *BadHoods*), tedy oblasti v IP prostoru, které obsahují výrazně nadprůměrné množství škodlivých adres. Tento fenomén je v jednotlivých pracech zkoumán z různých úhlů pohledu.

V práci [14] byla poprvé představena myšlenka špatných sousedství a termín *bad neigborhoods*, a to v souvislosti s filtrováním spamu. Bylo zde navrženo agregovat blacklisty uvádějící jednotlivé IP adresy jako zdroje spamu do seznamů celých prefixů[1] délky /24 (tzv. *bad neighborhood blacklists*, seznamy špatných sousedství). V prefixech s velkým počtem blacklistovaných adres je totiž vyšší pravděpodobnost, že i ostatní adresy začnou rozesílat spam, a adresa tedy může být jen na základě příslušnosti do špatného sousedství považována za podezřelou, ačkoliv z ní samotné (zatím) žádný spam odeslán nebyl. Bylo ukázáno, že tento přístup lze efektivně využít při filtrování spamu.

Problematika špatných sousedství v souvislosti se spamem je dále podrobněji analyzována v [74]. Bylo například zjištěno, že pouhých 10 % nejaktivnějších sousedství (prefixů délky /24) je zodpovědných za více než polovinu všech nevyžádaných zpráv.

V některých dalších pracech [77, 78] se pak autoři zabývají i špatnými sousedstvími v souvislosti s útoky na jiné protokoly. Ukazují, že nerovnoměrnost rozložení zdrojů škodlivého provozu a existenci špatných sousedství lze pozorovat i u jiných protokolů a typů útoků, ale konkrétní sousedství jsou pokaždé jiná. Blacklist vytvořený pro jeden typ škodlivého provozu tedy není možné použít pro jiné typy a je tak nutné vytvářet vždy blacklisty specifické pro daný typ.

Další z prací [75] je věnována prozkoumání možností agregace IP adres do větších skupin (špatných sousedství), než je prefix délky /24. Jednou z možností je agregovat všechny prefixy do větších bloků konstantní velikosti, tzn. použít kratší délku prefixu. Tato metoda se však ukázala jako ne příliš vhodná, protože zatímco umožňuje výrazně snížit množství záznamů v blacklistu, výrazně se tím snižuje i jeho přesnost. Druhou možností je agregovat sousední prefixy do větších bloků jen tehdy, pokud jsou si dostatečně podobné (dle počtu škodlivých adres v prefixu). Výsledný blacklist tak obsahuje prefixy různých délek. Výhodou je, že přesnost takto agregovaného blacklistu se oproti původnímu sníží jen mírně, snížení počtu záznamů však také není tak výrazné.

V práci [76] autoři analyzují řadu existujících blacklistů třetích stran z hlediska možnosti agregace zde uvedených IP adres do seznamů špatných sousedství a jejich následného použití pro filtrování spamu.

V roce 2013 pak Moura shrnul všechny výše uvedené práce ve své disertační práci [77].

Později byla publikována ještě jedna práce na téma špatných sousedství [79]. Jejím cílem je odhalit charakteristiky chování špatných sousedství v čase, tedy zda útočí v určitém časovém období opakovaně a pokud ano, tak kdy. Na základě několika datových sad o různých typech útoků bylo zjištěno například to, že 40–95 % (dle datové sady) špatných sousedství, zde vždy prefixů délky /24, útočí na stejný cíl opakovaně během více dní. U většiny datových sad pak platí, že útok z jednoho sousedství na jeden konkrétní cíl je v přibližně 65 % případů opakován do jednoho dne, ve více než 90 % případů do 5 dní.

Zhang a kol. v práci [80] pomocí analýzy devíti blacklistů různých typů a jejich porovnáním s provozem v síti regionálního poskytovatele připojení také odhalují řadu charakteristik škodlivých IP adres. Ukazují například, že seznamy škodlivých IP adres se rychle mění, u ně-

[1] Prefixem délky /n je zde a dále v této práci myšlena skupina IP adres, jejichž prvních n bitů má určitou, pro všechny adresy stejnou hodnotu. V IPv4 je takových adres vždy 2^{32-n}. Pojem a způsob zápisu vychází z definice způsobu směrování CIDR (*Classless Inter-Domain Routing*).

kterých blacklistů se během jednoho dne změní více než polovina záznamů. Podobně jako u předchozích prací i zde je analyzováno rozložení škodlivých IP adres v prostoru, které je také shledáno značně nerovnoměrným a zároveň závislým na typu útoku. Zde však bylo rozložení zkoumáno z geografického hlediska (pouze však na úrovni regionů, podle registrátora příslušného ASN), místo rozložení v IP prostoru. Dále bylo zjištěno, že některé blacklistované adresy generují mnohem více provozu, než ostatní (ne všechen tento provoz je však nutně škodlivý, protože adresa, která je na blacklistu kvůli určité škodlivé činnosti, může zároveň generovat i množství jiného provozu, který se škodlivou činností nijak nesouvisí; žádná analýza provozu v tomto ohledu v práci provedena nebyla).

Charakteristikami škodlivých IP adres se zabývá i skupina autorů ze společnosti Google v práci [53] (jež už byla zmíněna v souvislosti se sdílecími platformami v kap. 2.3.2). Namísto veřejných blacklistů jsou zde pro analýzu použity seznamy IP adres, u nichž byly detekovány pokusy o zneužití na šesti různých službách společnosti Google (např. Gmail, YouTube či ReCaptcha). Každý den je pro útoky na tyto služby zneužito kolem 8 milionů IP adres. V souladu s předchozími pracemi je zde zjištění, že geografické rozložení škodlivých IP je značně nerovnoměrné, a to i na úrovni jednotlivých zemí. Dále je v práci ukázáno, že některé zdroje jsou výrazně aktivnější než jiné, konkrétně například 1 % nejaktivnějších adres je zodpovědné za 48–82 % útoků (dle služby, na niž útok míří). V rozporu s ostatními pracemi zde však autoři uvádějí, že existují významné korelace mezi zdroji různých typů útoků, tedy že jedna IP adresa je často využívána k různým typům aktivit. Tento rozdíl může být dán změnou způsobu využívání napadených strojů útočníky během let (tato práce je z roku 2016, ostatní výše uvedené práce převážně z let 2010–2013), spíše je však důvodem jiný typ použitých dat. Lze tedy usuzovat, že korelace mezi některými typy útoků mohou být významné, mezi jinými nikoliv. Z hlediska časových korelací pak práce uvádí, že až 66 % škodlivých adres se objeví jen v jediný den a dále se útoky z nich po delší dobu neopakují.

Dále byly některé vlastnosti škodlivých IP adres zkoumány například v práci od Wahid a kol. [81] a také v některých dřívějších pracech autora [63, 82]. Závěry těchto prací ohledně prostorového rozložení škodlivých IP adres i opakování útoků jsou v zásadě stejné jako u prací uvedených výše.

Celkově lze nejdůležitější poznatky ohledně charakteristik škodlivých IP adres shrnout do následujících bodů:

- Škodlivé IP adresy jsou v prostoru (a to jak v IP prostoru, tak geograficky) rozloženy značně nerovnoměrně. Existují tzv. špatná sousedství, tedy oblasti s výrazně nadprůměrným zastoupením škodlivých adres.
- Toto rozložení zdrojů i další charakteristiky se mohou výrazně lišit podle typu škodlivého provozu, který je sledován.
- Většina IP adres je škodlivá jen po krátkou dobu.
- Některé adresy jsou však aktivní velmi výrazně a dlouhodobě. I malé množství takových adres může být zdrojem velké části útoků.

3.2 Ohodnocování síťových entit a predikce útoků

Tato kapitola uvádí práce, které se zabývají tématy přímo souvisejícími s hlavní náplní této práce, tedy především hodnocením reputace síťových entit a predikcí útoků, a uvádí, v čem se od nich tato práce liší.

3.2.1 Hodnocení sítí

V některých výše uvedených pracech zabývajících se korelacemi škodlivých IP adres v prostoru, tedy existencí sítí s větším podílem škodlivých adres, než je běžné, byly navrženy i metriky umožňující škodlivost sítě vyjádřit číselně. V případě [12] jde o metriku zvanou *uncleanliness*. Ta vyjadřuje pravděpodobnost, že jsou v dané síti nějaké škodlivé adresy, a to na základě počtu zde detekovaných škodlivých adres v minulosti a na pozorované tendenci adres v této síti zůstat škodlivé po delší dobu, jinými slovy na základě toho, jak rychle bývají problémy v této síti odstraňovány. Za sítě jsou zde považovány prefixy pevné délky (vyhodnocovány jsou délky prefixů od /16 po /24).

Ohodnocování prefixů pevné délky (zde pouze /24) je také součástí metody filtrování spamu popsané v práci [14]. Způsob ohodnocení je zde velmi jednoduchý – skóre je rovno počtu adres z daného prefixu, které jsou uvedené na určité skupině blacklistů. Toto skóre je pak použito jako jedno z kritérií v rozhodovacím algoritmu pro klasifikaci emailů.

V práci [13] je pak navrhována metrika hodnotící škodlivost celých autonomních systémů. Metoda je podobná té v předchozí práci, skóre je určeno jednoduše jako procento IP adres v daném AS, které jsou uvedené na některém z několika použitých blacklistů. V závěru této práce pak autoři navrhují i možná využití takové metriky. Mezi ně patří i zveřejňování seznamu nejvíce škodlivých AS za účelem vyvíjení tlaku na jejich provozovatele. Podobný způsob hodnocení autonomních systémů byl později skutečně implementován, a to ve formě služby BGP Ranking[2] od organizace CIRCL, kde si může kdokoliv zjistit hodnocení konkrétního AS vypočítané na základě řady zdrojů dat o škodlivých IP adresách.

Celkově mají metody číselného hodnocení celých sítí výhodu v tom, že jsou poměrně jednoduché, výpočetně nenáročné, a hlavně umožňují předpovědět škodlivé chování i u těch adres, které dosud nebyly detekovány jako škodlivé, stačí, aby škodlivé chování vykazovaly jiné adresy ve stejné síti. Toto je však zároveň i nevýhodou a velkým problémem tohoto přístupu. Na všechny adresy v dané síti je totiž pohlíženo stejně, podle skóre celé sítě. Adresa, která žádné nežádoucí aktivity neprovádí, je tedy hodnocena stejně jako jí sousedící, skutečně škodlivé adresy. Ačkoliv často skutečně platí, že z chování několika sousedících adres lze odvozovat podobné chování ostatních adres v blízkém okolí, neplatí to vždy a pokud ano, ne nutně pro všechny stejně, roli může hrát více faktorů a chování sousedních adres má jen částečný vliv. Správnější by tedy bylo hodnotit jednotlivé adresy a chování ostatních blízkých adres uvažovat jen jako jeden ze vstupů, taková metoda však zatím navržena nebyla.

3.2.2 Prediktivní blacklistování

Další zajímavou skupinou souvisejících prací, zejména v souvislosti s požadavkem na prediktivitu navrhovaného reputačního skóre (tedy schopnost vyjádřit očekávanou míru hrozby v budoucnosti namísto pouhého shrnutí předchozího chování), jsou práce na téma tzv. prediktivního blacklistování (*predictive blacklisting*) [83, 84, 85, 86, 87, 88]. Cílem těchto prací je na základě hlášení o útocích sdílených mezi různými organizacemi vytvořit pro každou zapojenou organizaci blacklist obsahující IP adresy těch útočníků, kteří v blízké budoucnosti nejpravděpodobněji proti dané organizaci zaútočí. Blacklisty v těchto pracech neobsahují jednotlivé IP adresy, ale celé prefixy délky /24, a mají předem danou pevnou velikost (obvykle 1000 prefixů).

[2] https://www.circl.lu/projects/bgpranking/

Základní metodou pro vytváření blacklistů v případě sdílení hlášení více přispěvateli je tzv. *global worst offender list* (GWOL), seznam útočníků s celkově největším počtem nahlášených útoků za určité období. V případě, že by data nebyla sdílena a každá organizace vycházela jen z vlastních dat, pak jde o tzv. *local worst offender list* (LWOL).

Práce od Zhang a kol. [83] je první prací zabývající se vylepšením těchto základních metod, konkrétně je zde navržena metoda vytváření blacklistů nazvaná *highly predictive blacklisting* (HPB). V práci se předpokládá existence centrálního úložiště hlášení, do kterého všechny organizace přispívají a které na jejich základě generuje blacklisty (konkrétně je uvažován systém DShield, jehož data jsou pak využita pro vyhodnocení metody).

Klíčovou myšlenkou metody HPB je vyhodnocení korelací mezi množinami útočníků nahlášenými jednotlivými organizacemi, což pak umožňuje zařadit do blacklistu určité organizace přednostně ty útočníky, kteří nejvíce útočili na organizace podobné. Využívá tak toho, že někteří útočníci vybírají své cíle podle určitých kritérií, nikoliv náhodně, a skupiny organizací, které bývají často cílem podobných útoků, tak mohou těžit ze vzájemné výměny informací více, než z výměny s organizacemi značně odlišnými. Kromě této vlastnosti, tedy jak relevantní je hlášení o určitém útočníkovi pro danou organizaci, je do procesu tvorby blacklistu zahrnut ještě odhad závažnosti hlášených útoků. Ten je založen na rozpoznávání vzorů typických pro známé druhy malwaru, např. na základě čísel cílových portů či rozpoznáním horizontálního skenování.

Kvalita vytvořených blacklistů je měřena pomocí metriky *hit count* – počtu záznamů v blacklistu, které odpovídají některému z útočníků detekovaných v době používání blacklistu, tedy počet záznamů, které se ukázaly jako užitečné. V práci je ukázáno, že blacklisty generované metodou HPB mají pro přibližně 90 % organizací větší *hit count* než blacklisty GWOL či LWOL, v některých případech dokonce několikanásobně.

Na tuto práci pak navazuje Soldo a kol. dvojicí prací [84, 85], ve kterých je metoda značně vylepšena. Nová metoda, nazvaná *Blacklisting Recommendation System* (BRS) je inspirovaná oblastí doporučovacích systémů, jaké se používají např. pro doporučování zboží či filmů v online systémech podle toho, co si koupili či na co se dívali jiní uživatelé s podobnými zájmy. Metoda BRS kombinuje několik přístupů. Pro zachycení časových korelací mezi útoky je použita jednoduchá metoda predikce časových řad založená na exponenciálně váženém plovoucím průměru (EWMA). Pro korelace mezi jednotlivými sítěmi jsou pak využity různé shlukovací algoritmy. Navíc narozdíl od metody HPB se zde využívají i korelace mezi útočníky, nejen mezi cílovými organizacemi. Výsledná metoda dosahuje na stejných datech a se stejně velkými blacklisty výrazně vyšších hodnot *hit count* než HPB.

Jinou navazující prací je [86], v níž autoři upravují metodu HPB pro použití v síti spolupracujících honeynetů[3] a vylepšují algoritmus pro odhad závažnosti útoků zapojením více vstupních informací o útocích (jako je např. počet cílových portů, velikost paketů či doba trvání útoku).

Další práce, [87, 88], se pak spíš než vylepšováním přesnosti blacklistů zabývají ochranou soukromí při sdílení hlášení, tedy tím, jak minimalizovat množství sdílených informací i počet organizací, se kterými je nutné je sdílet. Místo centrálního úložiště hlášení je zde tedy využito peer-to-peer modelu sdílení [87], resp. částečně důvěryhodné centrální autority (která zná jen určitá metadata) [88], a pokročilých kryptografických metod umožňujících určit korelace mezi útoky pozorovanými jednotlivými organizacemi bez nutnosti sdílet celé seznamy útoků.

[3] Honeynet je zde skupina honeypotů umístěných uvnitř jedné sítě.

U všech prací zabývajících se vytvářením prediktivních blacklistů je klíčovým faktorem předpovídání, kteří útočníci budou v blízké době pravděpodobně útočit znovu, což je podobné i jednomu z cílů této disertační práce. U výše zmíněných prací je však jedinou motivací a cílem sestavení blacklistu (potenciálně libovolné velikosti, ačkoliv ve většině prací byla velikost vždy předem pevně dána), nepočítá se tedy s žádným číselným ohodnocením jednotlivých adres či prefixů. Číselné skóre má přitom výrazně větší možnosti využití než pouhá binární informace, zda je adresa na blacklistu či ne. Navíc u těchto prací nejsou nikdy vyhodnocovány samostatné IP adresy, ale vždy celé /24 prefixy, takže i zde platí nevýhody popisované v předchozí podkapitole.

3.2.3 Ostatní

Kromě výše uvedených prací na téma prediktivních blacklistů se predikcí bezpečnostních hlášení zabývá například práce Husáka a Kašpara [89], publikovaná v polovině roku 2018. V této práci jsou využity metody dolování sekvenčních pravidel pro predikci budoucích hlášení. Jako vstupní data jsou použita hlášení ze systému Warden, z nichž jsou vždy použity jen čtyři základní údaje: zdrojová adresa, identifikátor detektoru, kategorie útoku a cílový port. Na základě historických dat jsou pak automaticky odvozena pravidla popisující typické sekvence hlášení. Příkladem je pravidlo říkající, že pokud jedna adresa zaútočí na určitý port v organizaci A i v organizaci B, pak pravděpodobně zaútočí i v organizaci C. Jiný typický příklad říká, že pokud jedna adresa zaútočí na port 2323, pravděpodobně zaútočí i na port 23. Ke každému pravidlu jsou přitom k dispozici i míry vyjadřující jeho spolehlivost. Pravidla s vysokou spolehlivostí pak mohou být využita pro predikci budoucích hlášení, například toho, na jakou organizaci a jaký port určitá adresa příště zaútočí.

Taková predikce je však možná jen v případě, že chování dané adresy odpovídá jednomu z naučených vzorů s vysokou spolehlivostí, v opačném případě není predikováno nic. V této disertační práci není predikce zaměřena na předpovídání takto konkrétních parametrů útoku, je však prováděna pro všechny adresy bez ohledu na to, zda jejich chování odpovídá typickým vzorům či nikoliv. Ještě důležitějším rozdílem však je, že predikce v této práci slouží jinému cíli – je především prostředkem k číselnému ohodnocení škodlivosti dané adresy. Ačkoliv se tedy obě práce zabývají predikcí budoucích hlášení, navrhované metody fungují velmi odlišně a mají jiný cíl. Mohou se tedy vzájemně doplňovat.

Prací, která má z jistého pohledu skutečně podobný cíl jako tato disertační práce, je nedávno publikovaná práce [90, 91] od autorů platformy MISP. Je zde představena myšlenka přiřazování skóre atributům v této sdílecí platformě, tedy např. IP adresám či URL. Toto skóre by mělo sloužit k odhadu, zda je atribut stále relevantní, tedy zda je stále aktivní a představuje hrozbu. To by mělo být založeno jednak na důvěryhodnosti zdroje dat, přiřazených tagů, zejména však na ohlášených pozorováních daného atributu (v platformě MISP tzv. *sighting*). Přestože autoři používají jinou terminologii a cíle jsou popisovány z jiného úhlu pohledu, jde o podobný typ úlohy jako v této disertační práci – na základě předchozích hlášení o nějaké škodlivé entitě určit, zda tato entita i v určitém pozdějším čase stále představuje hrozbu. Navrhovaná metoda výpočtu skóre je však jen velmi jednoduchá – při každém ohlášeném pozorování atributu je jeho skóre nastaveno na tzv. základní skóre a se stoupajícím časem od posledního pozorování klesá dle pevně daného vzorce. Pokud skóre dosáhne nuly, je atribut označen za neaktuální a může být např. odstraněn z blacklistů používaných pro blokování provozu. Význam hodnot mezi maximálním skóre a nulou však není nijak definován. Základní skóre je odvozeno od váženého průměru příznaků přiřazených danému atributu či události, přičemž jsou však váhy příznaků založeny jen na tom,

jak často se který z nich používá (nikoliv na jejich významu). Hodnocení důvěryhodnosti zdrojů pak není v práci řešeno vůbec, resp. je ponecháno na budoucí výzkum. Práci tak lze považovat spíše za představení základní myšlenky, nikoliv kompletní vyspělou metodu. I sami autoři v závěru uvádějí, že jde zatím jen o prvotní výsledky práce na toto téma a metoda je stále ve vývoji. V kontextu této disertační práce je tak spíše potvrzením, že o podobný způsob hodnocení síťových entit je v praxi skutečně zájem.

Kapitola 4

Obecná metoda vyhodnocování reputace síťových entit

Z motivace v úvodu této práce a ze shrnutí současného stavu a jeho nedostatků v předchozí kapitole vyplývá potřeba číselného hodnocení reputace jednotlivých IP adres, případně i jiných identifikátorů, které by vyjadřovalo míru hrozby asociovanou s danou entitou, využívalo všech dostupných informací a navíc bylo prediktivní, tedy vyjadřovalo očekávanou míru hrozby v nejbližší budoucnosti. Tato kapitola se zabývá návrhem právě takové metody hodnocení reputace.

Zde představená metoda je velmi obecná. Uvedený princip lze použít pro hodnocení jakýchkoli entit, které mohou vykazovat nějakým způsobem škodlivé či nežádoucí chování, a to i mimo oblast počítačových sítí. Pro účely této práce jsou však za entity považovány pouze síťové identifikátory, jako např. IP adresy, doménová jména, či čísla autonomních systémů.

V této kapitole je tedy metoda představena v obecné formě, aplikovatelná na různé typy síťových entit. V kapitole 8 je pak tato obecná metoda konkretizována pro použití s IPv4 adresami a bezpečnostními hlášeními ze sdílecího systému.

4.1 Základní koncept

Nejprve je neformálně vysvětlen základní koncept navržené metody. Poté následuje formální definice a další podkapitoly se věnují doporučením pro konkrétní aplikace metody.

4.1.1 Neformální definice FMP skóre

Hlavní myšlenkou navržené metody je ohodnotit každou entitu číslem, které odpovídá *pravděpodobnosti, že bude daná entita vykazovat škodlivé chování během určitého časového intervalu v blízké budoucnosti (predikční okno)*. Tuto pravděpodobnost, a tedy výsledek navrhované metody pro hodnocení reputace síťových entit, označujeme jako *Future Misbehavior Probability score* (zkráceně *FMP skóre*).

Definice reputačního skóre pomocí pravděpodobnosti budoucího škodlivého chování je zvolena proto, že právě predikce budoucích útoků je důležitá pro prevenci a obranu (minulým útokům již nezabráníme), protože je ale predikce vždy založená jen na informacích z minulosti, funguje zároveň tato pravděpodobnost dobře i jako shrnutí předchozích aktivit dané entity.

Teoreticky je možné do skóre zahrnout kromě pravděpodobnosti budoucích útoků i očekávanou míru jejich závažnosti. To je však velmi problematické. Neexistuje totiž žádný obecně použitelný způsob hodnocení míry závažnosti kyberbezpečnostních událostí, ta vždy závisí na konkrétních vlastnostech cílové sítě, v ní aplikovaných pravidlech a mnoha dalších faktorech, a jde tedy o věc značně subjektivní. Tato práce se tedy závažností predikovaných útoků přímo nezabývá. Částečně je ale problematika závažnosti pokryta tím, že jsou predikovány pravděpodobnosti různých typů útoků zvlášť (viz dále). V případě potřeby prioritizace incidentů dle závažnosti lze také FMP skóre zkombinovat s jinými kritérii v rámci existujících prioritizačních metod (viz kap. 2.3.3).

4.1.2 Výpočet FMP skóre

Hodnocení entit pomocí FMP skóre je tedy založeno na předpovědi jejich budoucího škodlivého chování. Tato předpověď by měla být založena na všech dostupných datech o dané entitě – především na informacích o jejím předchozím škodlivém chování, ale i na dalších relevantních datech, která lze k entitě získat.

Způsob odvození pravděpodobnosti budoucích útoků na základě takových dat však nemusí být přímočarý a navrhnout příslušný prediktor ručně by bylo velmi obtížné. Obvykle však není problém získat velké množství dat o předchozích škodlivých činnostech entit z historických záznamů, predikční model je tedy možné vytvořit pomocí metod strojového učení s učitelem. Tento přístup je použit v této práci.

Ideální prediktor, tedy takový, který dokáže vždy přesně předpovědět budoucí chování dané entity, by přiřazoval FMP skóre pouze s hodnotami 1.0 nebo 0.0, podle toho, jestli se entita bude nebo nebude během predikčního okna chovat škodlivě. Takový prediktor je však v praxi nedosažitelný. Jakýkoliv reálný prediktor může pouze určit pravděpodobnost škodlivého chování na základě jemu dostupných informací v čase predikce. Cílem je tedy navrhnout co nejpřesnější prediktor ve smyslu co nejlepšího odhadu pravděpodobnosti pro všechny entity.

Dále je vhodné poznamenat, že v praxi je obvykle nemožné vědět o veškerém škodlivém chování dané entity, jsou známy jen ty události, které byly detekovány a predikční systém o nich přijal příslušná hlášení. Část útoků mohla zůstat nepovšimnuta, buď kvůli nedokonalosti detektorů, nebo jednoduše proto, že zdroj i cíl útoku leží mimo monitorovanou síť. Ve výsledku je tedy možné predikovat pouze budoucí *hlášení* související s danou entitou, nikoliv skutečně provedené útoky. Kvalita predikce útoků tedy do značné míry závisí na kvalitě vstupních dat ve smyslu jejich přesnosti a pokrytí. Čím přesnější detektory a lepší pokrytí sítě, tím více bude FMP skóre odpovídat skutečné pravděpodobnosti útoků a bude tak užitečnější pro praktické použití. Nicméně princip navržené metody je dostatečně robustní na to, aby fungoval i s málo kvalitními daty.

4.1.3 Varianty

FMP skóre může být obecné, předpovídající jakýkoliv typ škodlivé aktivity, resp. jakoukoliv kategorii hlášení, nebo specifické jen pro konkrétní typ. Například můžeme určit FMP skóre v kontextu DDoS útoků a zvlášť FMP skóre v kontextu skenování portů, každé odpovídající pravděpodobnosti budoucích útoků daného typu. Podobně je možné mít různá FMP skóre pro specifické cíle, určující pravděpodobnost útoků například pro konkrétní podsítě či typy služeb. Pokud je v textu potřeba taková FMP skóre rozlišit, je možné použít dolní index, např. FMP_{scan}. Ve zbytku této kapitoly však mezi těmito variantami nebudeme rozlišovat,

protože jediný rozdíl je v tom, co konkrétně je považováno za škodlivé aktivity, které mají být predikovány.

Důležitým parametrem FMP skóre je také délka predikčního okna. Tu je třeba zvolit s ohledem na očekávané použití. V některých aplikacích může být vhodné dlouhé časové okno, jindy je lepší predikovat chování jen do velmi krátké budoucnosti a predikci často opakovat. Někdy může být vhodné provádět krátko- i dlouhodobou predikci zároveň a počítat tak různá FMP skóre s různými délkami predikčního okna. V takovém případě lze pro odlišení použít horní index, např. $FMP^{24\text{h}}$. V této práci je uvažována délka predikčního okna 24 hodin. V praxi by tak mělo být FMP skóre pro každou entitu aktualizováno alespoň jednou denně, vždy na další predikční období.

4.2 Formální definice

Hlavním vstupem pro výpočet FMP skóre jsou hlášení o škodlivých aktivitách prováděných konkrétními entitami. Tato hlášení mohou mít obecně různé formáty a obsahovat různé informace, pro účely navrhované metody však musí obsahovat minimálně následující: (*i*) čas detekce, t, a (*ii*) identifikátor entity (např. IP adresu), která je hlášena jako zdroj dané aktivity, e. Dále je vhodné, pokud hlášení obsahují: (*iii*) typ či kategorii hlášené události, c, (*iv*) objem či intenzitu události, v (přesný význam závisí na typu události, může to být např. počet pokusů o navázání spojení), a (v) identifikátor detektoru, d. V následujícím textu předpokládáme, že hlášení obsahují všech pět těchto atributů, ale metoda může být s určitými omezeními aplikována i pokud jsou dostupné jen první dva.

Hlášení (angl. *alert*) tedy můžeme definovat jako pětici $a = (t, e, c, v, d)$. Množinu všech dostupných hlášení označme A. Čas, ve kterém je prováděna predikce (*aktuální čas* či *čas predikce*), je označován jako t_0. *Predikční okno*, T_p, je definováno jako časový interval délky w_p bezprostředně následující po t_0, tedy $T_p = (t_0, t_0 + w_p)$. Prediktor využívá jako vstup informace o hlášeních přijatých v určitém období v minulosti, v *historickém okně*, $T_h = (t_0 - w_h, t_0)$, kde w_h je délka tohoto okna.

Jeden *vzorek* dat (*sample*) je definován jako souhrn vlastností entity e v konkrétní čas predikce t_0. Každý vzorek je reprezentován tzv. *feature vectorem*[1] $\mathbf{x}_{e,t_0} = (x_1, x_2, ..., x_k)_{e,t_0}$. Tento vektor se skládá z různých atributů získaných jak z dat o hlášeních přijatých během historického časového okna tak z dalších doplňujících dat známých o dané entitě v čase t_0 (více o volbě a výpočtu atributů je uvedeno v kap. 4.3).

Výstup, který má být předpovězen, y_{e,t_0}, je binární hodnota označující, zda k dané entitě existuje nějaké hlášení v příslušném predikčním okně,

$$y_{e,t_0} = \begin{cases} 1 & \text{pokud } \exists a \in A : a = (t, e, \cdot, \cdot, \cdot), t \in T_p \\ 0 & \text{jinak.} \end{cases} \tag{4.1}$$

V případě, že má být vypočítáno FMP skóre specifické pro určitý kontext, může být výše uvedená podmínka více omezující, např. kategorie hlášení musí mít konkrétní hodnotu. Vzorky, pro něž platí $y_{e,t_0} = 1$, náleží do tzv. *pozitivní třídy*, ostatní tvoří tzv. *negativní třídu*.

Úlohou pro strojové učení je vytvořit model, který dokáže pro zadaný feature vector $\mathbf{x}_{e,t_0}$ co nejpřesněji odhadnout pravděpodobnost, že $y_{e,t_0} = 1$, tedy že daná entita bude

[1] Česky např. *vektor rysů* či *vektor atributů*, žádný český ekvivalent však v oboru ustálený není a proto je dále v tomto textu používán původní anglický výraz. Jednotlivé prvky vektoru pak budou nazývány *atributy*.

v predikčním okně nahlášena jako škodlivá. Tato úloha je v oblasti strojového učení označována jako *odhad pravděpodobnosti binárních tříd* (angl. *binary class probability estimation problem*). Jde vlastně o běžnou klasifikaci do dvou tříd, kdy nás však nezajímá přiřazení jedné konkrétní třídy každému vzorku, ale spíš pravděpodobnost příslušnosti vzorků do jednotlivých tříd.

Výstup predikčního modelu, označený $\hat{y}_{e,t_0}$, je odhadem pravděpodobnosti pozitivní třídy pro příslušný feature vector,

$$\hat{y}_{e,t_0} \approx p(y_{e,t_0} = 1 | \mathbf{x}_{e,t_0}). \tag{4.2}$$

Pro vytvoření prediktoru je použit standardní proces strojového učení. Nejprve je potřeba připravit datovou sadu pro trénování modelu. To v tomto případě znamená, že je vybráno několik časových okamžiků v rozmezí, ze kterého jsou dostupná data. Označme je jako vzorkovací časy, $T_s = t_1, ..., t_m$. Pro každý takový časový okamžik, $t_0 \in T_s$, je pak pro každou entitu $e \in E$ vypočítán feature vector a příslušná třída, $(\mathbf{x}_{e,t_0}, y_{e,t_0})$. Tím vznikne datová sada o počtu $|E \times T_s|$ vzorků. Dále bude pro označení vzorku pro jednoduchost používán pouze index i, tedy např. $\mathbf{x}_i$ a y_i.

Takto vytvořená datová sada je pak náhodně rozdělena na dvě části, trénovací a testovací sadu, z nichž první je použita pro natrénování modelu, druhá pro jeho vyhodnocení.

V této obecné části návrhu metody není doporučen žádný konkrétní model strojového učení. Pro různé typy dat mohou být vhodné různé modely a jejich konfigurace, obvykle je proto nutné provést experimenty s různými modely a vybrat ten s nejlepšími výsledky.

Pro vyhodnocení kvality predikce lze použít metriku zvanou *Brierovo skóre* (BS). V případě binárního problému a za předpokladu označení tříd čísly 0 a 1 je BS definováno jako střední hodnota čtverců rozdílů mezi predikovanou pravděpodobností a hodnotou skutečné třídy:

$$BS = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2, \tag{4.3}$$

kde N je počet vzorků použitých k vyhodnocení. Brierovo skóre nabývá hodnot mezi 0 a 1, nižší hodnoty znamenají lepší predikci, tedy přesnější odhad pravděpodobnosti. Cílem při trénování modelu je tedy minimalizace BS.

Jakmile je predikční model natrénován a dosahuje uspokojivých výsledků, může být použit pro přiřazování FMP skóre novým vzorkům síťových entit. Pro každou novou entitu je tedy vypočítán odpovídající feature vector, popisující předchozí s ní související hlášení a další dostupné informace, a ten je použit jako vstup natrénovaného modelu. Výstup modelu, $\hat{y}$, je pak přímo použit jako FMP skóre dané entity v daný predikční čas,

$$FMP(e, t_0) = \hat{y}_{e,t_0} = f(\mathbf{x}_{e,t_0}), \tag{4.4}$$

kde funkce f reprezentuje natrénovaný model.

Protože charakteristiky chování škodlivých entit, na nichž je založena predikce, se mohou v čase měnit, podobně jako se občas může změnit konfigurace detektorů, z nichž jsou získávána hlášení, je v praktickém nasazení nutné v pravidelných intervalech přetrénovávat model nad aktuálními daty.

Pokud je vyžadován výpočet různých FMP skóre pro specifický kontext (např. předvídajících zvlášť jednotlivé typy útoků), je nutné pro každé takové skóre natrénovat samostatný model. Vzorky použité pro trénování takových modelů jsou označeny jako pozitivní ($y_i = 1$)

pouze tehdy, pokud v predikčním okně existuje hlášení splňující příslušná kritéria, např. ohlašuje konkrétní typ útoku, ostatní hlášení jsou ignorována (vzorky mají $y_i = 0$). Vstupní atributy, tedy hodnoty ve feature vectoru, však stále mohou obsahovat informace i o ostatních typech hlášení. To je užitečné zvláště v případě, že se očekávají významné korelace mezi jednotlivými typy útoků (např. útokům typu hádání hesel na SSH často předchází skenování sítě na portu 22 [82, 36], takže informace o hlášeních typu skenování mohou pomoci predikovat hlášení o hádání hesel). Samozřejmě je v takovém případě vhodné udržovat informace o jednotlivých typech hlášení v oddělených atributech (tzn. například počítat počet hlášení pro každý typ zvlášť, nikoliv jako celkový počet).

4.3 Návrh feature vectoru

Jak již bylo zmíněno, feature vector použitý jako vstup pro predikci budoucích hlášení o škodlivém chování síťové entity obsahuje dva základní typy informací: (*i*) atributy odvozené z informací o předchozích hlášeních vztahujících se k této nebo k podobným entitám (např. k sousedním IP adresám) a (*ii*) atributy odvozené z jiných informací než z hlášení (např. zda je entita na nějakém veřejném blacklistu).

Konkrétní množinu atributů je nutné vždy navrhnout specificky pro daný typ entity, tedy např. zda jde o IP adresy či doménová jména, a s ohledem na to, jaká data jsou k dispozici. Zejména v případě atributů založených na hlášeních je však možné navrhnout doporučenou skupinu atributů, které by měly být použitelné ve většině případů.

Jako základní informace, které by měly být vždy obsažené ve feature vectoru, tedy navrhujeme použít:

- Počet hlášení
- Celkový objem či intenzita nahlášených událostí
- Počet detektorů, které příslušná hlášení vygenerovaly
- Čas od posledního hlášení
- Průměr a medián intervalů mezi jednotlivými hlášeními v historickém okně.

První tři hodnoty mohou být vypočítány přes různě dlouhá časová období, např. počty za poslední den a za celé historické okno. Pro každý interval tak dostaneme samostatný atribut. Případně je možné se na tyto hodnoty dívat jako na časovou řadu, spočítat tedy např. počet hlášení za každý den v historickém okně, a pak za atribut použít *exponenciálně vážený plovoucí průměr* (*exponentially weighted moving average*, EWMA) této časové řady. EWMA je často používán jako jednoduchý, ale efektivní prediktor následující hodnoty v časové řadě (např. [92, 39, 93, 84]). Obvykle je definován jako:

$$\bar{x}_t = \alpha x_t + (1 - \alpha)\bar{x}_{t-1}, \tag{4.5}$$

kde x_t je hodnota časové řady v čase t, $\bar{x}_t$ průměr EWMA v čase t a $\alpha \in (0, 1)$ je tzv. *smoothing factor*, hodnota určující, jak rychle klesá váha starších hodnot, tzn. vyšší hodnoty α znamenají větší váhu pro posledních několik hodnot, nižší α pak dává významnou váhu i starším hodnotám. V případě pevně dané délky historického okna, w_h, pak lze místo rekurzivní definice uvedené výše použít následující součet:

$$\bar{x}_{t_0} = \sum_{t'=t_0-w_h}^{t_0} \alpha(1-\alpha)^{t_0-t'} x_{t'}. \tag{4.6}$$

Dále, protože FMP skóre je dáno pouze tím, zda se v určeném čase vyskytovalo hlášení, nebo ne, bez ohledu na jejich počet, může být užitečné použít jako vstupní atribut i EWMA vypočítaný z podobné časové řady, avšak obsahující pouze binární hodnoty – pro každý časový interval buď 0, pokud v tomto intervalu nepřišlo žádné hlášení, nebo 1, pokud alespoň jedno hlášení přijato bylo.

Při výpočtu všech výše zmíněných atributů by měla být brána v úvahu ta hlášení, která hlásí jako škodlivou právě tu entitu, pro kterou je tvořen feature vector. Tak však zachytíme pouze korelace mezi hlášeními o stejné entitě v čase. Pokud pro daný typ entity dává smysl zabývat se i korelacemi v prostoru, tedy že existují korelace mezi chováním blízkých či obecně nějak podobných entit, je možné jako další část feature vectoru použít stejné atributy, avšak beroucí v úvahu i hlášení o všech ostatních entitách, které jsou dle nějakého kritéria dostatečně podobné hlavní entitě, tedy té, pro kterou je vektor počítán (v případě IP adres to mohou být např. všechny IP adresy ve stejném /24 prefixu nebo ve stejném autonomním systému).

Mnohé z atributů mohou nabývat velmi vysokých hodnot (např. počet hlášení či jejich celkový objem) což může některým metodám strojového učení činit problémy. Je proto vhodné transformovat hodnoty takových atributů do nižších hodnot, ideálně do řádu jednotek či přímo do intervalu $[0, 1]$. Ve většině případů je navíc vhodné, aby taková transformace byla nelineární, např. logaritmická. To je motivováno tím, že např. rozdíl, zda je počet hlášení 1000 nebo 1001, je intuitivně méně významný, než rozdíl mezi 1 a 2 hlášeními, přestože aritmetický rozdíl je stejný. Právě logaritmická transformace zachovává větší rozdíly u malých hodnot a snižuje rozdíly u hodnot větších. Doporučujeme tedy všechny atributy vyjadřující nějaký počet transformovat funkcí $log(x + 1)$. U atributů vyjadřujících časový interval je situace složitější, protože např. v případě, že nebylo v historickém okně přijato žádné hlášení, není jak určit čas od posledního hlášení či průměrný interval mezi hlášeními. Logicky se nabízí možnost nastavit takový interval na nekonečno, to by však i po logaritmické transformaci zůstalo nekonečnem, což může být pro další zpracování problém. Pro časové atributy je proto vhodnější funkce $exp(-x)$, která nekonečno převádí na 0 a nízké hodnoty převádí na čísla blízká 1. Takto transformované atributy jsou navíc konzistentní s ostatními v tom smyslu, že mají hodnotu 0, pokud k dané entitě neexistuje žádné předchozí hlášení, a nabývají tím vyšších hodnot, čím je entita hlášena častěji (ačkoliv taková konzistence samozřejmě není pro strojové učení nezbytná, je to příjemná vlastnost usnadňující orientaci v datech v případě, že je při nějakém experimentu třeba určité vzorky analyzovat ručně).

4.4 Nevyvážená data a rekalibrace

Mnoho metod strojového učení dosahuje špatných výsledků, pokud je datová sada použitá pro učení významně nevyvážená, tedy počet vzorků v jednotlivých třídách se velmi liší [94, 95]. Lze očekávat, že ve většině aplikací výše navržené metody budou data právě takto nevyvážená, konkrétně že entity nahlášené v predikčním okně jako škodlivé budou tvořit jen malý zlomek z celkového počtu všech známých entit. V datové sadě tak bude mnohem více vzorků negativní třídy, než pozitivní třídy.

Obecně existuje několik přístupů, jak datovou sadu vyvážit. Jednoduchý a často používaný způsob je podvzorkování (*subsamplig*) třídy s více vzorky (*majoritní třída*). Dalším způsobem je nadvzorkování (*supersamplig*) třídy s méně vzorky (*minoritní třída*), a to buď duplikací existujících vzorků, nebo vytvořením nových vzorků podobných těm stávajícím (např. metodou SMOTE [96]). Nadvzorkování je složitější a přináší jisté nevýhody, takže je obvykle voleno jen v případě, že je datová sada příliš malá na to, aby bylo možné použít podvzorkování. To však není případ hlášení o škodlivých síťových entitách, v praxi obvykle není problém získat miliony takových hlášení, takže je dále v této práci použit jednoduchý přístup náhodného podvzorkování majoritní třídy.

Vzorkování by mělo být aplikováno jen na trénovací datovou sadu, v testovací sadě by měl být zachován realistický poměr počtu vzorků v jednotlivých třídách. Toto však porušuje základní předpoklad strojového učení, že vzorky v trénovací a testovací sadě musí mít stejné rozložení. Při běžné binární klasifikaci to nevadí, avšak odhady pravděpodobnosti příslušnosti do tříd mohou být tímto značně zkresleny (pro podrobnosti viz [97]). Naštěstí, jak je ukázáno v [97], toto zkreslení lze snadno kompenzovat, tzv. rekalibrovat. Model je natrénován na podvzorkované (a tedy vyvážené) datové sadě a jeho výstup, $\hat{y}_s$, je pak transformován pomocí následujícího vzorce:

$$\hat{y} = \frac{\beta \hat{y}_s}{\beta \hat{y}_s - \hat{y}_s + 1}, \tag{4.7}$$

kde $\beta = \frac{N^+}{N^-}$ a N^+, N^- představují počet vzorků pozitivní a negativní třídy v původní datové sadě (za předpokladu, že negativní třída je majoritní).

Alternativně, pokud to dovoluje implementace použitého modelu strojového učení, je možné místo podvzorkování nastavit váhu všech vzorků negativní třídy na hodnotu β. Dále v této práci (kapitoly 8 a 9) je však přesto použita metoda podvzorkování a následné rekalibrace, mj. proto, že podvzorkování znamená výrazně menší trénovací sadu a tedy nižší paměťové a časové nároky na trénování modelu, přičemž dle několika provedených experimentů jsou výsledky téměř stejné jako při použití vah.

Kapitola 5

Detekce bezpečnostních událostí

Metoda ohodnocování reputace prostřednictvím FMP skóre, tedy pomocí predikce budoucího chování, navržená v předchozí kapitole, je obecná, aplikovatelná na různé typy entit a různá vstupní data. Aby ji bylo možné vyhodnotit na reálných datech, zabývají se další kapitoly její aplikací v konkrétnější podobě – hodnocení reputace škodlivých IP adres na základě hlášení z různých detekčních systémů a dalších doplňujících dat z reputační databáze – a souvisejícími činnostmi. Tato kapitola se věnuje získávání hlášení, tedy oblasti detekce bezpečnostních událostí.

Pro analýzu chování škodlivých IP adres, potřebnou k návrhu konkrétních parametrů výpočtu jejich FMP skóre, i k vyhodnocení výsledné metody je zapotřebí získat dostatečné množství dat typu hlášení o bezpečnostních událostech, a to nejlépe z různých zdrojů a o různých typech útoků. Žádné volně dostupné datové sady tohoto typu však neexistují. Díky spolupráci s organizací CESNET, provozovatelem české národní sítě pro výzkum a vzdělávání (NREN), má však autor přístup k různým datům z této sítě. V době začátku prací na této disertační práci (tj. kolem roku 2012) zde z hlediska hlášení bezpečnostních událostí sice již bylo nasazeno několik detektorů škodlivého provozu, především různých honeypotů, a byl zajištěn centrální sběr hlášení z těchto detektorů prostřednictvím systému Warden, pro potřeby kvalitní analýzy chování útočníků však množství a různorodost takto dostupných dat nebyly ani zdaleka dostatečné.

Část úsilí věnovaného této disertační práci byla proto směřována do oblasti detekce škodlivého provozu, především za účelem zvýšení počtu, různorodosti a kvality detektorů, které by následně mohly být nasazeny jak v síti CESNET, tak případně i v jiných sítích.

Hlavním výsledkem práce autora v této oblasti je nový framework pro snadnou implementaci detekčních nástrojů využívajících data o IP tocích a řada v něm implementovaných detektorů, včetně několika zcela nových detekčních metod. Ty výrazně pomohly v získání vhodné a dostatečně velké datové sady pro hlavní části této práce, samozřejmě jsou však i samy o sobě významným přínosem. Následující podkapitoly popisují výsledky v této oblasti.

Jedním z dalších výsledků zaměření autora na detekci škodlivého provozu je i práce [30], ve které je navržen nový způsob detekce DDoS útoků. Detekční metoda je unikátní tím, že k analýze dochází již v exportéru toků, dřív než jsou záznamy o tocích exportovány na kolektor. Motivací je skutečnost, že při velkých DDoS útocích je často generováno obrovské množství záznamů o tocích (často jeden záznam pro každý paket útoku), což může vést k přetížení kolektoru. Detekce útoku již na exportéru umožňuje související toky odfiltrovat a přetížení tak zabránit. Na kolektor jsou pak zasílány jen stručné statistiky shrnující data

z odfiltrovaných toků. Vzhledem k tomu, že výsledky tohoto detektoru nakonec pro účely této práce nebyly použity, není zde tato metoda podrobněji popisována.

5.1 Zaměření

Všechny dále popisované detekční metody i framework pro jejich implementaci jsou založené na analýze dat o IP tocích (dále také *flow data*). Tento přístup má několik výhod. První je škálovatelnost, neboť nemá tak velké hardwarové nároky jako zpracování jednotlivých paketů, ani nevyžaduje instalaci žádného software na koncové systémy v síti. Druhou výhodou je to, že monitorování provozu pomocí NetFlow či IPFIX je již v mnoha sítích běžně nasazeno, ať už prostřednictvím routerů nebo specializovaných sond, takže pro nasazení takovýchto detekčních metod často stačí nainstalovat analyzátor na jeden centrální kolektor a nevyžaduje tedy žádné zásahy do síťové infrastruktury.

Narozdíl od mnoha jiných prací a většiny existujícího software je práce popisovaná v této kapitole zaměřena nejen na detekci útoků v koncových sítích, ale i na detekci v „transportních“ sítích, jako jsou sítě ISP či NREN. V těchto sítích sice výsledky detekce nebývají přímo využity pro blokování škodlivého provozu (tyto sítě zpravidla žádný provoz neblokují, s výjimkou velkých DDoS útoků ohrožujících infrastrukturu či případů, kdy o blokování požádá sám zákazník, tedy správce připojené koncové sítě), díky možnosti pozorovat velké množství síťového provozu na mnoho různých cílů však umožňuje detekovat větší množství útoků a tedy i odhalit více jejich zdrojů. To je výhodné (mimo jiné) právě pro účely této práce, kde je třeba získat mnoho dat o škodlivém provozu a jeho zdrojích.

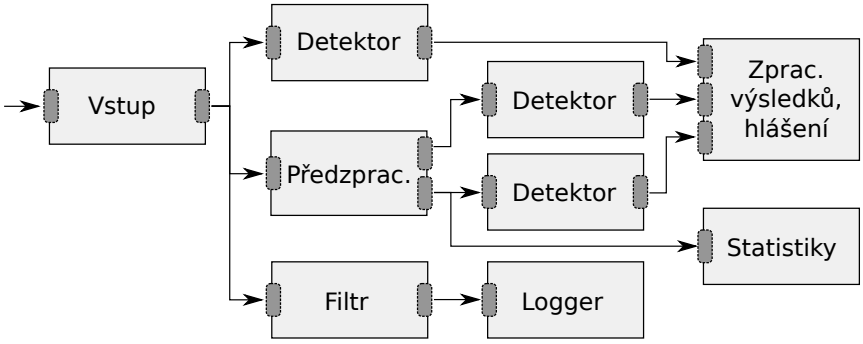
Protože v době začátku těchto prací neexistoval žádný framework pro snadné prototypování a implementaci nástrojů pro analýzu flow dat, bylo rozhodnuto takový framework vytvořit. Cílem bylo jednak usnadnit vlastní práci na budoucím vývoji nových detekčních metod, zároveň byl ale framework zpřístupněn široké komunitě a usnadňuje tak snáze navrhovat a implementovat nové nástroje pro analýzu síťového provozu komukoliv.

V následujících letech tento framework skutečně umožnil vytvoření a implementaci mnoha detektorů škodlivého provozu. Podstatná část z těchto detektorů je založena na zcela nových metodách, framework byl využit pro jejich rychlé prototypování a otestování a významně tak přispěl k usnadnění a urychlení výzkumu v dané oblasti. Některé z těchto detekčních metod jsou zcela nebo zčásti výsledkem autora této práce, ty jsou popsány podrobněji v kapitolách 5.4.1 a 5.4.2.

5.2 Systém pro proudové zpracování dat o síťových tocích (NEMEA)

V této kapitole je popsán framework NEMEA (*Network Measurements Analysis*). Ten byl navržen jako platforma pro snadné a efektivní zpracování dat o IP tocích, především za účelem detekce různých bezpečnostních událostí. Mezi jeho hlavní vlastnosti patří:

- Modulární flexibilní architektura, snadná rozšiřitelnost
- Proudové zpracování flow dat v reálném čase
- Vysoká propustnost
- Podpora flow dat rozšířených o položky z aplikační vrstvy



Obrázek 5.1: Příklad několika modulů NEMEA a jejich propojení

Systém NEMEA byl vytvořen týmem lidí z organizace CESNET ve spolupráci s univerzitami ČVUT a VUT v Brně, přičemž autor je jedním ze zakládajících členů tohoto týmu.

Tato kapitola vychází z článku [25]. Podrobně popisuje systém NEMEA a vybrané případy jeho použití.

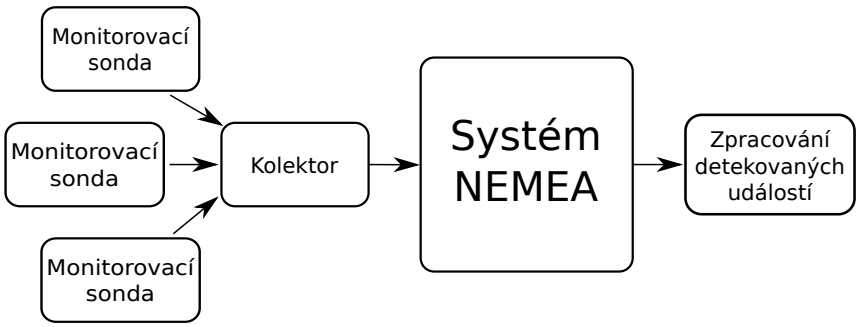
5.2.1 Popis systému NEMEA

NEMEA je navržena jako heterogenní modulární systém. Každý modul implementuje nějaký konkrétní úkol, například předzpracování dat, filtrování, detekci konkrétního typu útoků či anomálií, hlášení výsledků apod. Moduly běží jako nezávislé procesy a předávají si mezi sebou data pomocí jednosměrných komunikačních rozhraní. Data jsou předávána jako potenciálně nekonečný proud jednotlivých zpráv – flow záznamů, hlášení o detekovaných událostech apod. Příklad jednoduché instance systému NEMEA je znázorněn na obrázku 5.1.

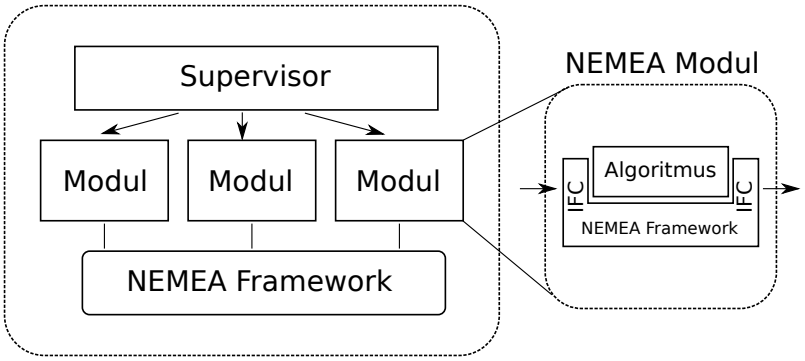
Každá instance systému NEMEA je složena z různých modulů, které mohou být různě propojeny. Různé konfigurace systému tak mohou jako celek pracovat velmi odlišně a poskytovat různé výsledky, závisí to na konkrétních možnostech a potřebách správce, který systém nasazuje pro analýzu dat z určité sítě. Obvykle bývají moduly propojeny do stromové struktury nebo acyklického orientovaného grafu s jedním modulem sloužícím jako hlavní vstup dat do systému. Tento modul sbírá, případně generuje, flow data a posílá je ke zpracování dalším modulům. Na druhé straně systému jsou pak obvykle moduly pro záznam výsledků do souborů, databáze, či pro odesílání hlášení emailem či do jiných systémů.

V typickém nasazení pro velké sítě (viz obr. 2), jaké je například použito v síti CESNET, je jako vstupní modul použit plugin pro kolektor IPFIXcol, který získává NetFlow nebo IPFIX data ze sond rozmístěných na různých místech v síti, převádí je do formátu používaného systémem NEMEA a posílá je dalším modulům. Při monitorování malých sítí či pro testovací účely je však možné použít jako hlavní vstup modul *flow meter* [98], který monitoruje přímo pakety na lokálním síťovém rozhraní (případně čte *pcap* soubor), agreguje je do toků a odesílá flow záznamy přímo ve formátu vhodném pro ostatní NEMEA moduly, takže není nutné provozovat žádné externí sondy ani kolektor.

Systém NEMEA není jen snadno rekonfigurovatelný, je také snadno rozšiřitelný o novou funkcionalitu. Framework tvořící základ systému byl navržen tak, aby umožňoval rychlou a snadnou implementaci nových modulů. Systém tak může být použit nejen pro produkční nasazení, ale i pro rychlé prototypování nových detekčních metod a jejich snadné otestování,



Obrázek 5.2: Typické zapojení systému NEMEA do monitorovací infrastruktury



Obrázek 5.3: Architektura systému NEMEA

ať už offline na testovacích datech či online na reálné síti, a porovnání jejich výsledků s jinými metodami.

Celý systém je volně dostupný jako *open-source*[1]. Součástí základní distribuce systému je množství modulů pro běžné úlohy zpracování dat i několik modulů pro detekci nejběžnějších typů škodlivého provozu. Zamýšleným hlavním přínosem je však poskytnutí jednotné platformy pro analýzu flow dat, nad níž může výzkumná komunita implementovat stávající i zcela nové metody analýzy a tyto implementace sdílet, čímž systém podporuje výzkum v této oblasti.

5.2.2 Architektura

Na obrázku 5.3 je znázorněna základní architektura systému NEMEA. Je zde zobrazeno několik běžících modulů. Každý modul implementuje nějaký algoritmus zpracování dat či detekční metodu a využívá funkce poskytované frameworkem, např. implementaci rozhraní pro komunikaci s ostatními moduly. Celý systém může být řízen a monitorován speciálním nástrojem *Supervisor*.

Nejdůležitější částí frameworku je knihovna TRAP (*Traffic Analysis Platform*), která implementuje komunikační rozhraní a další základní funkce používané všemi NEMEA moduly. Další knihovna, *UniRec*, implementuje stejnojmenný datový formát, který je standardním formátem záznamů posílaných mezi moduly. Obě tyto knihovny jsou implementovány v jazyce C a používány jako binární sdílené knihovny, existuje k nim však i obálka pro jazyk Python, díky čemuž je možné psát moduly i v tomto jazyce. Poslední součástí

[1] https://github.com/CESNET/Nemea

frameworku je knihovna *nemea-common*, která poskytuje efektivní implementace funkcí a datových struktur často využívaných při zpracování síťových dat, jako jsou různé hashovací funkce, hashovací tabulky, Bloomův filtr, B+ strom, prefixový strom apod.

Systém je založen na principu proudového zpracování dat. Standardně jsou tedy všechna data mezi moduly předávána přímo v operační paměti, bez nutnosti meziukládání na disk nebo do databáze (v případě potřeby však uložení samozřejmě možné je). To umožňuje zpracovávat v reálném čase data i z velkých sítí (tj. například několik monitorovaných 100Gb/s linek) na jediném výkonném serveru. V případě potřeby je však možné celý systém i distribuovat – jednotlivé moduly mohou běžet na různých serverech a komunikovat přes síť.

5.2.3 Komunikační rozhraní

Komunikační rozhraní TRAP umožňují předávání dat mezi NEMEA moduly. Jsou jednosměrná, z pohledu modulu je každé rozhraní buď vstupní, nebo výstupní, podle toho, zda skrze něj modul data přijímá či je odesílá. Každý modul může mít více vstupních i výstupních rozhraní. Na výstup jednoho modulu může být připojeno více vstupních rozhraní jiných modulů (data jsou doručena všem), každý vstup je však vždy připojen právě na jeden výstup jiného modulu.

Data jsou přes rozhraní posílána jako potenciálně nekonečný proud krátkých zpráv (každá maximálně 64kB). Každá taková zpráva může reprezentovat záznam o toku, popis detekované bezpečnostní události (*alert*), určité statistiky vypočítané z dat v jednom časovém okně, nebo cokoliv jiného. Formát zpráv je popsán v následující podkapitole.

Rozhraní TRAP jsou ve skutečnosti abstrakcí několika různých metod meziprocesové komunikace. Hlavními typy jsou *UNIX domain socket* a *TCP socket*. První je používán pro komunikaci mezi moduly v rámci jednoho systému, druhý pro komunikaci po síti. K dispozici je i verze TCP rozhraní zabezpečená pomocí TLS. Dále existuje speciální typ *file*, který umožňuje uložit všechny záznamy poslané na takovéto rozhraní do souboru (pokud je použito jako výstupní rozhraní) a později je přehrát (pak je použito jako vstupní rozhraní). Další speciální typ *blackhole* jednoduše zahodí všechny zprávy na něj poslané.

Důležité je, že jádro modulu (tzn. algoritmus zpracovávající data) je zcela abstrahováno od toho, jaký typ rozhraní je použit a s jakými parametry. Pouze přijímá či odesílá data z/na rozhraní identifikované jeho pořadovým číslem. Typ rozhraní i parametry určující, kam má být rozhraní připojeno (např. jméno socketu, IP adresa a port, či název souboru), jsou specifikovány parametry příkazové řádky při spuštění modulu a jsou transparentně zpracovány knihovnou TRAP. Knihovna také automaticky řeší běžné chybové stavy, například při přerušení spojení s druhou stranou komunikačního rozhraní (protože byl druhý modul restartován, nebo došlo k výpadku síťového spojení) se automaticky snaží o znovunavázání spojení.

Vývojář modulu se tedy nemusí zabývat nízkoúrovňovými záležitostmi ohledně získávání a odesílání dat a může se soustředit pouze na algoritmus jejich zpracování. Obecně je cílem frameworku NEMEA výrazně zkrátit dobu nutnou k vyvinutí, otestování a nasazení nových metod analýzy provozu, a také zpřístupnit tyto možnosti i méně zkušeným programátorům, například vědcům, soustředícím se spíše na metody analýzy dat než na programování, či začínajícím studentům.

5.2.4 Datové formáty

Pro přenos dat v systému NEMEA byl vyvinut speciální binární formát UniRec. Je podporován i textový formát JSON a obecná nestrukturovaná data, tyto možnosti jsou však používány jen zřídka.

UniRec je vysoce efektivní binární formát pro ukládání a přenos jednoduchých datových struktur podobný struktuře v jazyce C. Oproti ní však podporuje i položky variabilní délky a především definici položek obsažených v konkrétní struktuře (tzv. *šablona*) až za běhu programu.

V porovnání s jinými formáty pro přenos krátkých zpráv, jako je JSON, IPFIX či MessagePack, se UniRec vyznačuje dvěma zásadními vlastnostmi. Zaprvé, umožňuje velmi rychlý přístup k jednotlivým položkám, protože jejich pozice v záznamu je vždy předem známa, a není tak vůbec nutné záznamy syntakticky analyzovat (tzv. parsovat). Zadruhé, všechny záznamy posílané přes jedno rozhraní mají vždy stejnou šablonu, tedy množinu položek. Toto ve většině případů není problém a výrazně to zjednodušuje práci s daty (pokud se přecejen množina položek u jednotlivých záznamů výrazně liší, lze místo UniRec použít formát JSON).

Každé výstupní rozhraní v NEMEA hlásí svému protějšku šablonu zpráv, které posílá, modul přijímající data zároveň může specifikovat, které položky vyžaduje. Automaticky tak při propojení modulů dochází ke kontrole kompatibility šablon. Formáty šablon navíc mohou být specifikovány za běhu, např. po načtení z konfiguračního souboru, což zvyšuje flexibilitu celého systému.

Dále je vhodné zmínit, že díky flexibilitě datového formátu framework NEMEA přirozeně podporuje zpracování flow záznamů rozšířených o data z aplikační vrstvy (tzv. *L7 extended flow records*, viz kap. 2.1.2).

5.2.5 Centrální konfigurace a dohled

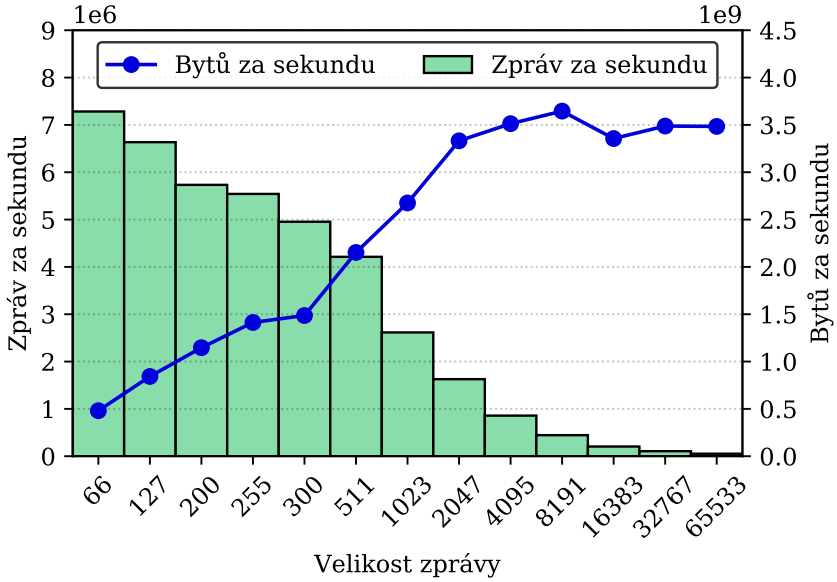
Moduly systému NEMEA mohou být spouštěny ručně jako běžné UNIX procesy. V případě složitějšího systému složeného z mnoha modulů je však vhodnější využít speciální nástroj *nemea-supervisor* [99, 100], který umožňuje celý takový systém centrálně spravovat a monitorovat jeho běh.

Nemea-supervisor je systémový démon, který zajišťuje správu NEMEA modulů dle zadané konfigurace. Umožňuje spouštět a vypínat jednotlivé moduly, jejich skupiny či celý systém, případně upravovat konfiguraci jednotlivých modulů, a to buď pomocí klienta pro příkazový řádek nebo vzdáleně přes protokol NETCONF [101]. Zároveň je sledován stav systému, např. množství zpráv poslaných přes jednotlivá rozhraní či spotřebu CPU a paměti jednotlivými modly. V případě selhání nějakého modulu je automaticky proveden pokus o jeho restart.

5.2.6 Výkonnost

Celková propustnost systému NEMEA samozřejmě závisí především na skladbě modulů, ze kterých se konkrétní instance systému skládá. Je zde však jeden limitující faktor, který lze měřit obecně, a to propustnost komunikačních rozhraní.

Počet zpráv za sekundu, které je možné přenést přes rozhraní typu *UNIX domain socket*, v závislosti na velikosti zprávy, je zobrazen na obrázku 5.4. Rozhraní přenášející nejvíce dat jsou obvykle ta pro předávání flow záznamů a ty bývají poměrně malé, obvykle do 200 B (základní flow záznam, tzn. bez dat z aplikační vrstvy, má ve formátu UniRec 66 B). Jak je



Obrázek 5.4: Propustnost komunikačního rozhraní mezi dvěma NEMEA moduly v závislosti na velikosti zpráv

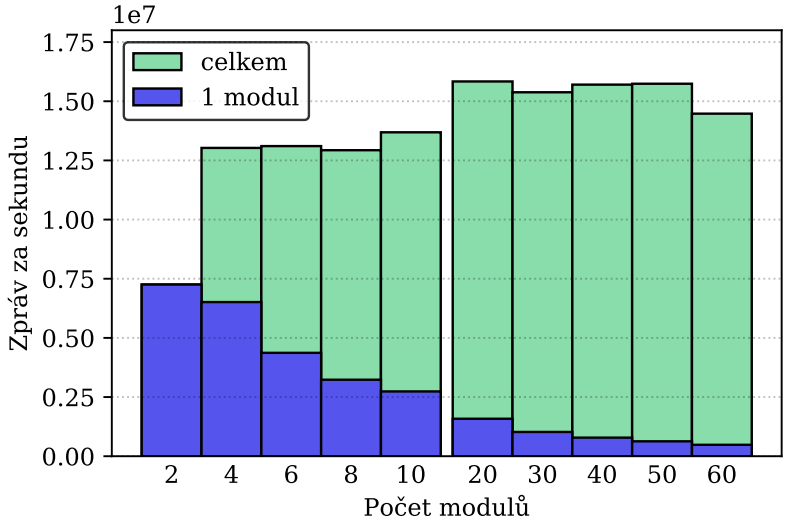
vidět z grafu, v takovém případě lze přes rozhraní mezi dvěma moduly přenést až 7 milionů záznamů za sekundu. Delší zprávy se používají např. pro přenos výsledků detekce, takových zpráv je však řádově méně a vysoká propustnost v tomto případě není důležitá.

V reálném systému NEMEA zpravidla běží modulů více a ty se musí dělit o systémové prostředky. Graf na obrázku 5.5 proto ukazuje propustnost rozhraní při použití více párů modulů zároveň (pro délku zprávy 66 B). Fialová barva znázorňuje průměrnou propustnost jednoho rozhraní, zelená pak propustnost všech rozhraní dohromady. Dle očekávání se stoupajícím počtem modulů klesá propustnost na jedno rozhraní, avšak i při poměrně vysokém počtu současně běžících modulů, např. 20, je stále možné přenést přes každé rozhraní více než 1 milion zpráv za sekundu. Více informací o měření propustnosti rozhraní lze nalézt v [102, 25].

Celková propustnost systému je samozřejmě o něco nižší, než propustnost samotných rozhraní protože velká část výkonu serveru je spotřebována na samotné zpracování dat v modulech. Zkušenosti z praktického nasazení však ukazují, že i při systému složeném z přibližně 20 modulů stačí jeden výkonný server pro zpracování stovek tisíc flow záznamů za sekundu, což je dostatečné pro monitorování středně velké sítě (konkrétně CESNET). V případě potřeby je navíc možné zpracování rozdělit na více serverů.

5.3 Možnosti použití systému NEMEA

NEMEA byla navržena především jako framework pro snadnou implementaci metod pro detekci škodlivého provozu, už základní systém však obsahuje množství modulů, včetně některých jednoduchých detektorů. Tato kapitola stručně shrnuje možnosti, jak lze systém NEMEA v praxi použít. Podrobný popis vybraných detektorů škodlivého provozu je ponechán na pozdější kapitoly.



Obrázek 5.5: Propustnost komunikačních rozhraní při současném běhu více modulů (velikost zprávy 66 B)

5.3.1 Detekce útoků na síťové a transportní vrstvě

Mezi nejčastější nežádoucí aktivity na internetu patří skenování sítě. Samo o sobě je obvykle neškodné, je však útočníky využíváno pro získávání informací a pro hledání zranitelných cílů. Často je i nezbytnou součástí šíření malwaru (síťových červů). Je tedy vhodné mít možnost takové skeny detekovat, zároveň je detekce většiny typů skenování poměrně snadná. NEMEA poskytuje moduly pro detekci jak horizontálního, tak vertikálního skenování portů. Výsledky těchto modulů posloužily např. pro analýzu skenů v práci [33].

Podstatně škodlivějším typem útoku detekovatelným na síťové vrstvě jsou volumetrické útoky odepření služby – *Denial of Service* (*DoS*) a *Distributed DoS* (*DDoS*). Jejich detekce je zdánlivě jednoduchá – v síťovém provozu se obvykle projevují velmi výrazným zvýšením objemu provozu na konkrétní cíl, takové zvýšení však může mít i jiné příčiny a proto v praxi není detekce tak snadná. Navíc pro efektivní blokování útoku je zapotřebí škodlivý provoz co nejpřesněji popsat, navíc spolehlivě a rychle. Systém NEMEA poskytuje pro detekci DoS útoků několik možností, jednoduché útoky dokáže detekovat modul HostStats (viz kap. 5.4.1), jiný modul umožňuje detekovat překročení nastavených limitů množství provozu pro jednotlivé podsítě, dále byla implementována metoda založená na struktuře MULTOPS tree [103], a v době psaní tohoto textu je ve vývoji detektor, jehož metoda je inspirována nástrojem FastNetMon[2]. Dále je k dispozici modul specializující se na detekci amplifikačních útoků (viz kap. 5.4.3).

S pomocí tradičních flow dat obsahujících informace po transportní vrstvu lze detekovat i některé útoky na konkrétní aplikace. V systému NEMEA byl například implementován detektor slovníkových útoků na autentizované služby, jako např. SSH. Detekční metoda, inspirovaná prací [104], využívá statistiky o provozu mezi dvojicemi adres na portu dané služby – počet spojení, počet přenesených paketů a bytů a TCP příznaky v každém spojení. Slovníkové útoky mají v rámci těchto statistik specifickou charakteristiku odlišnou od běžných spojení a na základě toho jsou detekovány.

[2] https://github.com/pavel-odintsov/fastnetmon/

5.3.2 Detekce útoků využívající data z aplikační vrstvy

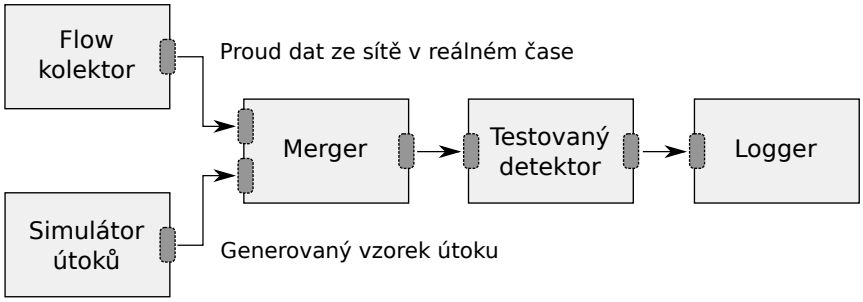
Pokud jsou k dispozici flow data rozšířená o informace z hlaviček vybraných aplikačních protokolů, lze systém NEMEA využít pro detekci řady dalších útoků. Typickým příkladem je např. protokol DNS, analýzou jehož provozu lze odhalit řadu škodlivých činností. Příkladem může být práce [105], ve které je navržena metoda pro detekci DNS tunelů[3] – v případě přenosu dat tunelováním přes DNS jsou data zakódována do doménových jmen a taková doménová jména mají výrazně odlišné charakteristiky (např. průměrnou délku, entropii), než je běžné. Analýzou obsahu DNS požadavků a odpovědí lze tedy takové tunely odhalit. Navržená metoda byla ověřena pomocí nově vyvinutého modulu pro NEMEA.

Dalším příkladem protokolu, na nějž bývá směřována řada útoků, je protokol *Session Initiation Protocol* (SIP). Byly navrženy NEMEA moduly pro detekci skenování uživatelských jmen, slovníkové hádání hesel [23] a pokusy o zneužití SIP ústředen hádáním vytáčecího schématu [22]. Tyto metody jsou podrobněji popsány v samostatné kapitole 5.4.2.

Jako ukázka toho, že pomocí frameworku NEMEA skutečně lze rychle implementovat novou detekční metodu a rychle tak reagovat na novou hrozbu, může posloužit například detekce pokusů o zneužití zranitelnosti *Heartbleed*. Toto označení dostala kritická chyba v knihovně OpenSSL objevená a zveřejněná v roce 2014, která umožňuje vzdálenému útočníkovi číst náhodné úseky paměti ze serveru se zranitelnou verzí knihovny. Přestože framework NEMEA byl v té době ještě v rané fázi vývoje, podařilo se autorovi této práce ve spolupráci s několika kolegy během několika dní od zveřejnění zranitelnosti implementovat zásuvný modul pro exportér síťových toků, extrahující několik položek z SSL/TLS hlaviček paketů, a modul pro NEMEA, který tato data analyzoval a byl schopen detekovat úspěšné pokusy o zneužití této zranitelnosti. Během krátké doby tak bylo odhaleno více než 1000 zařízení v síti CESNET, ze kterých byla pomocí zranitelnosti Heartbleed vyčtena část paměti. Zároveň se podařilo získat zajímavé statistiky o četnosti a charakteru těchto útoků. Například se ukázalo, že v naprosté většině případů jde ve skutečnosti pravděpodobně jen o testy, zda je cílový server zranitelný, či ne, ale objevily se i případy, kdy útočník přečetl z jednoho serveru stovky MB dat. Další informace lze nalézt v [24].

Pro detekci škodlivého provozu však není vždy nutné navrhovat speciální metodu a implementovat nový modul, často stačí základní moduly, např. obecný filtr. Tímto způsobem byly například později v roce 2014 detekovány pokusy o zneužití další známé zranitelnosti – *Shellshock*. V tomto případě stačilo vyhledávat jistý regulární výraz v hlavičkách protokolu HTTP, a protože flow data rozšířená o informace z HTTP již byly v síti CESNET k dispozici, stačilo spustit filtrovací modul s daným regulárním výrazem jako filtračním pravidlem a zaznamenávat výsledky. Bylo tak odhaleno velké množství skenů testujících zranitelnost i mnoho skutečných pokusů o zneužití (snaha o spuštění kódu, který na serveru stáhne a spustí malware). Žádný z nich však podle dostupných dat nebyl úspěšný. Dalším příkladem ad-hoc detekce s pomocí filtrovacího modulu je zaznamenávání přístupů na IP adresy a URL *command & control* serverů získané při předchozí analýze malware, což umožňuje odhalení zařízení v monitorované síti, která jsou tímto malware napadena (podrobněji v [106]).

[3] V některých, obvykle bezdrátových, sítích je přístup k internetu povolen jen přihlášeným uživatelům, DNS protokol však často nijak omezen není. V takovém případě lze k internetu přistupovat i bez přihlášení pomocí tunelování dat přes DNS. DNS tunely lze samozřejmě využít i k jiným účelům, např. pro skrytí provozu před IDS systémy.



Obrázek 5.6: Příklad testování detektoru pomocí smíchání reálného provozu s uměle vygenerovaným vzorkem útoku

5.3.3 Zpracování hlášení o detekovaných událostech

Detekční moduly zpravidla generují jako výstup jednoduché záznamy s informacemi popisujícími detekované bezpečnostní události. Další zpracování těchto záznamů, jako je např. logování či hlášení operátorům, probíhá v systému NEMEA unifikovaným způsobem prostřednictvím speciální sady modulů. Ty převádí výstupy jednotlivých detektorů do jednotného formátu a výsledné zprávy mohou ukládat do souboru či do databáze, odeslat je emailem, nebo do systému Warden (viz kap. 2.3.2). Protože jsou tyto moduly implementovány jako jednoduché skripty v jazyce Python, je snadné přidat podporu pro další typ výstupu detektoru nebo možnost exportu do jiného systému.

5.3.4 Offline testování

Systém NEMEA nemusí být používán jen pro zpracování dat v reálném čase přímo ze sítě (tj. „online“). Může také pracovat „offline“, tedy zpracovávat dříve uložená data. Byly vytvořeny moduly pro načítání flow dat ze souborů ve formátu populárního nástroje *nfdump*, databáze *fastbit* používané kolektorem *IPFIXcol*, a také z běžných CSV souborů. Také je možné na úrovni knihovny TRAP uložit do souboru data posílaná na výstupní rozhraní nějakého modulu a později je použít jako zdroj dat pro vstupní rozhraní jiného modulu.

Toto umožňuje posílat opakovaně stejná data do množiny modulů, což je užitečné jednak pro vývoj a testování, ale i pro výzkum, např. pro porovnání výsledků různých detekčních metod či jedné metody s různým nastavením parametrů na stejných datech.

Další užitečnou možností použití systému NEMEA je smíchání více proudů dat do jednoho pomocí modulu *merger*. Například tak lze smíchat proud „online“ flow dat ze sítě s flow daty uměle vygenerovanými simulátorem útoků či s uloženým vzorkem útoku (viz obr. 5.6). Takové zapojení může být použito pro testování schopností detekčních modulů.

5.4 Vybrané detekční metody vzniklé díky frameworku NEMEA

Tato podkapitola stručně popisuje několik detekčních metod vyvinutých v rámci frameworku NEMEA. Detektor HostStats byl navržen a implementován autorem této práce, dále se autor podílel na detektorech útoků na VoIP infrastrukturu. Ostatní uvedené detektory byly s pomocí frameworku vytvořeny jinými autory, jsou zde uvedeny jako ilustrace dalších zajímavých metod, které díky novému frameworku vznikly.

5.4.1 HostStats

Důležitým detekčním modulem, který je standardní součástí NEMEA, je modul HostStats. Detekční metoda implementovaná tímto modulem sice nebyla nikde publikována — je poměrně jednoduchá a přímočará a z vědeckého hlediska nepřináší podstatnou inovaci — z hlediska této práce je však tento modul přesto významný, protože je jedním z nejvýznamnějších zdrojů dat systému NEMEA (podle počtu generovaných hlášení) a tedy jedním z hlavních zdrojů dat použitých v dalších kapitolách.

Tento NEMEA modul přijímá flow data ze sítě a pro každou IP adresu, která se v datech objeví jako zdroj či cíl nějakého toku, vytváří a udržuje záznam s různými statistikami o provozu dané adresy, tzv. profil adresy. Tyto statistiky zahrnují například počet toků, počet přenesených paketů a bytů, počty toků s jednotlivými TCP příznaky a také počet adres, se kterými daná adresa komunikovala. To vše zvlášť pro příchozí a odchozí provoz. Tyto statistiky jsou počítány jednak z veškerého zachyceného provozu a dle konfigurace také zvlášť z provozu protokolů SSH a DNS (resp. portů TCP/22 a UDP/53).

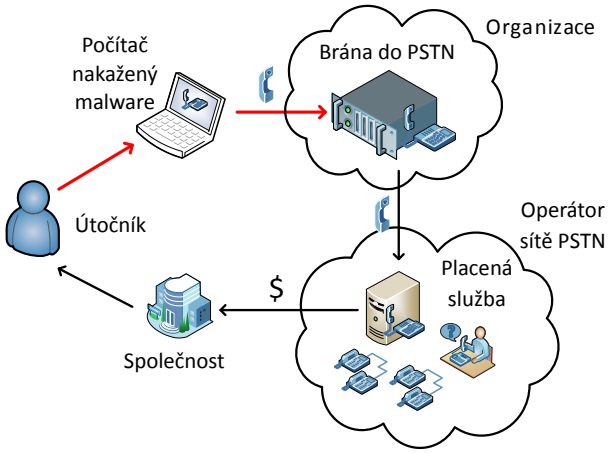
Profily všech adres jsou pravidelně vyhodnocovány (vždy nejpozději po 5 minutách aktivity adresy) a porovnávány s předdefinovanými pravidly pro detekci podezřelého chování. Například horizontální skenování portů je nahlášeno, pokud profil adresy splňuje následující:

- Počet odchozích toků s příznakem SYN je vyšší než nastavená mez (standardně 200).
- Většina ($> 95\,\%$) odchozích toků má nastaven pouze příznak SYN, nikoliv ACK (porovnání čítačů odchozích SYN a ACK), tzn. jde především o neúspěšné pokusy o navázání spojení.
- Podstatná část cílů ($> 20\,\%$) na pokus o navázání spojení neodpoví (porovnání čítačů odchozích a příchozích SYN), tj. druhá podmínka ověřující, že jde především o neúspěšné pokusy o navázání spojení.
- Odchozí provoz je směřován na více různých cílů, než je nastavená mez (200).
- SYN pakety, tedy pokusy o navázání spojení, tvoří podstatnou část ($> 1/3$) celkového odchozího provozu adresy.

Je zřejmé, že tyto podmínky nepostihují všechny případy skenování sítě. Toto pravidlo bylo navrženo (a po mnoho měsíců upravováno a doladňováno na základě zkušeností z reálného provozu) tak, aby bylo minimalizováno množství falešných hlášení, tedy tak, aby pravidlu neodpovídal žádný typ běžného provozu, i za cenu nižší citlivosti. V případě potřeby lze samozřejmě hodnoty mezí snadno upravit.

Podobná pravidla jsou stanovena i pro další typy nežádoucího chování. Konkrétně je modul HostStats schopen detekovat a nahlásit následující typy událostí:

- Horizontální skenování (tzn. skenování jednoho nebo několika portů na mnoha různých cílech)
- Příchozí či odchozí (D)DoS útok (přímý, tj. záplava SYN pakety nebo obecně jakýmkoli jiným provozem)
- DDoS útok pomocí DNS amplifikace (hlášen je buď cíl nebo server zneužitý k odrazu)
- Příchozí či odchozí slovníkový útok na SSH



Obrázek 5.7: Princip zneužití nezabezpečené VoIP ústředny s možností volat do běžné telefonní sítě (PSTN) [22]

Modul HostStats je v principu poměrně jednoduchý a jeho kvalita detekce (a zejména podrobnost hlášení) je obvykle horší než u detektorů specializovaných na konkrétní typ útoků, schopnost efektivně detekovat celou řadu různých typů útoků jediným modulem z něj však přesto dělá cenný nástroj.

5.4.2 Detekce útoků na VoIP infrastrukturu

Příkladem modulů, které se zaměřují na útoky na konkrétní protokol, je dvojice modulů pro detekci útoků na VoIP infrastrukturu, konkrétně na ústředny používající protokol SIP. Tyto moduly jsou zároveň příkladem těch, které využívají i data z aplikační vrstvy. Detekční metody implementované v těchto modulech byly publikovány v [22, 23].

První z modulů se zaměřuje na detekci specifického útoku na špatně zabezpečené SIP ústředny. Ústředny, provozované např. v rámci podnikových sítí, poměrně často umožňují volat z VoIP sítě i na běžná telefonní čísla, obvykle tak, že v rámci SIP protokolu se jako ID volaného uvede telefonní číslo s určitým speciálním předčíslím. V běžném případě by takové volání mělo být umožněno jen autorizovaným klientům z vnitřní sítě, v případě chybně nakonfigurované ústředny je však často umožněno komukoliv, kdo zná (či uhodne) správné předčíslí. Volání přes cizí ústředny se přitom dá snadno zpeněžit – útočník si přes fiktivní firmu zřídí tzv. prémiové číslo s vysokými poplatky za volání, následně si na takové číslo zavolá skrze cizí nezabezpečenou ústřednu a inkasuje poplatky, které musí skrze telefonního operátora zaplatit provozovatel ústředny (viz obr. 5.7).

Takové útoky se skutečně dějí a na síti lze pozorovat řadu pokusů o nalezení takových nezabezpečených ústředen a uhodnutí správného předčíslí. Toto hádání se projevuje jako velké množství požadavků na volání z jednoho klienta (požadavky typu INVITE), zpravidla všechny se stejným telefonním číslem, jen s různými předčíslími. Právě tento provoz se NEMEA modul snaží detekovat.

Na vstupu je nutné získat flow záznamy rozšířené o data ze SIP protokolu, ty zajišťuje exportér toků, např. pomocí zásuvného modulu pro analýzu daného protokolu. Využívány jsou zejména položky: typ požadavku, URI (tedy ID volaného) a návratový kód (pro rozpoznání, zda bylo volání úspěšné). Modul zpracovává pouze požadavky typu INVITE, ve kterých má URI tvar telefonního čísla. Pro každého klienta (tj. zdrojovou IP adresu) se pak vytváří tzv. prefixový strom, ve kterém jsou uložena všechna čísla, na která se tento

klient pokoušel volat. Analýzou tohoto stromu pak lze snadno rozpoznat situaci, kdy jde stále o stejné číslo, jen s mnoha různými předčíslími, a jde tedy zřejmě o snahu uhodnout předčíslí pro volání do telefonní sítě.

Během dvoutýdenního testování této detekční metody v síti CESNET bylo odhaleno téměř 16 tisíc takových útoků, přičemž asi 1 % z nich se dle dostupných dat zdálo být úspěšných (některé z takto „napadených“ ústředen však byly pravděpodobně honeypoty a k žádné škodě tak ve skutečnosti nedošlo).

Druhý modul se zaměřuje na útoky na autentizační mechanizmy protokolu SIP, konkrétně na hádání uživatelských jmen a hádání hesel. Tyto útoky se projevují velkým počtem požadavků nesoucích autentizační informace (obvykle typu REGISTER), buď s různými uživatelskými jmény nebo se stejným jménem a různými hesly (hesla se však přenáší šifrovaně, takže konkrétní zkoušené hodnoty nelze zjistit). Detekční metoda je v principu podobná té předchozí, opět jsou využívána data o síťových tocích obohacená o vybrané hlavičky protokolu SIP, jen jsou tentokrát sledovány opakované pokusy o přihlášení k ústředně. Při více než 20 neúspěšných pokusech o přihlášení z jednoho klienta je nahlášen útok. Podle zkoušených uživatelských jmen modul rozpoznává, zda jde o hádání hesel u konkrétního uživatelského jména, nebo jde zároveň i o hledání validních jmen. Sledováním návratových kódů je navíc modul schopen rozpoznat situaci, kdy bylo hádání hesla úspěšné a došlo tedy ke kompromitaci účtu.

Metoda byla opět po dva týdny vyhodnocována v síti CESNET. Při tom bylo odhaleno přibližně 7000 útoků, při největším z nich vyzkoušel útočník téměř 7 milionů hesel. Sedm detekovaných útoků bylo zřejmě úspěšných, tzn. po mnoha neúspěšných pokusech došlo k úspěšnému přihlášení. Některé z těchto útoků byly nahlášeny administrátorům příslušných ústředen a ti potvrdili prolomení účtu.

5.4.3 Detekce amplifikačních DDoS útoků

Dalším zajímavým modulem je detektor speciálně zaměřený na amplifikační DDoS útoky.

Při tomto typu útoku jsou zneužity servery některého z protokolů, které jsou založeny na UDP a mají tu vlastnost, že odpovědi na některé dotazy mohou být výrazně větší než dotazy samotné. Mezi nejznámější takové protokoly patří např. DNS, NTP, SNMP, chargen, či memcached [107]. Útok pak spočívá v tom, že útočník posílá velké množství takových dotazů na různé servery, přitom ale vždy podvrhne zdrojovou IP adresu a uvede na jejím místě adresu oběti. Servery pak posílají všechny odpovědi na adresu oběti, čímž ji mohou zahltit. Díky tomu, že odpovědi mohou být mnohonásobně větší než dotazy, dokáže takto útočník vygenerovat obrovský provoz i pokud má sám jen nízkou kapacitu připojení. Zároveň je kvůli podvrženým zdrojovým adresám velmi těžké vypátrat skutečný původ útoku.

Zmíněný detektor je zaměřený na identifikaci serverů takto zneužívaných k odrazu a zesílení útoků. Pracuje na základě porovnávání příchozího a odchozího provozu určitého protokolu. Využívá se toho, že při útoku je obvykle posílán opakovaně stále stejný dotaz, takže velikosti dotazů i odpovědí v rámci útoku jsou konstantní. V modulu jsou proto udržovány histogramy počtu paketů a bytů v tocích mezi dvojicemi komunikujících IP adres. Tyto histogramy jsou v pravidelných intervalech analyzovány. Zjednodušeně řečeno, když je zjištěno, že v provozu dvojice IP adres výrazně převažují toky určité velikosti (příp. několika málo velikostí), odpovědi jsou výrazně větší než dotazy a celkové množství takového provozu je větší než určitá mez, je nahlášen amplifikační DDoS útok.

Modul je univerzální a může být nakonfigurován pro jakýkoliv ze zneužívaných protokolů, vždy je třeba jen nastavit číslo portu a hodnoty několika prahů. V síti CESNET je modul nasazen pro detekci na protokolech DNS a NTP, což jsou dle zkušeností protokoly zneužívané k DDoS útokům nejčastěji.

Kvůli nutnosti vidět oba směry provozu je modul obvykle schopen detekovat jen ty situace, kdy je server zneužitý k odrazu umístěný uvnitř monitorované sítě a zdroj i cíl útoku leží vně (případně leží zdroj i cíl uvnitř a server vně, to se však v praxi téměř nestává), nikoliv příchozí DDoS útoky (k tomu slouží jiné detekční moduly). Díky svému specifickému zaměření je však detektor velmi spolehlivý a přesný.

Jeho typickým příkladem využití je situace, kdy je do sítě zapojen nový DNS server, který je nevhodně nakonfigurován (příp. se změní konfigurace stávajícího serveru) tak, že umožňuje zneužití pro amplifikační útoky. Takový server je obvykle brzy objeven některými útočníky a začne být zneužíván k řadě útoků. To je detekováno tímto modulem a hlášení je pak předáno příslušnému správci serveru s doporučením vhodnější konfigurace.

5.4.4 Detekce těžby kryptoměny bitcoin

Poměrně neobvyklým detektorem je modul pro detekci strojů využívaných pro těžbu kryptoměny bitcoin. Přesněji řečeno jsou detekovány stroje pracující v rámci větší skupiny uživatelů (tzv. *pool*) a komunikující se serverem koordinujícím danou skupinu (tzv. *pool server*) pomocí standardního protokolu *stratum*. Těžba kryptoměn samozřejmě není sama o sobě škodlivou činností, může však být nežádoucí v některých konkrétních prostředích. Příkladem jsou výkonné servery pro gridové výpočty, poskytované např. univerzitami a často dostupné studentům či akademikům zdarma pro vědecké výpočty. Pokud jsou takové servery zneužity pro těžbu kryptoměn, jde o vážné porušení pravidel, a proto je vhodné mít možnost takovou činnost detekovat.

Modul analyzuje flow data a na základě pravidel, zahrnujících velikosti paketů, počty paketů ve spojení, TCP flagy a informace o časových intervalech, detekuje spojení pravděpodobně využívající protokol stratum. Detekce založená jen na těchto datech však není příliš spolehlivá a generuje mnoho falešných hlášení. Proto je doplněna o aktivní testování podezřelých serverů. Pokaždé, když je objevena nová IP adresa, jejíž komunikace napovídá, že by mohlo jít o pool server, se modul k této adrese připojí a pokusí se komunikovat protokolem stratum. Teprve když je tento pokus úspěšný, uloží si modul danou IP adresu do seznamu známých pool serverů. O každé IP adrese z monitorované sítě, která komunikuje s některým z takto objevených pool serverů, je pak nahlášeno, že je pravděpodobně využívána pro těžbu bitcoinu.

Tento modul byl podrobně popsán v diplomové práci [108].

5.5 Shrnutí přínosu

Do termínu odevzdání této práce (prosinec 2018) přispěl systém NEMEA ke vzniku 9 recenzovaných publikací na vědeckých konferencích [22, 23, 24, 25, 33, 105, 109, 110, 111] a 7 dalších odborných publikací [102, 106, 112, 113, 114, 115, 116]. Na 5 z nich se přímo podílel i autor této práce. Systém NEMEA byl také využit v 20 bakalářských a 15 diplomových pracech (z nichž 8, resp. 2, vedl autor této práce, u několika dalších byl konzultantem), jejichž náplní byl zpravidla vývoj nějakého NEMEA modulu, příp. rozšíření samotného frameworku.

Většina z vyvinutých detektorů (ne všechny byly publikovány ani popsány v této práci) je v současnosti nasazena pro monitorování provozu v české akademické síti CESNET. Některé i v jiných sítích, systém NEMEA je např. nasazen v síti jednoho velkého českého telekomunikačního operátora a dle občasné zpětné vazby od různých uživatelů na platformě GitHub i na několika dalších místech. Hlášení o škodlivém provozu generovaná detektory v síti CESNET jsou denně používána v praxi pro zajišťování bezpečnosti sítě a také tvoří významnou část dat použitých pro výzkum popisovaný v následujících kapitolách.

Kapitola 6

Reputační databáze síťových entit

Reputační skóre navrhované v této práci je přiřazováno jednotlivým entitám, jako jsou IP adresy, sítě, domény apod., a mělo by být založeno na pokud možno všech dostupných informacích o dané entitě a pravidelně přepočítáváno dle aktuální situace. Je tedy třeba udržovat profily nahlášených zdrojů škodlivého chování, v nichž budou tyto informace ukládány a průběžně aktualizovány. Nejde přitom jen o agregaci hlášení, ale i o získávání dalších relevantních dat z jiných zdrojů.

Systém udržující takové profily můžeme nazývat reputační databází. Přestože vytváření profilů škodlivých IP adres a jiných entit není samo o sobě nová myšlenka, žádná veřejně dostupná databáze tohoto typu, ani žádná otevřená implementace, kterou by bylo možné nasadit pro zpracování vlastních dat, nebyla dosud k dispozici.

Autor této práce se proto mimo jiné věnoval i návrhu a implementaci právě takové otevřené reputační databáze. Jejím účelem je jednak získání a zpracování dat pro tuto disertační práci, ale také praktické využití bezpečnostními týmy při jejich každodenním boji proti kybernetickým hrozbám.

Původní myšlenka takového systému reputační databáze, včetně ohodnocování IP adres reputačním skóre, byla publikována v [117], plánované vlastnosti systému byly také představeny v [118]. Tato kapitola popisuje navrženou architekturu a princip fungování této reputační databáze, technologie použité pro její implementaci, současný stav a vybrané statistiky o jí udržovaných datech.

6.1 Popis systému NERD

Z uživatelského hlediska je systém NERD (*Network Entity Reputation Database*) webový portál, ve kterém může kdokoliv vyhledat jakoukoliv IP adresu, doménové jméno, či jiný síťový identifikátor (obecně *entitu*) a získá všechny dostupné informace týkající se dané entity a související nějak s bezpečností. Je to například seznam všech bezpečnostních hlášení, kde je daná entita uvedená jako zdroj škodlivé aktivity, informace, zda je entita na nějakém blacklistu či jiné relevantní databázi, související informace z DNS, databází whois, geolokační data, nebo data ze služeb skenujících celý internet. Systém také umožňuje vyhledat všechny entity splňující zadaná kritéria a seřadit je podle různých atributů nebo podle skóre shrnujícího škodlivost dané entity (tedy FMP skóre navrženého v této práci). Dále systém poskytuje REST API pro snadnou integraci s jinými bezpečnostními systémy.

Za tímto portálem a API, které poskytují přístup k datům, je komplexní modulární systém, který slouží k získávání dat z nejrůznějších zdrojů, jejich zpracování a uložení

do databáze a k pravidelnému obnovování. Různé aspekty tohoto systémy jsou popsány v následujících podkapitolách.

6.1.1 Data, jejich získávání a ukládání

Datový model reputační databáze musí být velmi flexibilní. Lze totiž očekávat, že občas budou přidávány nové zdroje dat, případně bude získávání dat z některých stávajících zdrojů zastaveno, například protože daný zdroj přestane fungovat nebo je shledán příliš nespolehlivým. Kromě toho, oblast kybernetické bezpečnosti je celkově velmi proměnlivá a požadavky na systém a jím poskytovaná data se můžou časem měnit. Konkrétní množina ukládaných dat a jejich formát tedy není příliš stabilní. Zde jsou proto popsány jen trvale platné obecné principy práce s daty, na kterých je systém postaven.

Základní datovou jednotkou je *záznam entity* – strukturovaný záznam (možné reprezentovat např. ve formátu JSON) uchovávající všechny informace o konkrétní entitě. Entitou může být např. IP adresa, síťový prefix, číslo autonomního systému (ASN), doménové jméno apod. Každý záznam entity se skládá z množství *atributů* a může obsahovat i odkazy na záznamy jiných entit.

Zdroje dat ukládaných do záznamů lze rozdělit do dvou tříd – *primární* a *sekundární*. Primární datové zdroje jsou ty, které označují určité síťové entity jako škodlivé a na základě nichž jsou vytvářeny záznamy entit v databázi. Mohou to být hlášení z honeypotů, systémů IDS nebo systémů pro detekci anomálií (získané buď přímo, nebo např. přes systém pro sdílení hlášení, jako je Warden) nebo i jednoduché blacklisty, pokud je možné je získat celé[1].

Sekundární zdroje dat jsou ty, z nichž se získávají dodatečné informace o již známých entitách. Nejdříve je tedy vždy na základě hlášení z nějakého primárního zdroje vytvořen záznam entity se základními informacemi z tohoto hlášení a poté je záznam obohacen o další data ze sekundárních zdrojů. To zahrnuje například dotazy do whois databází, DNS dotazy pro zjištění doménového jména navázaného k IP adrese či naopak, geolokaci nebo dotazy do blacklistů a jiných seznamů, které nejsou použité jako primární zdroje (buď protože není možné získat celý seznam, nebo protože entity na seznamu nejsou nutně škodlivé, jako je tomu např. u seznamů tzv. dial-up rozsahů nebo proxy serverů). Také je možné získávat data z jiných databází, jako je například Shodan[2] nebo VirusTotal[3], a ukládat lokálně souhrn výsledků.

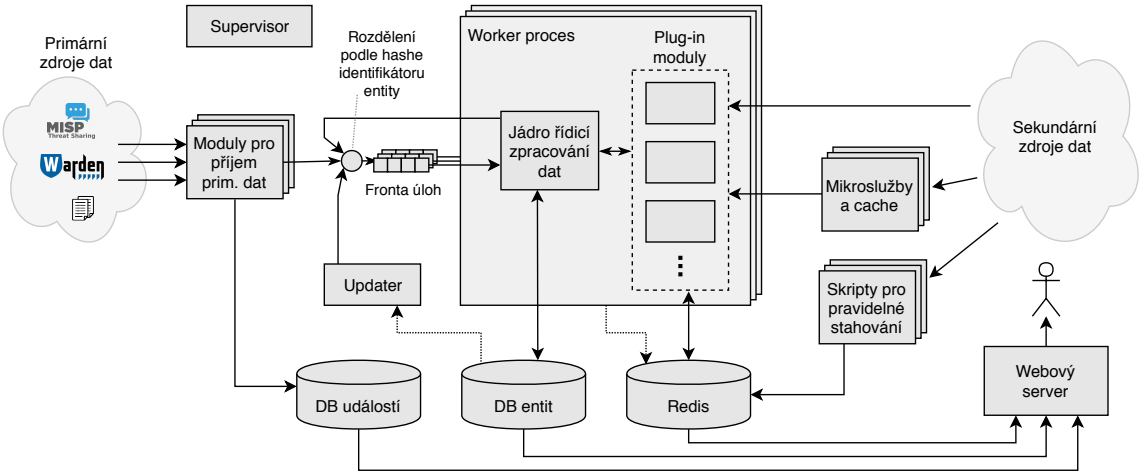
Primární data jsou obvykle získávána ve formě hlášení nebo jako záznamy převzaté z jiných systémů. V záznamech entit jsou o nich ukládána jen metadata, jako například počet hlášení za den. V některých případech je však vhodné ukládat i kopie původních hlášení, aby byl systém schopen poskytnout všechny potřebné detaily. Tato data jsou pak ukládána zvlášť v samostatné databázi a slouží pouze ke zobrazení uživateli, nejsou zpravidla dále zpracovávána. Sekundární data jsou obvykle ukládána přímo jako atributy záznamů entit.

Některé atributy nepochází přímo z primárních ani sekundárních zdrojů, ale jsou odvozeny lokálně z ostatních atributů. Příkladem může být přiřazování štítků podle převažujícího typu škodlivé aktivity určeného na základě metadat o přijatých hlášeních, nebo analýza doménového jména IP adresy, z nějž lze v některých případech pomocí jednoduchých regulárních výrazů odhadnout, zda jde o dynamicky přidělovanou adresu, NAT, VPN, či zda

[1] Někteří poskytovatelé blacklistů umožňují pouze dotázat se na konkrétní IP adresu či doménové jméno, nelze získat kompletní seznam. Takové zdroje pak nelze použít jako primární, tady vytvářet na základě nich nové záznamy entit.

[2] https://www.shodan.io/

[3] https://www.virustotal.com/



Obrázek 6.1: Architektura systému NERD

jde o běžné domácí DSL připojení, mobilní připojení, nebo například cloud server. Tyto atributy se nazývají *odvozené atributy*.

Systém také umožňuje pracovat s daty s jistým typem neurčitosti nebo s takovými, jejichž spolehlivost v čase klesá. Některé atributy tedy kromě samotné hodnoty obsahují také časovou známku jejího získání či poslední kontroly nebo míru důvěry v její správnost či přesnost. Tyto hodnoty pak mohou být zohledněny v algoritmech analyzujících data a samozřejmě jsou také zobrazovány uživateli.

U některých atributů, jejichž hodnota se může často měnit, jako je například přítomnost entity na různých blacklistech, je udržována nejen aktuální hodnota, ale i historie všech předchozích hodnot za určité období.

6.1.2 Shrnutí dat o entitách

Všechna uložená data o škodlivých entitách jsou uživateli přístupná ve všech podrobnostech. Pro rychlý přehled je však vhodné zároveň poskytnout i stručné shrnutí informací o každé entitě, například ve formě čísla či malé skupiny čísel a štítků, které popisují, jak velkou hrozbu a jakého typu daná entita představuje. Takové shrnutí umožňuje rychlé pochopení charakteristik dané entity uživatelem, případně ho lze použít i při vyhledávání a řazení.

Pro přiřazování štítků je implementován konfigurovatelný systém pravidel, které umožňují na základě existujících atributů v záznamu přiřadit libovolné štítky. Ty pak ukazují například převažující kategorii přijatých hlášení či výsledky heuristik odhadujících typ připojení.

Jako číselné ohodnocení entit je uvažováno FMP skóre navrhované v této práci. Systém NERD tedy v tomto kontextu slouží jako platforma pro získávání dat a zároveň jako místo, pro nějž je metoda primárně navrhována a kde bude nasazena, i když samozřejmě může najít uplatnění i v jiných systémech.

6.1.3 Architektura

Kvůli potřebě vysoké flexibility a snadné škálovatelnosti je systém NERD založen na principu modulární architektury. Skládá se z množství vzájemně spolupracujících komponent,

zpravidla pracujících jako samostatné procesy. Architektura systému je znázorněna na obrázku 6.1.

Součástí systému je několik databází. V té hlavní, *databázi entit*, jsou uloženy všechny záznamy entit, jak byly popsány v kapitole 6.1.1. Dále je zde samostatná databáze pro ukládání originálních dat z primárních zdrojů. To může být ve skutečnosti více různých databází, protože pro různé typy dat mohou být vhodné různé technologie. Nakonec je tu rychlá *in-memory key-value* databáze (Redis), která slouží pro ukládání různých, obvykle krátkodobých či rychle se měnících, pomocných dat, ke kterým je potřeba zajistit globální přístup z různých komponent. Také slouží k účelům logování různých operací.

Hlavní vstup systému je reprezentován množinou modulů pro příjem primárních dat. Ty přijímají zprávy, hlášení, či seznamy entit z externích zdrojů a do globální fronty vkládají *úlohy* požadující vytvoření nebo úpravu záznamů entit (tzv. *update requests*).

Tyto úlohy jsou zpracovávány jádrem systému – množinou pracovních procesů (*workers*). Ty aplikují požadované změny nad záznamy entit. Dále také obohacují záznamy o data ze sekundárních zdrojů a vypočítávají odvozené atributy. To je prováděno pomocí množiny zásuvných modulů, díky čemuž je snadné přidat, změnit či odstranit sekundární datové zdroje či pravidla pro odvozování atributů jen změnou těchto modulů.

Tyto worker procesy mohou běžet paralelně v libovolném množství, díky čemuž je výkon systému snadno škálovatelný. Úlohy jsou distribuovány podle hashe identifikátoru entity (každá úloha vždy pracuje jen s jedním záznamem entity), takže konkrétní záznam je vždy zpracováván stejným procesem, což umožňuje vyhnout se nutnosti zamykání záznamů a celkově pomáhá odstranit řadu problémů obvykle spojených se současným přístupem více procesů ke stejným datům.

Většina zásuvných modulů slouží k získávání dat z externích zdrojů. V závislosti na typu a dostupnosti těchto dat jsou používány tři základní způsoby jejich získávání: (*i*) přímé dotazy ze zásuvného modulu v NERD do rozhraní poskytovaného daným externím zdrojem (tímto způsobem jsou získávána např. whois data), (*ii*) prostřednictvím speciální mikroslužby či cache, která běží jako samostatná komponenta systému NERD a poskytuje snazší či efektivnější přístup k datům (příkladem může být rekurzivní DNS resolver použitý moduly pro zjišťování hostname a pro dotazy do DNS blacklistů, případně mikroslužba pro passive DNS přeposílající dotazy z modulu do několika passive DNS databází a slučující jejich výsledky), (*iii*) data jsou v pravidelných intervalech stahována, předzpracována a uložena lokálně, buď jako soubory nebo záznamy v Redis, modul pak využívá těchto lokálních dat (příkladem je geolokační databáze nebo některé blacklisty).

Pravidelné aktualizace dat v záznamech entit jsou řízeny komponentou *updater*. Ta kontroluje časové známky poslední aktualizace v záznamech a vydává úlohy (update requests) pro aktualizace daných atributů, pokud jsou starší než určitá doba.

Poslední komponentou systému je webový server. Jeho prostřednictvím jsou všechna data poskytována uživatelům přes webové grafické rozhraní nebo jiným systémům přes REST API.

Všechny komponenty systému jsou řízeny systémem pro správu procesů Supervisor[4].

6.1.4 Proces zpracování dat

Jak už bylo naznačeno výše, zpracování dat ve worker procesech je řízeno požadavky zvanými *update request*, které proces čte z globální fronty. Požadavek je zpráva s jasně daným formátem, která specifikuje množinu operací, které se mají provést s atributy nějakého kon-

[4] http://supervisord.org/ (jde o jiný nástroj než *nemea-supervisor* zmiňovaný v kap. 5.2.5)

krétního záznamu entity (např. u záznamu IP adresy X přičti k celkovému počtu hlášení 1).

Po přijetí požadavku worker proces načte odpovídající záznam z databáze (případně vytvoří nový, pokud v databázi ještě neexistuje) a aplikuje požadované změny. Tím však zpracování teprve začíná. Každý zásuvný modul může zaregistrovat tzv. *callback funkci*, která je zavolána vždy, když dojde k určité události. Tou může být buď speciální *pojmenovaná událost*, jako např. vytvoření nového záznamu, nebo změna hodnoty určitého atributu. Například DNS modul tak může na vytvoření nového záznamu IP adresy zaregistrovat funkci, která zjistí hostname přiřazené dané adrese a uloží ho do záznamu. To znamená změnu hodnoty atributu *hostname*, na což může reagovat jiná funkce, zaregistrovaná jiným modulem, například přiřazení určitých tagů podle klíčových slov nalezených v hostname. Tímto způsobem může jeden požadavek na změnu určitého atributu způsobit kaskádu řady dalších navazujících změn. Teprve když je celá tato kaskáda zpracována, upravený záznam je uložen zpět do databáze. Zpracování pak pokračuje načtením dalšího požadavku z globální fronty.

Každý takový požadavek, včetně všech navazujících změn prováděných callback funkcemi, pracuje vždy jen s jedním záznamem entity. Pokud je potřeba provést změny i v nějakém jiném záznamu (např. aktualizovat záznam o souvisejícím autonomním systému, pokud je přidána nová IP adresa), je vygenerován příslušný požadavek, vložen do hlavní fronty a zpracován později, potenciálně jiným worker procesem.

6.1.5 Implementační technologie

Tato kapitola stručně shrnuje, jaké technologie byly vybrány pro implementaci jednotlivých komponent systému NERD.

Většina komponent pro získávání a zpracovávání dat byla implementována specificky pro tento systém, a to jako programy v jazyce Python 3. Systém však využívá i řadu běžného software třetích stran.

Pro hlavní databázi uchovávající záznamy entit byla zvolena MongoDB. Důvodem je především různorodost ukládaných dat a již výše zmíněná potřeba flexibility. MongoDB je dokumentová databáze bez fixního schématu, takže umožňuje snadno přidávat nové atributy bez nutnosti úprav databázového schématu, nativně podporuje ukládání polí a podobjektů a umožňuje nad libovolnými atributy vytvářet indexy umožňující rychlé vyhledávání či řazení.

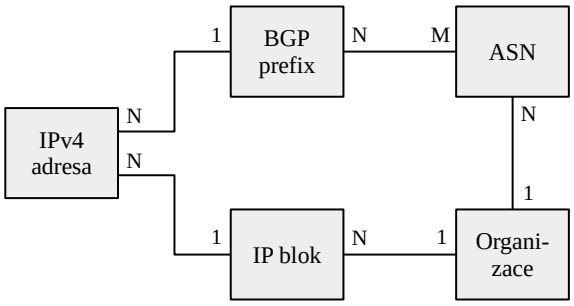
Databáze pro ukládání kopií primárních dat (např. hlášení ze systému Warden) nemá takové nároky na flexibilitu či speciální datové typy, takže je zde použita klasická SQL databáze, konkrétně PostgreSQL. Dále, jak už bylo zmíněno výše, je zde databáze Redis sloužící pro komunikaci mezi komponentami, jako rychlá cache a pro logovací účely.

Kromě databází je veškerá komunikace mezi komponentami zajišťována předáváním zpráv přes fronty RabbitMQ. Jde především o hlavní frontu úloh, ale RabbitMQ lze využít například i pro různé notifikace nebo pro speciální dotazy z webového rozhraní.

Jak již bylo zmíněno, pro správu běhu všech těchto komponent byl zvolen nástroj Supervisor.

Webový portál i API jsou implementovány pomocí Python frameworku Flask. Frontend používá jQuery a řadu javascriptových mikroknihoven.

V současnosti je celý systém provozován na jediném serveru, architektura je ale navržena tak, aby jednotlivé komponenty bylo možné distribuovat na více serverů, pokud by to



Obrázek 6.2: Typy entit v současnosti podporovaných systémem NERD a jejich vzájemné provázání

v budoucnu bylo potřeba. Jediná potřebná změna by bylo nahrazení nástroje Supervisor nějakým systémem schopným řídit takto distribuovaný systém.

6.2 Současný stav

V době odevzdání této práce (prosinec 2018) jsou všechny hlavní funkce systému NERD implementované, pouze množství datových zdrojů a podporovaných typů entit je zatím omezené. Systém je pilotně nasazen, běží jako služba dostupná komukoliv online[5]. Zdrojové kódy jsou dostupné jako open-source na platformě GitHub[6].

Hlavním podporovaným typem entit jsou IPv4 adresy[7]. Dále jsou zde záznamy pro BGP prefixy a čísla autonomních systémů (ASN) jakožto skupiny IP adres podle směrovacích informací. Podobně existují i záznamy o IP blocích a organizacích, které odpovídají záznamům regionálních registrátorů v tzv. whois databázích. Vztahy mezi jednotlivými typy podporovaných entit jsou znázorněny na obrázku 6.2.

V současnosti je implementován pouze jeden primární zdroj dat. Je jím Warden, systém pro sdílení bezpečnostních hlášení provozovaný organizací CESNET (viz kap. 2.3.2). Warden sbírá hlášení z více než 30 bezpečnostních monitorovacích nástrojů různých typů (honeypoty, IDS, systémy pro analýzu NetFlow dat, atd.) nasazených v několika velkých sítích. Podrobný popis těchto dat je uveden v kapitole 7. Metadaty, která jsou o hlášeních ukládána v systému NERD, je počet hlášení za den, a to zvlášť pro jednotlivé kategorie útoků a jednotlivé detektory.

V blízké době je plánováno zapojení více primárních zdrojů, konkrétně platforma MISP a vybrané zdroje z AlienVault Open Threat Exchange[8]. Také je v plánu využít jako primární zdroj některé blacklisty. Zatím se všechny používají jen jako sekundární zdroje, což znamená, že jsou dotazovány jen na adresy, které již jsou v databázi NERD (tzn. byly nahlášeny do systému Warden). Při použití blacklistu jako primárního zdroje budou přidány všechny IP adresy ze seznamu do databáze NERD, i když nebyly nahlášeny žádným jiným zdrojem.

Seznam implementovaných sekundárních zdrojů je delší. Zahrnuje zjišťování hostname pomocí PTR dotazu do DNS, přibližnou geografickou pozici (podle databáze GeoLite2 od

[5] https://nerd.cesnet.cz/

[6] https://github.com/CESNET/NERD/

[7] IPv6 adresy zatím podporovány nejsou, zejména proto, že útoky přes IPv6 jsou v dostupných zdrojích hlášeny jen minimálně

[8] https://otx.alienvault.com/

MaxMind[9]), nebo tzv. *origin* BGP prefixy a ASN a různá data z whois databází. Dále, jak už bylo zmíněno, jsou IP adresy vyhledávány na mnoha veřejně dostupných blacklistech (některé jsou pravidelně stahovány a ukládány do lokální cache, některé jsou dotazovány přes DNSBL[10]). Adresám jsou také přiřazovány různé příznaky (flagy), například podle nejčetnějších kategorií nahlášených událostí pro danou adresu, nebo na základě vyhledávání klíčových slov a regulárních výrazů v hostname.

Většina dat se tedy získává k záznamům o IP adresách, některé informace se ale získávají i k záznamům o ASN. Kromě názvu a popisu, převzatých z příslušné whois databáze, jde o tzv. *BGP rank* poskytovaný organizací CIRCL[11] a typ sítě (např. podniková či tranzitní) podle datové sady[12] poskytované organizací CAIDA.

6.3 Pravidla přístupu

Systém NERD shromažďuje data z mnoha různých zdrojů, většina je veřejná, ale k některým je přístup omezen. Některá data navíc mohou být považována za soukromá (např. data o spojeních provedených konkrétní IP adresou, která jsou součástí některých hlášení) nebo jinak citlivá (např. IP adresy honeypotů). Plný přístup ke všem datům je proto omezen pouze na důvěryhodné členy bezpečnostní komunity a vztahují se na ně pravidla omezující možnosti jejich použití.

Podmnožina dat je však zpřístupněna zcela volně komukoliv. Jsou to data, která nevyžadují žádnou speciální ochranu a pocházejí z veřejných zdrojů, ale také některá dostatečně agregovaná data z neveřejných zdrojů. I tato omezená podmnožina však poskytuje velké množství použitelných informací a systém tak lze v mnoha případech využívat ihned bez nutnosti žádat o plný přístup.

6.4 Statistiky

Pro získání lepší představy o množství a charakteru dat uchovávaných v systému NERD jsou v této kapitole uvedeny vybrané statistiky z jeho současného nasazení.

Každý záznam IP adresy je v databázi udržován po dobu 14 dnů od přijetí posledního hlášení označujícího danou adresu za škodlivou. Pokud tedy po tuto dobu žádné další hlášení nepřijde, je záznam smazán. Záznamy ostatních typů entit jsou uchovávány, dokud existuje nějaký jiný záznam, který na něj odkazuje (např. záznam IP bloku je tedy smazán v okamžiku, kdy již v databázi není žádná IP adresa patřící do tohoto bloku). S tímto nastavením je v databázi obvykle kolem 1,2 milionu záznamů IP adres. Obvyklé počty záznamů všech typů entit jsou uvedeny v tabulce 6.3.

Většina IP adres vykazuje škodlivé chování jen po krátkou dobu, což znamená, že většina záznamů je vytvořena jen kvůli jednomu nebo několika málo hlášením a po 14 dnech neaktivity je vymazána. Toto lze pozorovat na obrázku 6.4, který ukazuje histogram počtu hlášení za poslední měsíc pro každou IP adresu v databázi. Lze pozorovat, že u většiny adres je skutečně uloženo jen jedno nebo několik málo hlášení. Toto jsou pravděpodobně buď dynamicky přidělované adresy, nebo velmi málo aktivní útočníci. Obecně takové adresy nejsou považovány za velkou hrozbu (i když je nutné přihlédnout i k typu hlášené aktivity).

[9] https://dev.maxmind.com/geoip/geoip2/geolite2/

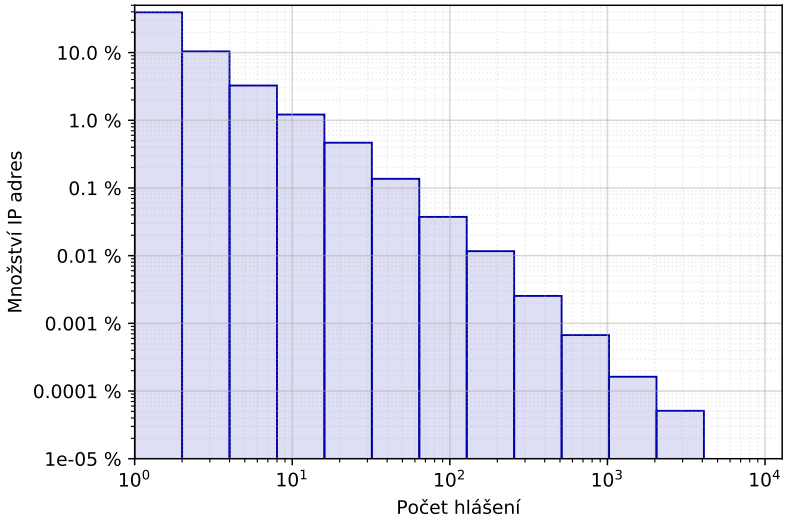
[10] *de facto* standard pro dotazování se blacklistů prostřednictvím protokolu DNS

[11] https://www.circl.lu/projects/bgpranking/

[12] "The CAIDA UCSD AS Classification Dataset", http://www.caida.org/data/as_classification.xml

Tabulka 6.3: Obvyklý počet záznamů v databázi NERD v první polovině roku 2018

Typ entity	Přibližný počet záznamů
IPv4 adresa	1,200,000
BGP prefix	115,000
ASN	16,000
IP blok	36,000
Organizace	15,000



Obrázek 6.4: Histogram počtu hlášení přijatých v posledním měsíci k jednotlivým IP adresám. V každém sloupci jsou započítány adresy s počtem hlášení spadajícím do rozsahu daného rozpětím sloupce na ose x (kvůli logaritmické stupnici tak sloupce ve skutečnosti nejsou stejně široké).

Na pravé straně grafu jsou pak adresy, které byly nahlášeny v posledním měsíci mnohokrát. Přestože je jich relativně málo, jsou zodpovědné za velké množství přijatých hlášení. Podrobnější pohled na tato data například ukázal, že jen asi 8400 (0.7 %) IP adres je nahlášeno více než 500krát v měsíci, hlášení o těchto nejaktivnějších adresách však tvoří celou polovinu ze všech přijatých hlášení. Podrobněji jsou tyto charakteristiky zdrojů škodlivého chování popisovány v kapitole 7.2.2.

Jiný příklad jednoduché statistické analýzy, kterou je možné provést nad daty ze systému NERD, využívá informace o typech útoků a geolokaci IP adres. Je totiž poměrně dobře známo, že z některých zemí pochází více škodlivého provozu na internetu než z jiných (viz kap. 3.1). Toto lze pozorovat i v systému NERD, navíc se ukazuje, že rozložení zdrojů se významně liší podle typu škodlivé aktivity. Tabulka 6.5 ukazuje 5 zemí s největším počtem IP adres nahlášených jako zdroj skenování nebo pokusů o neoprávněný přístup. Přestože některé země se vyskytují v obou seznamech, jejich relativní zastoupení se velmi liší. Tato problematika je podrobněji zkoumána v kapitole 7.2.1.

Dále lze například vytvořit statistiku výskytu IP adres na blacklistech. Systém NERD vyhledává každou IP adresu, která je uložena do databáze, na 46 veřejných blacklistech. Tabulka 6.6 ukazuje počty IP adres podle toho, na kolika blacklistech se adresa vyskytuje. Téměř 70 % adres se nachází alespoň na jednom z testovaných blacklistů, mnohé z nich

Tabulka 6.5: Země s největším počtem IP address nahlášených jako zdroje škodlivého provozu.

Skenování	
Země	IPs
Rusko	10.1 %
Čína	8.6 %
Indie	8.3 %
Vietnam	7.8 %
Mexiko	6.8 %

Neoprávněný přístup	
Country	IPs
Čína	26.6 %
USA	7.1 %
Rusko	6.8 %
Vietnam	6.5 %
Brazílie	4.9 %

Tabulka 6.6: Počet IP adres v databázi NERD vyskytujících se současně na daném množství blacklistů

Počet blacklistů	Počet IP adres	
0	374049	30.9 %
1	408314	33.7 %
2	315222	26.0 %
3	99011	8.2 %
4	10417	0.86 %
5	2297	0.19 %
6	973	0.080 %
7	304	0.025 %
8	54	0.004 %
9	19	0.002 %
10	4	0.000 %

na více, jde tedy o adresy, které jsou již obecně známé jako škodlivé. Na druhou stranu, 30 % adres nahlášených skrze systém Warden nebylo v době analýzy rozpoznáno žádným z poskytovatelů blacklistů a systém zde tedy poskytuje zcela nové informace.

Přestože je dotazováno vždy 46 blacklistů, žádná adresa není přítomna na více než 10 z nich. To je způsobeno tím, že část použitých blacklistů je úzce zaměřena na konkrétní typ škodlivých aktivit a obsahují tedy jen malé množství záznamů.

Data pro graf 6.4 a tabulky 6.5 a 6.6 byly získány v jeden konkrétní den v květnu 2018, nicméně tyto statistiky jsou poměrně stabilní a data z jiných dní jsou velmi podobná.

6.5 Shrnutí

Přestože je systém NERD především inženýrským dílem, je důležitý i z hlediska výzkumných aktivit, a to zejména jako zdroj dat pro další výzkum. Data ze systému Warden dokáže vhodným způsobem předzpracovat (agreguje je podle zdrojové IP adresy) a především je obohatí o data z dalších zdrojů. Zároveň systém NERD poskytuje platformu pro implementaci navržené metody výpočtu FMP skóre a její ověření v reálném prostředí.

Kapitola 7

Použitá data a jejich charakteristiky

Jak již bylo zmíněno, pro ověření obecného principu určování reputace síťových entit, představeného v kapitole 4, je v následujících kapitolách navržena a ověřena konkrétní varianta výpočtu FMP skóre pro ohodnocování IP adres. Pro návrh takové varianty je třeba nejprve znát charakteristiky chování škodlivých IP adres a typické vlastnosti souvisejících hlášení.

Cílem této kapitoly tedy je: (*i*) uvést, jaká konkrétní datová sada byla použita dále v této práci, a (*ii*) prostřednictvím analýzy této datové sady zjistit základní charakteristiky chování zdrojů škodlivého provozu.

Analýza dat má za cíl ověřit, že charakteristiky dat nasbíraných a použitých v této práci přibližně odpovídají těm uváděným v existující literatuře, jako je například nerovnoměrné rozložení zdrojů škodlivého provozu v prostoru, skutečnost, že velká část zdrojů se v hlášeních objevuje jen po velmi krátkou dobu, nebo to, že se tyto charakteristiky mohou významně lišit podle typu škodlivé činnosti (viz kap. 3.1). Zároveň by znalost konkrétních hodnot některých statistik měla pomoci při návrhu parametrů predikční metody v následující kapitole.

Tato kapitola částečně vychází z analýzy dat ze systému Warden provedené autorem v roce 2015 a popsané v technické zprávě [63]. Zde je však tato analýza zopakována nad novějšími daty, konkrétně z druhé poloviny roku 2017. Stejná data jsou pak použita i v kapitole 9 pro vyhodnocení metody určování reputace IP adres.

7.1 Datová sada

Základem pro metodu hodnocení reputace jsou data typu hlášení o bezpečnostních událostech. Data tohoto typu použitá v této práci pochází ze systému pro sdílení hlášení Warden, provozovaného sdružením CESNET. Do tohoto systému jsou svedena hlášení z více než 30 detekčních nástrojů různých druhů[1]. Zdroje dat jsou umístěny jak přímo v síti CESNET a v kampusových sítích připojených univerzit, tak i u několika komerčních a zahraničních partnerů. Systém Warden celkově zpracuje kolem 1,7 milionu hlášení za den (cca 20 za sekundu).

Přibližně polovina těchto hlášení pochází z jednoho honeypotu pozorujícího velký adresní rozsah a generujícího obrovská množství hlášení o skenování. Hned druhým největším

[1] Některé detektory však v době sběru dat nebyly dostatečně spolehlivé a fungovaly v tzv. testovacím režimu, jejich data nejsou v této práci použita

Tabulka 7.1: Počet hlášení, nahlášených IP adres a počet detektorů, které hlášení vygenerovaly, podle kategorie.

	září			říjen			listopad		
Kategorie	hlášení	adres	det.	hlášení	adres	det.	hlášení	adres	det.
Recon.Scanning	40 158 463	1 758 801	8	51 326 622	1 801 762	10	54 519 766	2 324 460	7
Attempt.Login	2 864 168	103 402	4	2 566 588	92 512	6	2 396 084	80 132	5
Attempt.Exploit	411 223	20 664	4	130 221	20 946	5	56 447	24 609	3
Intrusion.Botnet	144 861	142	4	263 621	304	4	93 883	610	4
Availability.(D)DoS	10 828	130	3	58 676	117	3	33 566	93	3
Abusive.Spam	10 143	5 802	2	7 469	1 974	2	6 670	1 507	2
Vulnerable.Config	3 867	224	2	3 612	257	2	3 190	221	2
Anomaly.Traffic	1 269	58	2	3 138	75	2	2 414	84	3

přispěvatelem jsou pak detektory vyvinuté v rámci frameworku NEMEA (viz kap. 5), které v souhrnu generují asi 1/4 všech hlášení. V jejich případě jde často o hlášení závažnějších a zajímavějších událostí, než jsou pouhé pokusy o navázání TCP spojení, které hlásí zmíněný honeypot. Například hlášení o DDoS útocích pocházejí výhradně z detektorů systému NEMEA a v případě neoprávněných pokusů o vzdálený přístup jich pochází z NEMEA asi 95 %.

Pro tuto práci byla použita data ze tří měsíců – od 1. 9. 2017 do 30. 11. 2017. Každý měsíc je v této kapitole analyzován samostatně. Porovnáním výsledků z různých časových období lze určit, jak jsou zjištěné charakteristiky stabilní v čase.

V tabulce 7.1 jsou uvedeny počty hlášení, počty nahlášených zdrojových IP adres, a počty detektorů pro jednotlivé kategorie hlášení. Kategorie odpovídají taxonomii formátu IDEA[2], který je pro předávání hlášení v systému Warden použit. Zde je uveden stručný popis kategorií, včetně informace o tom, jak jsou aktivity dané kategorie obvykle detekovány (dle autorových osobních znalostí systému Warden a připojených detektorů):

- Recon.Scanning – skenování portů a jiné průzkumné aktivity (např. ICMP echo). Detekovány zpravidla jako pokusy o připojení k honeypotu, či jako neobvyklé množství neúspěšných pokusů o připojení detekované při analýze flow dat.
- Attempt.Login – neočekávaný/neoprávněný pokus o přihlášení k nějaké autentizované službě. Obvykle jde o pokusy o přihlášení se k honeypotům, nejčastěji přes protokol SSH, některé detekce jsou však založeny i na analýze flow dat a mohou tak odhalit i útoky proti skutečným serverům.
- Attempt.Exploit – pokusy o ovládnutí cílového systému pomocí nějaké zranitelnosti. Nejčastěji jde o hlášení HTTP honeypotů a pokusy o přístup na specifická URL známých frameworků a redakčních systémů.
- Intrusion.Botnet – hlášení o zařízeních, které jsou pravděpodobně součástí botnetu. Tato hlášení pochází od externích služeb (např. Shadowserver) a týkají se pouze IP adres náležejících pod síť CESNET.
- Availability.(D)DoS – útoky odepření služby (jednoduché i distribuované, tj. DoS i DDoS). Tyto útoky jsou detekovány na základě analýzy flow dat, a to jak jednoduché útoky záplavou paketů (např. SYN flood), tak útoky odražené a zesílené (např. DNS

[2] https://idea.cesnet.cz/en/classifications

amplifikace). V případě odražených útoků je jako zdroj hlášen server zneužitý k odrazu (viz kap. 5.4.3).

- Abusive.Spam – hlášení o IP adresách odesílajících spam. Hlášeno především jedním z poštovních serverů sdružení CESNET, ale i z jistého externího zdroje.
- Vulnerable.Config – hlášení od externích služeb o adresách v síti CESNET, upozorňující na nevhodné nastavení některých služeb. Zpravidla jsou takto hlášeny otevřené DNS servery zneužitelné k amplifikačním útokům.
- Anomaly.Traffic – obecné anomálie v síťovém provozu, u kterých se nepodařilo s jistotou určit konkrétní typ události. Generováno na základě analýzy flow dat.

Z tabulky 7.1 je zřejmé, že zdaleka největší podíl hlášení je typu Recon.Scanning. To je dáno jednak tím, že různé typy skenování jsou na internetu skutečně velmi rozšířeným jevem, zároveň je typické skenování poměrně snadno detekovatelné. Skenování samo o sobě nepředstavuje hrozbu, v některých případech je dokonce prováděno samotnými administrátory sítě či globálními službami systematicky skenujícími internet pro výzkumné účely (např. Shodan[3] či Censys[4]). Ve většině případů však jde spíše o vyhledávání potenciálně zranitelných služeb, buď přímo útočníkem, nebo jako součást automatického šíření malware. Například Panjwani et al. [119] uvádí, že téměř 50 % útoků předchází skenování sítě. Proto má smysl skenování detekovat a hlášeními se zabývat.

Dále je v datové sadě poměrně velké množství hlášení typu Attempt.Login a Attempt.Exploit, tedy různých pokusů o neoprávněný přístup či kompromitaci systému.

Ostatní kategorie jsou v dostupných datech zastoupeny výrazně méně, zejména počet nahlášených adres je velmi malý, maximálně v řádu stovek, což není pro kvalitní statistiky dostatečné. Výjimkou je kategorie Abusive.Spam, ta však není pro analýzu chování zdrojů škodlivého chování příliš reprezentativní, protože většina hlášení pochází jen z jediného zdroje (poštovního serveru) a navíc je každá adresa z tohoto zdroje nahlášena vždy jen jednou (pak je totiž zařazena na blacklist a server již další spojení z dané adresy odmítá dřív, než může dojít k detekci a nahlášení dalšího spamu).

Z těchto důvodů jsou dále použita pouze hlášení kategorií Recon.Scanning, Attempt.Login a Attempt.Exploit.

Pro zachování jednoduchosti a stručnosti, pokud se dále v této práci zmiňují *škodlivé aktivity* či *útoky*, myslí se tím události jakékoliv kategorie, včetně skenování.

7.2 Analýza dat

Hlášení v této datové sadě byla analyzována za účelem odhalit vybrané charakteristiky zdrojů útoků, konkrétně jejich geografické rozložení, vlastnosti jejich chování v čase, a ověření možnosti predikce budoucích útoků. Těmto oblastem se postupně věnují následující tři podkapitoly.

Před samotnou analýzou je vhodné poznamenat, že jedno hlášení nemusí nutně odpovídat jednomu útoku – jeden útok může být nahlášen prostřednictvím více hlášení. Je zřejmé, že útok může být ve stejnou dobu pozorován a nahlášen různými detektory. Také ale může být stejný útok nahlášen mnohokrát jen jedním detektorem. To se obvykle stává, pokud detektor analyzuje data v pevných časových intervalech a doba trvání útoku zasahuje

[3] https://www.shodan.io/
[4] https://www.censys.io/

do více takových intervalů. Příkladem může být typický analyzátor flow dat, který data rozděluje do pětiminutových intervalů a každý interval vyhodnocuje zvlášť. Při dlouhotrvajícím útoku je tak po celou dobu jeho trvání každých 5 minut vygenerováno nové hlášení popisující příslušnou část útoku. Jiným typickým příkladem je honeypot, který zapisuje data o příchozích spojeních do log souboru, a skript, který v pravidelných intervalech tento soubor čte a pro každou zdrojovou adresu hlásí počet spojení a jejich parametry.

Takové detektory tedy mohou vygenerovat například hlášení s časem začátku t_0 a časem konce $t_0 + w$ a druhé hlášení s časem začátku $t_0 + w$ a časem konce $t_0 + 2w$, kde w je použitá délka časového intervalu. Navíc čas začátku druhého hlášení nemusí být přesně stejný jako čas konce prvního, může mezi nimi být (a často je) mezera, odpovídající času, kdy nebyla zachycena žádná aktivita útočníka. To může být dáno tím, že útočník skutečně nechává časové odstupy mezi jednotlivými pokusy (např. při hádání hesel), nebo sice provádí svou aktivitu neustále, ale na různé cíle a jen část je detekována (např. při skenování s náhodným výběrem cílů).

Jedním z možných řešení tohoto problému je více hlášení stejného typu, se stejnými adresami a blízko po sobě v čase sloučit do jednoho. Problém je ovšem určit, jak přesně daleko v čase mohou hlášení být, aby byly ještě považovány za jednu událost. Je tedy třeba nastavit mez, určující zda v případě dvou po sobě jdoucích hlášení jde jen o dvě hlášení jedné události, nebo jde o dvě různé události stejného typu. Takovou mez lze určit různě, žádnou konkrétní hodnotu nelze označit za objektivně správnou či nejlepší. Zároveň však platí, že její nastavení může výrazně ovlivnit výsledky jakékoliv analýzy takto agregovaných dat.

Počet hlášení přijatých k dané adrese je tedy velmi problematická metrika. V následujících analýzách se tedy raději pracuje především s informací, zda daná adresa byla či nebyla nahlášena v určitém delším časovém intervalu (jeden den či celý měsíc) a záměrně je ignorován počet hlášení, která byla za danou dobu přijata.

7.2.1 Geografické rozložení zdrojů škodlivého provozu

Pro každou IP adresu v datové sadě byla zjištěna její pravděpodobná geografická pozice na úrovni země, a to pomocí volně dostupné databáze GeoLite2 od společnosti MaxMind[5]. V grafech na obrázku 7.2 je znázorněno 10 zemí s nejvyšším počtem nahlášených adres pro jednotlivé typy útoků. Tři vnitřní kruhy vždy znázorňují poměr nejčastějších zemí v jednotlivých měsících (ve směru zevnitř ven: září, říjen, listopad), vnější kruh pak průměr přes všechny tři měsíce.

Při porovnání jednotlivých grafů si můžeme všimnout, že jsou zde značné podobnosti, například Čína a Brazílie jsou vždy na prvních dvou příčkách. Na dalších pak vždy najdeme v první desítce Indii, Rusko, USA a Turecko, jejich pořadí a hlavně relativní zastoupení se už však výrazně liší.

Obecně tedy můžeme říci, že geografické rozložení zdrojů síťových útoků se liší podle typu útoku. Je to pravděpodobně tím, že různé typy útoků jsou často prováděny odlišnými typy malware a zařízení používaná v některých zemích mohou být k určitým typům malware náchylnější než jiná (příkladem může být zneužití zranitelnosti v protokolu TR-069 variantou botnetu Mirai, které se dotklo domácích routerů dodávaných některými ISP, z nichž zdaleka největším byl Deutsche Telecom, a napadena tak byla zejména zařízení v Německu [120]). Také je známo, že některé typy malware si cíle nevybírají náhodně, ale specificky podle určitých kritérií, a region či konkrétní země může být jedním nich (příkla-

[5] https://dev.maxmind.com/geoip/geoip2/geolite2/

Obrázek 7.2: Deset zemí s největším počtem škodlivých IP adres pro různé typy útoků.

dem může být ransomware Petya, který byl zřejmě cílen především na Ukrajinu, více než 75 % nakažených cílů se nacházelo právě tam [121]).

Porovnání dat z jednotlivých měsíců odhalí, že sice v čase dochází k jistým změnám v geografickém rozložení škodlivých IP adres, ve většině případů jsou to však změny jen malé a rozložení je poměrně stabilní. Výjimkou je prudký nárůst počtu skenujících adres z Egypta v listopadu a postupný nárůst počtu adres pokoušejících se o zneužití zranitelnosti (kategorie *exploit*) z Brazílie. V obou případech jde o navýšení absolutního počtu nahlášených adres z dané země. V případě Egypta jde o zvýšení z přibližně 6 000 adres v září a říjnu na 230 000 v listopadu, přičemž absolutní počty u ostatních zemí se nijak významně neliší (většinou jen mírně stoupají). V případě Brazílie a útoků typu exploit pak počet adres stoupá ze 4 000 v září až na více než 15 000 (63 % celkového počtu) v listopadu. Přitom celkový počet adres nahlášených s tímto typem útoku stoupl jen o necelé 4 000, zároveň totiž došlo k poklesu počtu adres z jiných zemí, zejména Číny, Indie a Ruska.

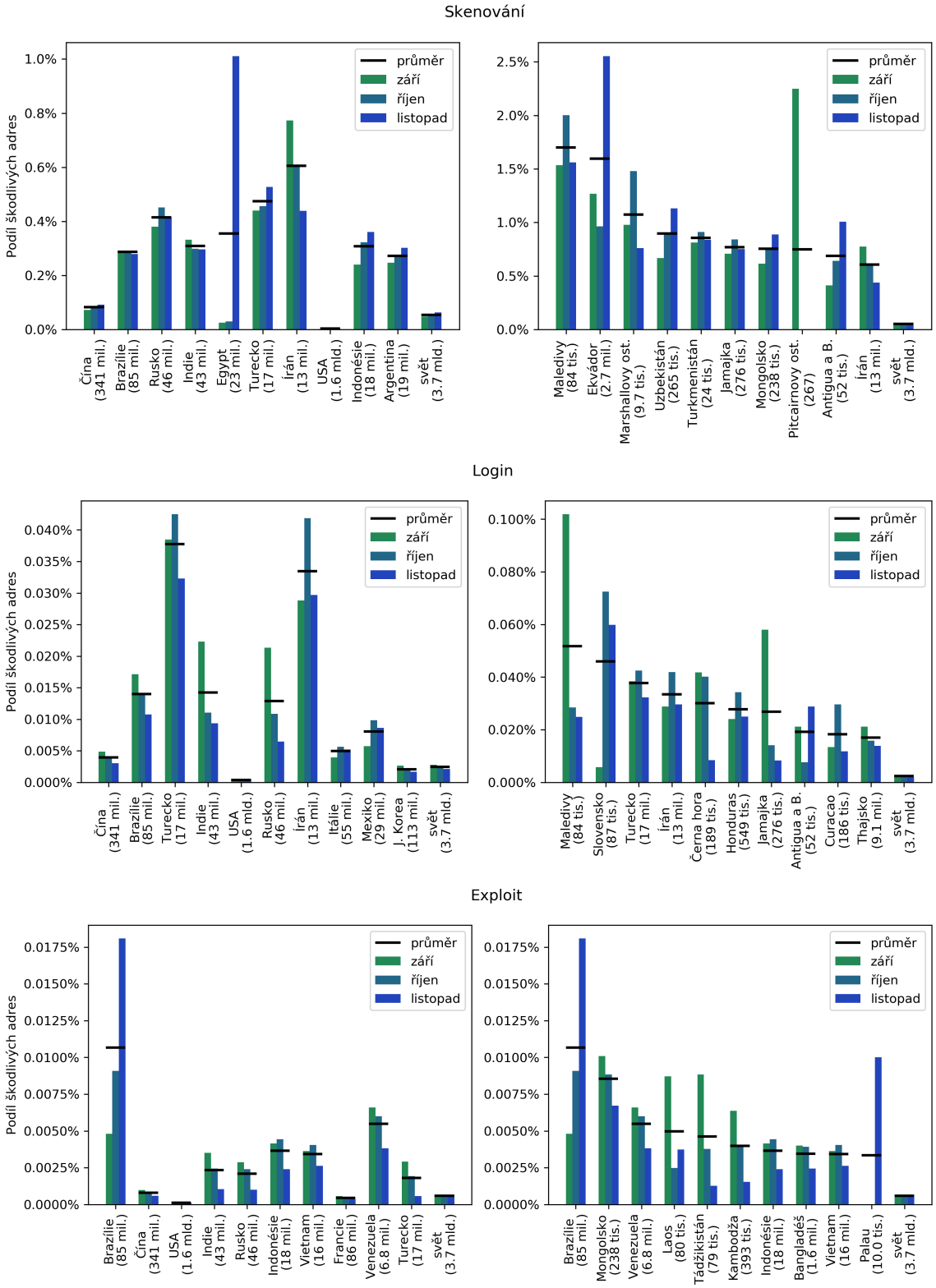
Příčiny těchto anomálií bohužel nejsou z informací dostupných v hlášeních patrné a důkladné pátrání po informacích vysvětlujících konkrétní výkyvy ve statistikách není cílem této práce. Jako shrnutí lze uvést, že ve střednědobém horizontu několika měsíců je rozdělení zdrojů útoků podle zemí poměrně stabilní, ačkoliv se mohou objevovat anomálie, kdy jedna země začne být náhle výrazně aktivnější než dříve.

Porovnáme-li však výše uvedené seznamy zemí s obdobnými seznamy z analýzy dat z roku 2015 [63], kde například prvních 5 zemí pro kategorii skenování jsou Čína, Rusko, USA, Španělsko a Brazílie, je zřejmé, že v dlouhodobém horizontu se geografické rozdělení škodlivých IP adres může měnit velmi výrazně.

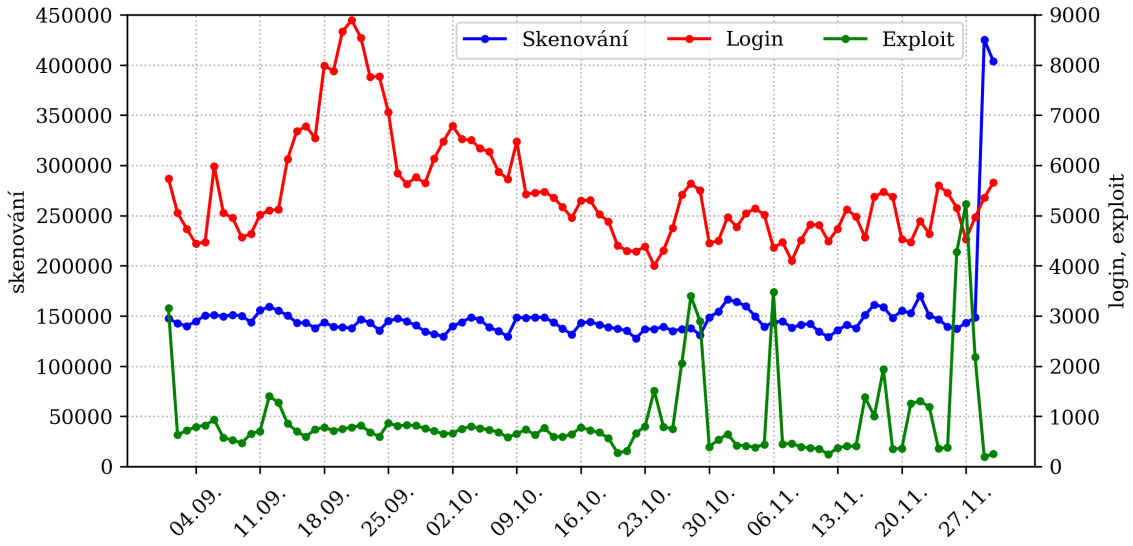
Všechny výše uvedené statistiky vycházely z absolutního počtu škodlivých IP adres v dané zemi. Logicky proto v grafech převažují poměrně velké země s velkým množstvím aktivních IP adres. Zároveň je však zřejmé, že rozložení škodlivých adres do zemí přímo neodpovídá velikostem adresních rozsahů (např. USA mají zdaleka největší adresní prostor [122], ale u žádné kategorie útoků nejsou na první pozici v počtu škodlivých adres, a např. Japonsko se třetím největším adresním prostorem vůbec není v první desítce).

Grafy na obrázku 7.3 proto ukazují relativní množství nahlášených IP adres vzhledem k celkovému množství IP adres nacházejících se dle geolokační databáze v dané zemi. Jinými slovy, hodnota na svislé ose odpovídá pravděpodobnosti, že náhodně vybraná IP adresa z určité země byla v daném měsíci nahlášena jako škodlivá. Pro porovnání je v posledním sloupci vždy uveden zlomek škodlivých adres pro celý svět, tedy celkový počet škodlivých adres v použité datové sadě vydělený součtem velikostí rozsahů všech zemí (tj. asi 3.7 mld. adres, zbytek tvoří adresy rezervované pro speciální účely).

Grafy vlevo ukazují zlomek škodlivých adres vždy pro stejných deset zemí jako na obrázku 7.2, tedy těch s největším absolutním počtem škodlivých adres. Je vidět, že při zahrnutí velikosti adresního prostoru jsou mezi těmito zeměmi obrovské rozdíly. Např. v již zmíněném Egyptě bylo v listopadu celé 1 % tamních IP adres detekováno jako zdroj skenování. A například Čína, která je v absolutních počtech škodlivých adres pro kategorie skenování a login na prvním místě, je v relativních počtech jen mírně nad celosvětovým průměrem. Celkově je nad celosvětovým průměrem většina těchto zemí. Výjimkou jsou Spojené státy u nichž je relativní zastoupení nahlášených škodlivých adres jen velmi malé (0,0037 %, 0,00038 % a 0,00011 % pro jednotlivé typy útoků). To může být zčásti dáno zdaleka největším přiděleným rozsahem (35,9% celkového IPv4 prostoru), jehož část pravděpodobně není využita, případně je využita pro různé servery či datová centra, jež jsou méně náchylná k napadení malwarem než např. domácí sítě. Spekulace nad důvody tohoto rozložení však nejsou cílem této práce.



Obrázek 7.3: Podíl nahlášených škodlivých IP adres vzhledem k celkové velikosti adresního prostoru dané země (uveden v závorce vedle názvu země). Vlevo: 10 zemí s největším absolutním počtem nahlášených adres. Vpravo: 10 zemí s největším relativním zastoupením škodlivých adres.



Obrázek 7.4: Počet unikátních adres nahlášených jako zdroj daného typu útoku v jednotlivých dnech (rok 2017). Pro data o skenování platí osa y vlevo, pro ostatní kategorie útoků osa vpravo.

Na grafech vpravo je uvedeno vždy deset zemí s největším podílem škodlivých adres. Dle očekávání je tu několik velmi malých zemí (např. Pitcairnovy ostrovy jsou zámořským územím Velké Británie s jen asi 50 stálými obyvateli [123]), v jejichž případě je zařazení mezi první desítku spíše dílem náhody, neboť vzhledem k jejich velmi malému IP rozsahu stačí jen málo škodlivých adres k dosažení vysokého poměru. Jsou zde však i poměrně velké země, jejichž zařazení v první desítce skutečně vypovídá o výrazně vyšší koncentraci zdrojů škodlivého provozu, než je obvyklé.

Celkově tedy můžeme konstatovat, že relativní zastoupení škodlivých IP adres v adresním prostoru se mezi jednotlivými zeměmi velmi liší, a to i o několik řádů. Tato metrika tedy může dobře sloužit jako jeden z parametrů pro predikci budoucích útoků.

Zároveň však z porovnání sloupců z jednotlivých měsíců vyplývá, že přinejmenším u některých zemí se tento poměr v čase rychle mění. Způsob výpočtu statistik pro predikci by s tím tedy měl počítat a brát v úvahu jen data z nepříliš vzdálené minulosti.

7.2.2 Korelace hlášení v čase

Pravděpodobně nejdůležitějšími informacemi pro vyhodnocení reputace síťové entity a pro predikci případných budoucích škodlivých aktivit jsou data o jejím předchozím chování. Intuitivně lze předpokládat, že entity škodící v nedávné minulosti budou v budoucnu zdrojem dalších útoků pravděpodobněji, než ostatní entity. V této podkapitole je tedy provedena základní analýza korelací hlášení v čase, zaměřená především na to, zda, jak často a jak dlouho se v hlášeních opakují stejné zdroje útoků.

Pro studium těchto charakteristik byla hlášení v datové sadě rozdělena podle dne detekce, tzn. pro každý den v uvedených třech měsících byl vytvořen seznam IP adres, které byly v tento den nahlášeny jako zdroj daného typu útoku.

Na obrázku 7.4 je zobrazen časový průběh těchto počtů adres za den. Lze pozorovat, že se tyto počty v čase mění. U počtu skenujících adres je dokonce patrný týdenní vzor

– vždy v neděli dojde k mírnému poklesu počtu nahlášených adres, v některých týdnech je pokles patrný už v dřívějších dnech (svislé tečkované čáry v grafu označují pondělí). To lze vysvětlit tím, že část ze skenujících zařízení mohou být malwarem nakažené pracovní stanice, které jsou mimo pracovní dny často vypnuté (tento vzor je podobný typickým týdenním průběhům celkového objemu provozu v sítích, jen je v tomto případě mnohem méně výrazný). Celkově je však množství skenujících adres poměrně stabilní. Jedinou výjimkou je velmi výrazný nárůst v posledních dvou dnech. Ten je způsoben nebývalým množstvím skenujících adres z Egypta a jde tedy o anomálii již popisovanou v předchozí podkapitole (náhled na data z následujícího měsíce, zde jinak neanalyzovaná, prozrazuje, že počet těchto adres postupně klesá a asi po týdnu se vrátí do normálu kolem 150 tisíc adres za den).

Počty adres zodpovědných za ostatní typy útoků se mění spíš náhodně a žádný pravidelný vzor zde patrný není. Celkově jsou tyto počty mnohem méně stabilní, než je tomu u skenování. Zejména u hlášení typu exploit je pak vidět řada anomálií (pátrání po jejich původu však není pro tuto práci podstatné). Celkově je větší proměnlivost těchto dat jistě zčásti daná i tím, že adres je u těchto kategorií až od dva řády méně než u skenování, takže i méně významná událost může statistiku výrazně ovlivnit.

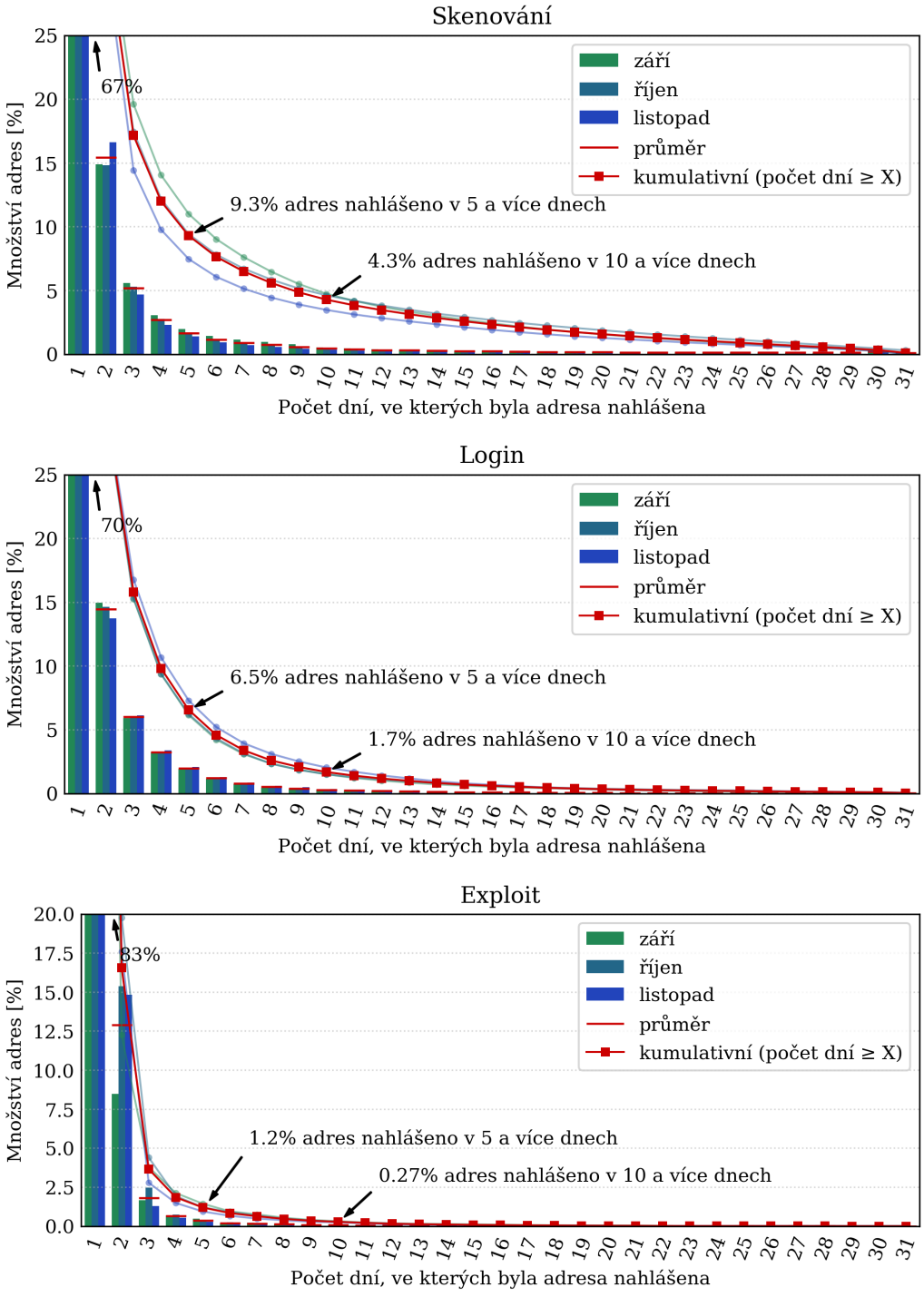
Pro zjištění, zda je běžné, že je stejná adresa hlášena opakovaně po delší dobu, bylo pro každou adresu spočítáno, v kolika dnech v určitém měsíci byla nahlášena jako škodlivá. Tyto počty jsou uvedeny na obrázku 7.5. Na vodorovné ose je uveden počet dnů v měsíci, výška každého sloupce ukazuje, kolik adres se v daném měsíci vyskytlo *právě* v tolika dnech. Čára označená jako „kumulativní“ hodnota pak ukazuje, jaké procento adres bylo nahlášeno v *alespoň* tolika dnech (jde tedy o součet výšky sloupce na dané pozici a všech sloupců napravo).

Ve všech případech je většina adres nahlášena jen v jediném dni, viz první sloupec v každém grafu (67 %, 80 %, resp. 83 % adres pro hlášení typu skenování, login a exploit). Navíc další analýzou lze zjistit, že velká část adres je nahlášena jen jedním jediným hlášením, konkrétně 44 %, 20 %, 57 % adres pro jednotlivé typy hlášení. Útoky z většiny adres tedy netrvají dlouho – buď proto, že jsou tyto adresy skutečně škodlivé jen krátce (buď je příslušné nakažené zařízení rychle opraveno nebo, a to je pravděpodobnější, jde např. o mobilní zařízení, které se pohybuje po různých IP adresách), nebo může jít o útoky nízké intenzity, navíc rozptýlené přes velké množství cílových sítí, takže sítě pokryté detektory přispívajícími do systému Warden nevidí útok vícekrát než jednou za měsíc. Tato zjištění jsou mimochodem v souladu s výsledky prezentovanými např. v [53] či [81], kde byla sice zkoumána data z jiných zdrojů, ale i v těch bylo zjištěno, že velká část adres se v hlášeních vůbec neopakuje, případně jen po krátkou dobu.

Na pravé straně grafů jsou pak adresy, které jsou hlášeny velmi často. Například v případě skenování je v průměru 9,3 % adres nahlášeno v pěti či více dnech v měsíci. Takové adresy již lze považovat za dlouhodobě aktivní útočníky a lze u nich očekávat podstatně vyšší pravděpodobnost dalších hlášení.

Podrobnější analýza dat navíc odhalí, že těchto 9,3 % (180 tis.) skenujících adres je zodpovědných za 65 % všech hlášení typu skenování. Podobně zatímco jen k 6,5 % (5 800) adres byl nahlášen pokus o přihlášení (login) v pěti a více dnech, tyto adresy jsou zodpovědné za 60 % příslušných hlášení. V případě hlášení typu exploit je pak pouhých 1,2 % (265) adres zodpovědných za rovných 50 % hlášení (vždy jde o průměr za všechny tři měsíce).

Přestože k těmto hodnotám je třeba přistupovat opatrně, protože počet hlášení je do značené míry závislý na nastavení detektorů a neodpovídá přímo počtu útoků (jak bylo diskutováno na začátku kap. 7.2), naznačuje to, že zablokováním relativně malého počtu nejvíce aktivních škodlivých IP adres lze zabránit velké části útoků.



Obrázek 7.5: Rozložení množství adres podle počtu dní v měsíci, v nich byly nahlášeny jako škodlivé.

Porovnání jednotlivých grafů na obrázku 7.5 odhalí, že míra opakovaných hlášení o stejné adrese se liší podle typu útoku. Zatímco hlášení typu skenování a login vykazují podobný podíl adres nahlášených jen v jednom dni, čára znázorňující kumulativní hodnoty leží u skenování podstatně výš. To znamená, že pokud je nějaká skenující adresa hlášena opakovaně, pak její aktivita trvá v průměru déle, než u hlášení typu login. Hlášení typu exploit pak vykazují zdaleka nejmenší míru opakování – jen velmi málo adres je nahlášeno ve více než několika málo dnech.

U typů skenování a login je v grafech zcela napravo několik sloupců, které jsou sice velmi malé, přesto však ne nulové. Jde o počty adres, které byly detekovány ve všech nebo téměř všech dnech v měsíci a jde tedy o permanentně aktivní zdroje škodlivých aktivit. Ačkoliv jde v celkovém množství jen o zlomky procent, např. v případě skenování se každý z několika posledních sloupců pohybuje v řádu jednotek tisíců adres. U hlášení typu login jde v každém sloupci o desítky adres, u typu exploit je pak většina sloupců napravo od hodnoty 20 nulová.

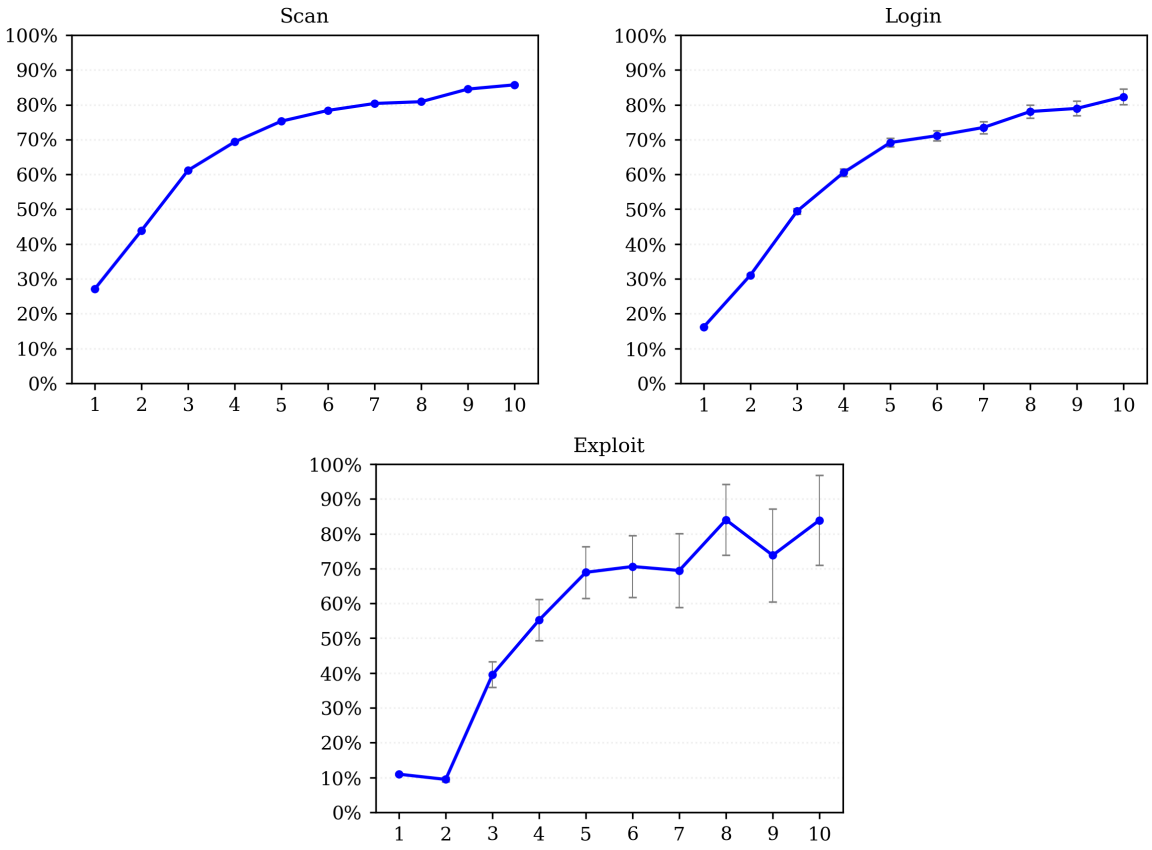
Zejména u skenování tedy existuje velké množství adres, které jsou aktivní prakticky neustále. Protože existuje řada služeb, které pravidelně skenují celý internet pro výzkumné a jiné neškodné účely, jako např. ShadowServer, Shodan či Censys.io, nabízí se vysvětlení, že mnoho z těchto skenujících adres může být součástí takových služeb. Pro ověření této možnosti lze využít faktu, že je zvykem, že takové služby samy sebe identifikují pomocí srozumitelných DNS záznamů typu PTR přiřazených jednotlivým skenujícím adresám (např. `scan-08.shadowserver.org`), obvykle navíc na stejném doménovém jméně běží i HTTP server poskytující webovou stránku s podrobnějším vysvětlením (pro konkrétní doporučení viz kap. 5 v [124]).

Pro odhad, kolik z permanentně skenujících adres tedy patří pod takové služby, byl vybrán náhodný vzorek 2000 adres ze všech IP adres nahlášených jako zdroj skenování ve více než 25 dnech v měsíci (těch je celkem 17 tisíc). Pomocí reverzních DNS dotazů byla k těmto adresám zjištěna jejich doménová jména a ta byla ručně analyzována. Výsledkem je, že pouze asi 1,3 % z těchto adres patří dle DNS záznamu pod nějakou legitimní skenující službu. Většina ostatních adres má generický DNS záznam (např. `198-51-100-12.dyn.provider.com`) napovídající, že jde pravděpodobně o běžnou domácí DSL přípojku, případně nemá DNS záznam žádný. Přestože i některé z těchto adres mohou být neškodné, je zřejmé, že ve většině případů jde spíš o stanice nakažené malwarem.

7.2.3 Možnosti predikce útoků

V předchozí podkapitole bylo ukázáno, že mnoho adres je hlášeno opakovaně, a to často mnohokrát po delší dobu. To odpovídá předpokladu uvedenému již v kapitole 4, že informace o předchozím škodlivém chování by mohla být použitelná pro predikci budoucích útoků. Tato podkapitola na jednoduchém příkladu ilustruje, jak taková předpověď může fungovat.

Velmi jednoduchým prediktorem budoucích hlášení může být například počet po sobě jdoucích předchozích dní, ve kterých byla adresa nahlášena. V grafech na obrázku 7.6 je vynesena pravděpodobnost, že bude adresa v určitý den detekována, právě na základě počtu po sobě jdoucích dní předcházejících tomuto dni, ve kterých bylo k dané adrese přijato hlášení. Pokud tedy byla nějaká adresa například nahlášena jako zdroj skenování v jeden den, s pravděpodobností 27 % bude takto nahlášena i den následující. Pokud však již byla nahlášena v pěti po sobě jdoucích dnech, zvýší se pravděpodobnost nahlášení další den až na 76 %. Velmi podobný trend lze pozorovat u hlášení typu login, jen s o něco menšími prav-



Obrázek 7.6: Pravděpodobnost, že bude adresa nahlášena následující den, pokud byla nahlášena v N předchozích po sobě jdoucích dnech (N je hodnota na ose x).

děpodobnostmi. U hlášení typu exploit je pak zajímavé, že je téměř stejná pravděpodobnost dalšího hlášení, ať již bylo hlášení přijato v jednom nebo ve dvou předchozích dnech. To je pravděpodobně anomálie v této konkrétní datové sadě. Při pohledu zpět na obrázek 7.4 si totiž můžeme všimnout, že velká část hlášení typu exploit pochází z několika krátkých anomálií o délce dva až tři dny. To znamená, že mnoho adres se vyskytuje právě ve dvou či třech po sobě jdoucích dnech, což může významně ovlivnit i graf na obrázku 7.6. Neobvykle kolísající hodnoty na pravé straně grafu jsou pak pravděpodobně jen statistické nepřesnosti dané velmi malým počtem adres, které jsou hlášeny v tolika po sobě jdoucích dnech. To je vidět i z chybových značek, které znázorňují intervaly spolehlivosti uvedených hodnot pravděpodobnosti[6]. S větší datovou sadou by tedy graf pravděpodobně vypadal podobně jako u ostatních typů hlášení.

Celkově můžeme pozorovat, že informace o počtu předchozích dní, ve kterých byla adresa nahlášena, je dobrým prediktorem pravděpodobnosti budoucích útoků. To samozřejmě není jediný takový prediktor. V dřívější práci autora [63] byla například zkoumána i pravděpodobnost budoucího hlášení podle toho, v kolika (ne nutně po sobě jdoucích) n dnech z posledních m dní byla adresa nahlášena. Více vstupních parametrů samozřejmě pomáhá lépe charakterizovat předchozí chování adresy a tedy i určit pravděpodobnost jí způsobených budoucích útoků. Pokud je však používána triviální metoda výpočtu takové pravděpodobnosti jako poměru nahlášených adres pro každou kombinaci vstupních hodnot (jako v grafech výše), již při poměrně malém množství vstupních parametrů nastane problém, že možných kombinací je tolik, že pro velkou část z nich není k dispozici dostatek vzorků pro spolehlivou statistiku. Proto je nutné pro predikci použít pokročilé metody strojového učení.

[6] Pro výpočet intervalů spolehlivosti je uvažováno binomické rozdělení (zda adresa je či není nahlášena) a konfidenční hladina 95 %.

Kapitola 8

Predikce škodlivého chování IP adres

Tato kapitola podrobně popisuje, jak je obecný princip hodnocení reputace síťových entit pomocí FMP skóre, navržený v kapitole 4, aplikovaný v konkrétní situaci – hodnocení IPv4 adres na základě hlášení ze systému Warden a některých dalších dat dostupných v systému NERD. Jde tedy o tzv. *proof-of-concept* navržené metody, ukázku a vyhodnocení jejího použití v praxi.

Je zde popsán proces přípravy dat, konkrétní použité atributy feature vectoru a způsob použití predikčního modelu. Výsledky vyhodnocení jsou pak podrobně popsány v kapitole 9.

8.1 Zdroj dat

Jako hlavní zdroj dat pro tuto variantu výpočtu FMP skóre jsou uvažována data ze systému Warden, pro vyhodnocení je konkrétně použita datová sada popsaná v kapitole 7.1. Každé hlášení v této sadě obsahuje přinejmenším informace o času události, kategorii útoku, zdrojovou adresu, identifikátor detektoru a v naprosté většině případů i objem útoku, vyjádřený jako počet síťových toků či pokusů o navázání spojení. Hlášení tedy splňují požadavky definované v kapitole 4.2.

Dále jsou použita některá doplňující data o nahlášených IP adresách, odpovídající těm dostupným v systému NERD, např. přiřazená doménová jména či přítomnost adres na blacklistech.

8.2 Volba klíčových parametrů

Délka historického okna w_h použitá dále v této práci je 7 dní. Délka predikčního okna w_p je 1 den. Cílem je tedy určit pravděpodobnost, že k dané IP adrese bude přijato hlášení během následujících 24 hodin, a to na základě informací o hlášeních z předchozího týdne.

Tyto hodnoty byly stanoveny na základě analýzy dat a předchozích zkušeností s nimi (viz kap. 7.2.2 nebo tech. zprávu [63]), přibližně ale odpovídají i hodnotám používaným v jiných pracech. Například v [85] autoři zkoumali vliv délky historického i predikčního okna na kvalitu generovaných blacklistů, a jako nejlepší délka predikčního okna jim vyšel také jeden den, nejvhodnější délka historického okna je pak u jimi použitých dat 5–6 dní.

8.3 Příprava datové sady

Originální datová sada, tedy všechna hlášení za období září–listopad 2017 (viz kap. 7.1), obsahuje 155 milionů hlášení z 23 různých zdrojů. Celkem je v ní nahlášeno 5,3 milionu různých IP adres. Jak již bylo zmíněno v popisu datové sady, naprostá většina hlášení (téměř 95 %) hlásí různé typy skenování sítě (kategorie *Recon.Scanning*). Dále jsou výrazně zastoupeny kategorie *Attempt.Login* a *Attempt.Exploit* označující neoprávněné pokusy o přístup k cílovému systému, buď pomocí automatizovaných pokusů o přihlášení (typicky uhodnutím hesla) nebo prostřednictvím zneužití nějaké zranitelnosti. Pro účely této a následující kapitoly jsou hlášení těchto dvou kategorií sloučeny do jedné, nazvané *přístup*.

Pro každou IP adresu tedy budou určovány dva typy FMP skóre – $FMP_{\text{skenování}}$, predikující hlášení kategorie *Recon.Scanning*, a $FMP_{\text{přístup}}$, predikující hlášení libovolné z kategorií *Attempt.**. Ostatní typy hlášení v tomto vyhodnocení nebudou použity.

Připomeňme, že jeden *vzorek* pro strojové učení je množina vlastností IP adresy v konkrétní predikční čas. Pro přípravu datové sady tedy bylo zvoleno 24 časových okamžiků v rámci tří měsíců, ze kterých pochází použitá data, a každý byl použit jako jeden predikční čas[1]. Pro každý takový čas byl vytvořen seznam všech IP adres, které byly alespoň jednou nahlášeny daným typem hlášení v příslušném historickém okně (tedy v předchozích 7 dnech), a pro každou z nich byl vypočítán feature vector $\mathbf{x}_i$ a určena třída y_i.

To znamená, že predikce je prováděna jen pro ty adresy, které v předchozím týdnu byly již alespoň jednou nahlášeny jako škodlivé, a určuje se tedy pravděpodobnost, že budou nahlášeny znovu. Teoreticky je možné určit pravděpodobnost budoucího hlášení i pro nové, v posledním týdnu nepozorované adresy (za pomoci atributů neodvozených z předchozích hlášení), ale protože takových adres jsou miliardy a každý den je nově nahlášen jako škodlivý jen nepatrný zlomek z nich, všechny by získaly FMP skóre velmi blízké nule, což tyto případy činí nepříliš zajímavými. Kromě toho by bylo velmi obtížné získat potřebné informace, jako např. doménová jména či přítomnost na blacklistech, pro všechny takové adresy (nebo alespoň dostatečně velký reprezentativní vzorek). V neposlední řadě je provádění predikce jen pro adresy již dříve nahlášené jako škodlivé v souladu se souvisejícími pracemi o prediktivních blacklistech [83, 85].

Celkově bylo takto získáno 12,3 milionu vzorků pro predikci hlášení typu *skenování* a 765 000 vzorků pro predikci hlášení typu *přístup*, tedy dvě datové sady, jedna pro vytvoření modelu určujícího $FMP_{\text{skenování}}$, druhá pro $FMP_{\text{přístup}}$. Z obou těchto datových sad byla náhodně vybrána podmnožina vzorků, které jsou použity jako testovací sada (600 000 pro *skenování*, 100 000 pro *přístup*). Zbývající vzorky jsou použity pro trénování[2].

8.4 Feature vector

Návrh atributů feature vectoru odvozených z předchozích hlášení vychází z obecných doporučení z kapitoly 4.3. Pro každou kategorii hlášení (*skenování* a *přístup*) je pro každou IP adresu vypočítána následující sada atributů, přičemž jsou brána v úvahu hlášení obsahující danou adresu:

1. Počet hlášení v posledním dni

[1] Mezi tyto predikční časy záměrně nebyly vybrány poslední dva dny listopadu, ve kterých došlo k velké anomálii v počtu skenujících adres, jak bylo popsáno v kapitole 7.2.2.

[2] Trénovací sada je oproti testovací ponechána poměrně velká, protože bude v další fázi podvzorkováním výrazně zmenšena

2. Celkový počet pokusů o navázání spojení (objem útoku) za poslední den
3. Počet detektorů, které tuto adresu nahlásily během posledního dne
4. Počet hlášení v posledním týdnu
5. Celkový počet pokusů o navázání spojení (objem útoku) za poslední týden
6. Počet detektorů, které tuto adresu nahlásily během posledního týdne
7. EWMA počtu hlášení za den (z dat za poslední týden)
8. EWMA celkového počtu pokusů o spojení za den (z dat za poslední týden)
9. EWMA binární posloupnosti označující přítomnost hlášení (0 nebo 1) v každém dni (z dat za poslední týden)
10. Čas od posledního hlášení (v počtu dní)
11. Průměrný interval mezi hlášeními v posledním týdnu (v počtu dní, nekonečno v případě méně než dvou hlášení)
12. Medián intervalů mezi hlášeními v posledním týdnu (v počtu dní, nekonečno v případě méně než dvou hlášení)

Tato sada atributů je doplněna ještě jednou podobnou sadou, pro kterou jsou však brána v úvahu všechna hlášení obsahující jakoukoliv IP adresu ze stejného /24 prefixu, jako má vyhodnocovaná adresa (tato délka prefixu byla zvolena jako nejvhodnější pro určování podobně se chovajících adres, stejná byla použita pro agregaci zdrojů či cílů útoku v řadě dřívějších prací, např. [83, 84, 14, 76]). Tato tzv. *prefixová* sada atributů obsahuje atributy 1–9 z předchozího seznamu a navíc tyto dva následující:

- Počet různých IP adres v daném prefixu nahlášených za poslední den
- Počet různých IP adres v daném prefixu nahlášených za poslední týden

Protože existují nezanedbatelné korelace mezi událostmi typu skenování a pokusy o přístup [82, 36], vždy jsou jako vstup použity atributy vypočítané z obou kategorií hlášení, bez ohledu na to, která kategorie má být predikována.

Další dva atributy využívají informací o geolokaci IP adres a jejich příslušnosti pod určitý autonomní systém (AS). Jak bylo ukázáno v kapitole 7, relativní zastoupení škodlivých adres se může výrazně lišit podle země, do níž adresy náleží, a podle existující literatury [13] lze stejnou vlastnost očekávat i na úrovni AS. Pro každou zemi a AS byla tedy vypočítána její tzv. *škodlivost*, která vyjadřuje poměr počtu škodlivých IP adres (dle hlášení přijatých v posledním týdnu) z dané země či AS vůči celkovému počtu adres v této zemi či AS. Jako vstupní atributy jsou pak u každé adresy použity tyto poměry pro zemi a AS, do nichž daná adresa náleží.

Celkem tedy feature vector obsahuje 48 atributů odvozených z přijatých hlášení.

Další část vektoru tvoří několik atributů vycházejících z jiných zdrojů dat. Tyto jsou všechny binární, nabývají hodnotu 1, pokud je určitá vlastnost splněna, jinak 0. Zaprvé

jde o přítomnost dané IP adresy na 5 veřejných blacklistech[3] a na seznamu dynamicky přidělovaných adresních rozsahů[4].

Dále je pomocí DNS dotazů ke každé IP adrese zjištěno odpovídající doménové jméno a na něj je aplikována sada ručně navržených pravidel. Například jsou vyhledávána klíčová slova jako „static“, „dynamic“, „dsl“ nebo různými způsoby vložená IP adresa. Výsledkem jsou další 4 atributy. Důvodem pro odhadování, zda je adresa dynamicky či staticky přidělovaná je předpoklad, že u dynamicky přidělených adres se může krátce po detekci útoku zařízení za touto adresou změnit a pravděpodobnost opakování útoků z takových adres je tedy nižší.

Protože přítomnost IP adres na blacklistech se obvykle v čase mění a v některých případech se mohou měnit i doménová jména, jsou tyto údaje pro každý vzorek vždy zjišťovány v čase blízkém t_0.

Celkem se tedy feature vector skládá z 58 atributů, určených vždy pro konkrétní IP adresu a okamžik v čase. Kompletní seznam a popis atributů je uveden v příloze A.

8.5 Předzpracování

Data v obou datových sadách jsou velmi nevyvážená. V sadě *skenování* náleží do pozitivní třídy pouze 16,5 % vzorků, v sadě *přístup* je to ještě méně, pouze 8,1 %. V obou případech je tedy na trénovací část datové sady aplikováno náhodné podvzorkování majoritní, tedy negativní ($y_i = 0$), třídy, jak bylo popsáno v kapitole 4.4. Po tomto podvzorkování zůstane v datové sadě *skenování* 3,88 milionu vzorků, v sadě *přístup* 107 000 vzorků.

Dále jsou hodnoty většiny atributů transformovány nelineární funkcí, jak bylo popsáno v kapitole 4.3. Konkrétně jsou všechny atributy vyjadřující počet hlášení, objem útoku či počet detektorů transformovány funkcí $log(x+1)$. Atributy vyjadřující časové intervaly jsou transformovány funkcí $exp(-x)$. Atributy vyjadřující poměr škodlivých adres v dané zemi či AS ani binární atributy nepotřebují žádnou transformaci.

8.6 Trénování a způsob použití prediktoru

Podvzorkovaná a transformovaná data jsou pak využita pro natrénování daného modelu strojového učení, přičemž cílem trénování je minimalizace Brierova skóre vypočítaného přes všechny vzorky trénovací datové sady. Pro data typu *skenování* a *přístup* je vždy vytvořen samostatný model.

Při vyhodnocování jsou pak natrénovanému modelu předloženy vzorky testovací datové sady. Výstupy modelu, tedy predikované pravděpodobnosti budoucích hlášení pro jednotlivé IP adresy, jsou vždy nejdříve rekalibrovány pomocí vzorce 4.7 a až poté jsou výsledky vyhodnoceny.

Při praktickém nasazení by se pak postupovalo stejně – ke každé IP adrese by byl vypočítán feature vector, ten by byl předložen předem natrénovanému modelu (či modelům, pro různé typy predikovaných útoků), výstup transformován vzorcem 4.7 a výsledek by pak byl použit jako FMP skóre dané adresy.

[3] UCEPROTECT, blocklist.de-SSH a Spamhaus PBL, PBL-ISP, XBL-CBL. Bylo otestováno i několik dalších, ty ale mají jen velmi málo adres společných s použitou datovou sadou a nejsou tedy pro predikci příliš užitečné.

[4] SORBS DUL

Kapitola 9

Vyhodnocení

V této kapitole jsou popsány experimenty vyhodnocující metodu určování FMP skóre IP adres na základě předchozích hlášení a dalších informací, jak byla definována v kapitole 8.

Nejprve je vyhodnocena kvalita predikce různých modelů strojového učení v různých konfiguracích. Dále je vyhodnocen vliv jednotlivých skupin atributů feature vectoru na kvalitu predikce. Nakonec jsou vyhodnocena dvě možná praktická využití FMP skóre, konkrétně pro vytváření prediktivních blacklistů volitelné velikosti a jako kritérium pomáhající odlišit škodlivý provoz od toho legitimního při obraně proti DDoS útokům.

9.1 Výsledky modelů strojového učení

Na základě studia a řady předběžných experimentů s různými metodami strojového učení byly vybrány dvě třídy modelů strojového učení, které patří mezi v současnosti nejlepší a zdají se být pro daný problém vhodné. Jsou to *neuronové sítě* (NN) a tzv. *gradient boosted decision trees* (GBDT), což je model založený na skupině (*ensemble*) mnoha rozhodovacích stromů.

Pro implementaci NN byla použita knihovna *Keras*[1] s backendem *Theano*[2] [125], pro GBDT knihovna *xgBoost*[3] [126]. Pro některé další operace, jako je předzpracování dat a vyhodnocení výsledků, byl využit Python framework *scikit-learn*[4] [127].

Vyhodnocení sestávalo z experimentů s neuronovými sítěmi o až třech skrytých vrstvách a různými konfiguracemi GBDT modelu. Tabulka 9.1 ukazuje Brierovo skóre několika vybraných modelů pro obě datové sady.

[1] https://keras.io/
[2] http://deeplearning.net/software/theano/
[3] https://github.com/dmlc/xgboost
[4] http://scikit-learn.org/

Tabulka 9.1: Brierovo skóre různých modelů vypočítané na testovací části datových sad *skenování* a *přístup*

	skenování	přístup
NN, 2 vrstvy	0,06462	0,05486
NN, 3 vrstvy	0,06459	0,05424
GBDT(100, 3)	0,06713	0,05287
GBDT(200, 7)	0,06284	0,05065

Neuronové sítě měly 2, resp. 3, skryté vrstvy, všechny sestávající z 56 uzlů (stejně jako je počet atributů feature vectoru) s aktivační funkcí typu ReLU (rectified linear unit, tj. funkce $y = max(0, x)$). Výstupní vrstva má jeden uzel a aktivační funkci typu sigmoid. Byly provedeny i experimenty s jinými počty uzlů ve vrstvě i jinými aktivačními funkcemi, ale výsledky byly buď velmi podobné těm prezentovaným, nebo horší. Výsledky v tabulce 9.1 tedy odpovídají té nejlepší nalezené konfiguraci pro daný počet vrstev.

Modely typu GBDT sestávají ze 100 rozhodovacích stromů o maximální hloubce 3, resp. 200 stromů o maximální hloubce 7. I v tomto případě byly provedeny experimenty i s jinými konfiguracemi, ale výsledky nejsou nijak překvapivé – se stoupající složitostí modelu se pomalu snižuje Brierovo skóre, ale zároveň významně stoupají nároky na výpočetní výkon.

Čas potřebný pro natrénování modelu na datové sadě *skenování* s využitím dvou jader CPU průměrného notebooku[5] je v případě modelu $GBDT(200, 7)$ přibližně 1 hodina, u ostatních uvedených modelů je to méně než 15 minut (trénování na datové sadě *přístup* je dokončeno za méně než minutu, protože datová sada je výrazně menší).

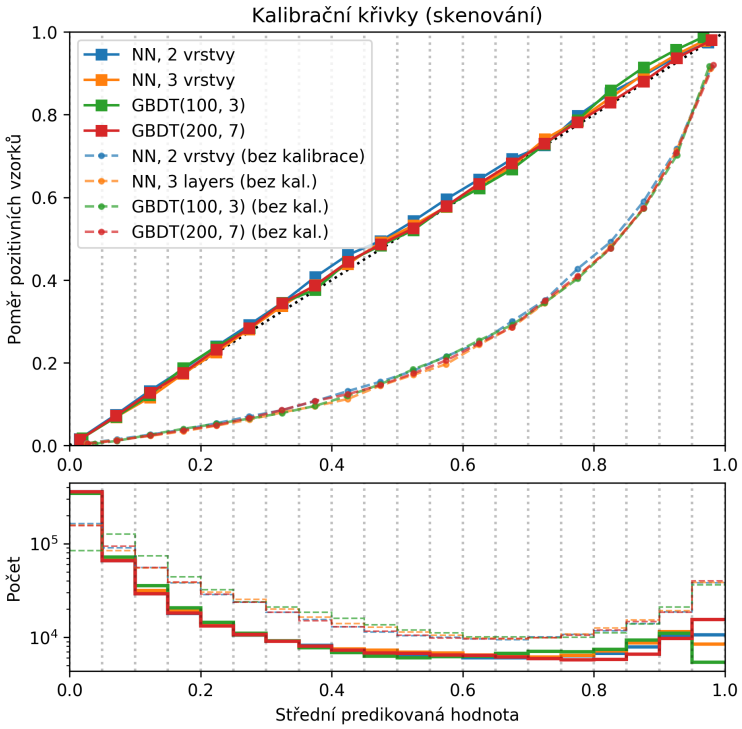
Nejlepšího Brierova skóre dosáhl na obou datových sadách model $GBDT(200, 7)$, hodnoty jsou však u všech modelů velmi podobné a blízké nule, což znamená, že všechny uvedené modely dokáží kvalitně predikovat budoucí hlášení.

Důležitým požadavkem na FMP skóre je, aby jeho hodnota skutečně odpovídala pravděpodobnosti přijetí hlášení k dané adrese. Tato vlastnost je sice zahrnuta v Brierově skóre, lze ji však přehledněji znázornit i graficky, a to pomocí tzv. kalibračních křivek pravděpodobnosti (angl. *probability calibration curves* nebo také *reliability curves*). Na obrázcích 9.2 a 9.3 jsou zobrazeny tyto křivky pro všechny čtyři modely a obě datové sady. Křivky jsou vytvořeny tak, že rozsah možných hodnot pravděpodobnosti, $[0, 1]$, je nejprve rozdělen do několika intervalů (zde 20, každý o šířce 0,05) a všechny testovací vzorky jsou rozděleny do skupin podle toho, do jakého intervalu spadá jim predikovaná pravděpodobnost, $\hat{y}_i$ (podobně jako např. při tvorbě histogramu). Každé takové skupině pak odpovídá jeden bod v grafu, jehož vodorovná souřadnice je daná průměrnou hodnotou predikované pravděpodobnosti vzorků dané skupiny, na svislou osu je pak vynesen poměr vzorků v dané skupině, které skutečně náleží do pozitivní třídy ($y_i = 1$). Pokud predikční model funguje správně, pak by předpovězené pravděpodobnosti měly odpovídat skutečnému zastoupení pozitivních vzorků v jednotlivých skupinách a všechny body grafu by tedy měly ležet velmi blízko hlavní diagonály. Pod každým grafem je navíc uveden i histogram ukazující počet vzorků v dané skupině.

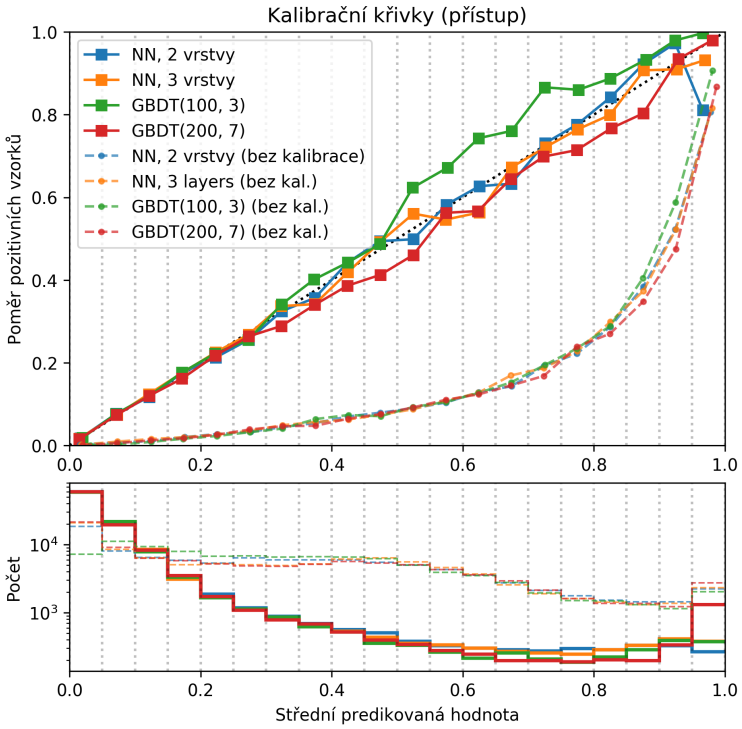
Na obrázku 9.2 můžeme vidět, že v případě datové sady *skenování* jsou všechny plné čáry (čárkované budou popsány později) velmi blízko diagonály, což znamená, že všechny modely určují pravděpodobnost budoucích hlášení velmi přesně. U datové sady *přístup* (obrázek 9.3) je výsledek o něco horší, zejména u vyšších hodnot predikované pravděpodobnosti (cca od 0,4 do 0,9). To je pravděpodobně dáno celkově nižším počtem vzorků v této datové sadě – jak je vidět z histogramu v dolní části grafu, počty vzorků ve skupinách v daném rozsahu se pohybují jen v řádu stovek. Přesto jsou však výsledné čáry poměrně blízko diagonály a přesnost predikovaných pravděpodobností lze tedy i u této datové sady hodnotit jako vyhovující.

Pro ilustraci důležitosti rekalibrace dle vzorce 4.7 je v obrázcích ukázáno i to, jak by kalibrační křivky vypadaly, pokud by kalibrace provedena nebyla (čárkovanou čarou). Je vidět, že v takovém případě jsou výstupy predikčního modelu značně zkreslené a predikovaná pravděpodobnost vůbec neodpovídá té skutečné. Například pokud nekalibrovaný

[5] Intel Core i5 4310M @2,7 MHz, 8 GB RAM, SSD disk



Obrázek 9.2: Kalibrační křivky pravděpodobnosti pro 4 různé modely a testovací datovou sadu *skenování*



Obrázek 9.3: Kalibrační křivky pravděpodobnosti pro 4 různé modely a testovací datovou sadu *přístup*

model pro určitou skupinu vzorků predikuje pravděpodobnost kolem 0,6, křivka ukazuje, že jen asi 20 % z takových vzorků patří skutečně do pozitivní třídy, ačkoliv by to mělo být přibližně 60 % (hodnoty pro datovou sadu *skenování*, u datové sady *přístup* je situace podobná).

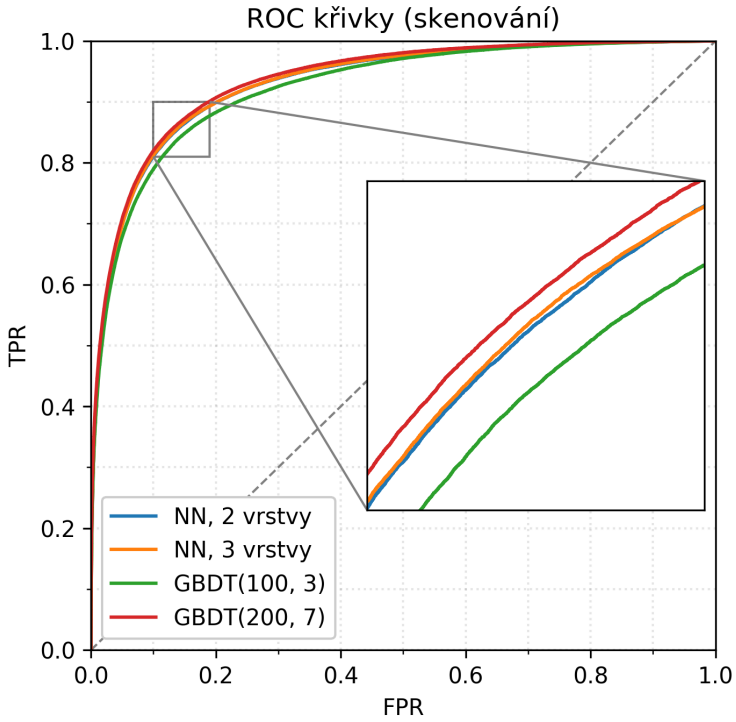
V mnoha případech použití bude na FMP skóre aplikována nějaká mez za účelem rozdělení IP adres na škodlivé a neškodné, např. pro vytvoření blacklistu. Tím se problém odhadu pravděpodobnosti binárních tříd redukuje na běžnou binární klasifikaci.

Zde je vhodné poznamenat, že cílem této práce není navrhnout dokonalý klasifikátor. Vstupní data rozhodně nejsou dostatečná pro jednoznačné určení, zda daná adresa bude či nebude v budoucnu útočit, protože chování útočníků je ovlivňováno mnoha neznámými faktory, včetně například náhodného výběru cílů. Ve většině případů je proto možné pouze odhadnout pravděpodobnost budoucího útoku – což je hlavní cíl této práce a vyhodnocení je zaměřeno především na něj. Nicméně metriky pro vyhodnocování výsledků binární klasifikace jsou obecně dobře známé a snadno srozumitelné a mohou tedy pomoci lépe vyhodnotit kvalitu modelů.

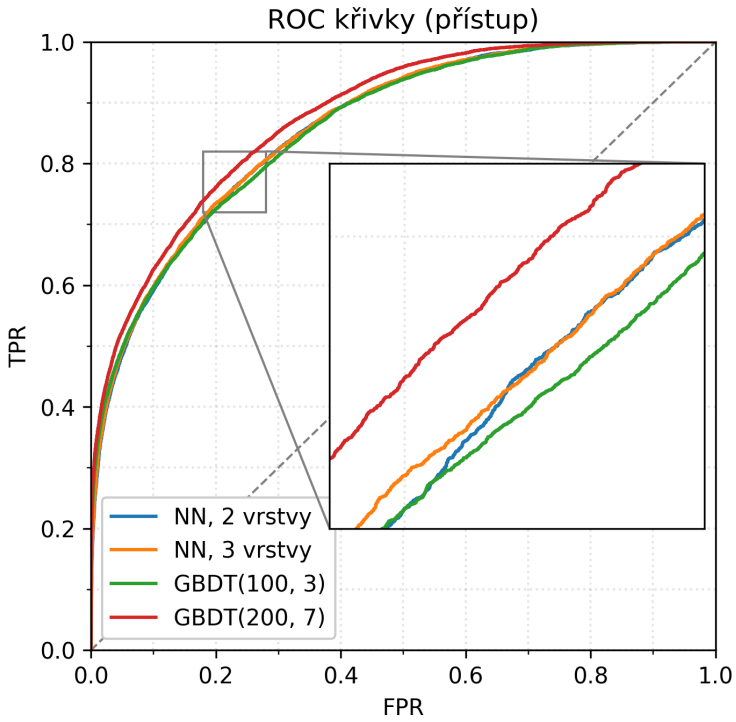
Nejběžnějším způsobem vizualizace výsledků v případě binární klasifikace jsou tzv. ROC křivky (z angl. Receiver Operating Characteristic). Tyto křivky ukazují vztah mezi poměrem pravdivě pozitivních (*true positives*, TP) a falešně pozitivních (*false positives*, FP) výsledků binárního klasifikátoru v závislosti na nastavení meze [128]. V našem případě je za falešně pozitivní výsledek považována situace, kdy je IP adresa označena za škodlivou (tzn. její FMP skóre je vyšší než zvolená mez), ale žádné hlášení k ní během predikčního okna nepřijde. Pokud k takové adrese hlášení přijde, jde o pravdivě pozitivní výsledek. V grafu je pak vynesen poměr počtu TP, resp. FP, vzhledem k celkovému počtu pozitivních, resp. negativních, vzorků (nazývané TPR, resp. FPR, z angl. *true/false positive rate*). Každý bod ROC křivky pak odpovídá hodnotám TPR a FPR při jiné volbě meze. Interpretace ROC křivek je snadná – prediktor, který by dával zcela náhodné výsledky, by měl ROC křivku rovnu hlavní diagonále, čím více se křivka od této diagonály odchyluje a blíží se levému hornímu rohu, tím přesnější predikce je.

Obrázky 9.4 a 9.5 ukazují tyto křivky pro 4 výše uvedené modely a obě datové sady. Všechny křivky jsou poměrně hladké a vzájemně velmi podobné. Jediný výraznější rozdíl je mezi datovými sadami, kdy hlášení typu *skenování* se zdají být snáze predikovatelné než hlášení typu *přístup*. Při bližším pohledu je vidět, že nejlepších výsledků dosahuje model $GBDT(200, 7)$ (stejně jako při porovnání Brierových skóre). Konkrétně lze z ROC křivek vyčíst například to, že pokud je v případě skenování mez nastavena tak, aby míra falešně pozitivních výsledků byla 10 %, je možné zachytit (a zablokovat v případě použití jako blacklistu) až 80 % všech opakujících se zdrojů skenování[6]. Je důležité poznamenat, že falešně pozitivní výsledek zde nutně neznamená zablokování legitimní IP adresy, příslušná adresa může být stále škodlivá, pouze během predikčního okna neprovedla žádný útok vůči sledované síti. V takovém případě jde pouze o zbytečně zabrané místo na blacklistu. To umožňuje posunout mez i do oblasti s poměrně vysokým počtem falešně pozitivních výsledků a zablokovat tak naprostou většinu opakujících se zdrojů škodlivého provozu. Jedinou cenou za to je větší velikost blacklistů. Podrobněji je možnost použití FMP skóre pro generování blacklistů zkoumána v kapitole 9.3.

[6] Připomeňme, že jsou brány v úvahu pouze ty IP adresy, které již byly alespoň jednou nahlášeny jako škodlivé v rámci historického okna (ty, které se objeví poprvé až během predikčního okna je téměř nemožné předem odhadnout).



Obrázek 9.4: ROC křivky pro 4 různé modely a testovací datovou sadu *skenování*



Obrázek 9.5: ROC křivky pro 4 různé modely a testovací datovou sadu *přístup*

Celkově jsou výsledky uvedených modelů podle všech metrik – Brierova skóre, kalibračních křivek i ROC křivek – velmi podobné, přičemž model $GBDT(200, 7)$ je zpravidla o trochu lepší než ostatní. Ve zbytku práce je tedy nadále používán pouze tento model.

9.2 Skupiny atributů feature vectoru

Pro ověření, zda jsou všechny atributy feature vectoru pro predikci skutečně užitečné, dále vyhodnocujeme výsledky modelu s různými skupinami vstupních atributů. Ve všech případech je model trénován a testován na stejných datech jako v předchozí kapitole, pouze jsou odstraněny některé skupiny atributů z feature vectoru.

Výsledné ROC křivky jsou na obrázcích 9.6 a 9.7. Základem je prediktor využívající pouze informace z hlášení stejné kategorie, jako je ta predikovaná, a pouze z těch hlásících přímo danou IP adresu (tedy ne ostatní adresy se stejným /24 prefixem). Další křivky pak ukazují výsledky v případech, kdy je k tomuto základu přidána jedna z dalších skupin atributů – hlášení o IP adresách se stejným prefixem, hlášení druhé kategorie, množina binárních atributů (příznaků) odvozených z doplňkových dat a atributy vyjadřující celkovou škodlivost země a autonomního systému. Poslední křivka pak ukazuje výsledek při použití všech atributů zároveň (ta je stejná jako v kapitole 9.1).

Z grafů je zřejmé, že lze dosáhnout poměrně dobrých výsledků i se základní sadou atributů, především u datové sady *skenování* jsou rozdíly mezi jednotlivými variantami velmi malé. Přesto přidání jakékoliv další sady atributů způsobí viditelné zlepšení výsledků. Zároveň však žádná skupina atributů sama o sobě nezlepší výsledky tolik, jako kombinace všech. Z toho vyplývá, že informace ze všech skupin atributů dokáže predikční model využít pro zpřesnění predikce a všechny jsou tedy skutečně užitečné.

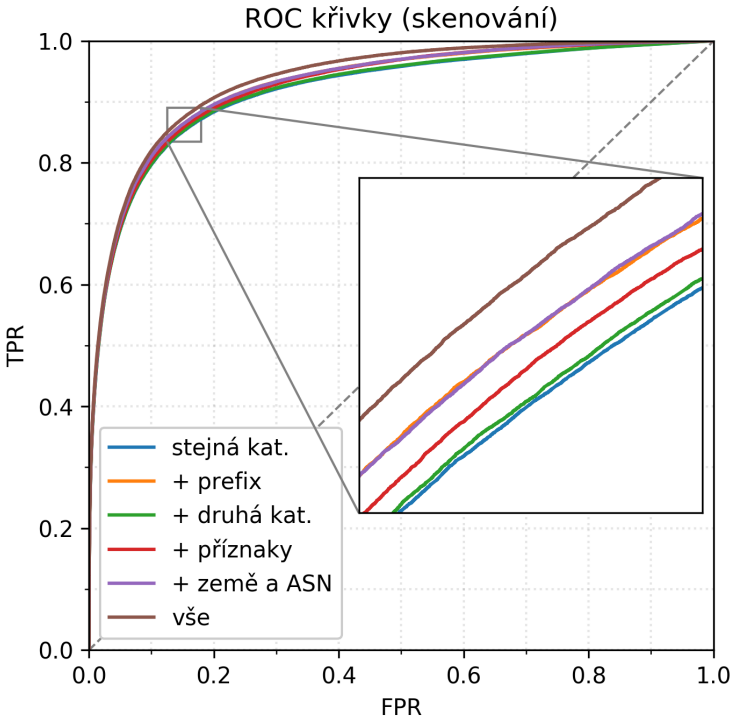
Zajímavé je však i porovnání významu jednotlivých skupin atributů u různých datových sad. Například v případě predikce hlášení typu *přístup* mají na výsledek největší vliv informace z hlášení druhého typu, tedy *skenování*. To potvrzuje, že skutečně existují významné korelace mezi těmito dvěma typy hlášení. Tento vliv je více znatelný při vyšší míře falešně pozitivních výsledků – důvodem je to, že po skenování zdaleka ne vždy přichází pokus o přístup a predikce hlášení typu *přístup* pouze na základě skenování je tedy vždy značně nejistá, tj. predikovaná pravděpodobnost takového hlášení je relativně malá. Teprve při nastavení nízké meze se tak může možnost predikovat jeden typ hlášení pouze na základě druhého plně projevit (zároveň s tím se však zvýší i počet falešně pozitivních výsledků).

Na druhou stranu, v případě predikce hlášení typu *skenování* informace o druhém typu příliš nepomáhají – to proto, že hlášení typu *přístup* je podstatně méně a výše zmíněné korelace zde tedy v mnoha případech nelze využít.

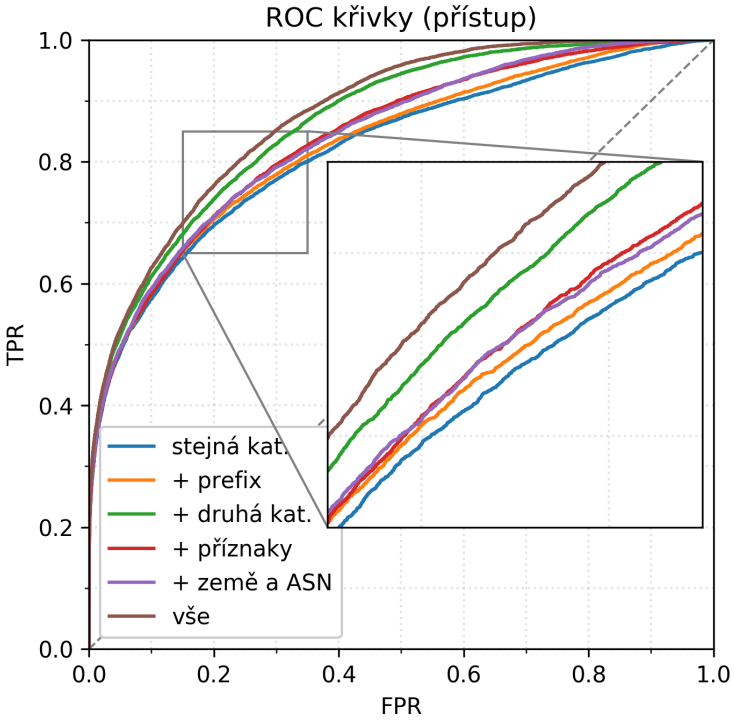
Dále se ukázalo, že jedna z hlavních nových myšlenek této práce, tedy využití i jiných zdrojů dat než jen předchozích hlášení o škodlivém chování, se ukázala u obou datových sad jako užitečná. Zlepšení výsledků díky této skupině atributů sice není velké, rozhodně však není zanedbatelné.

9.3 Využití FMP skóre pro vytváření prediktivních blacklistů

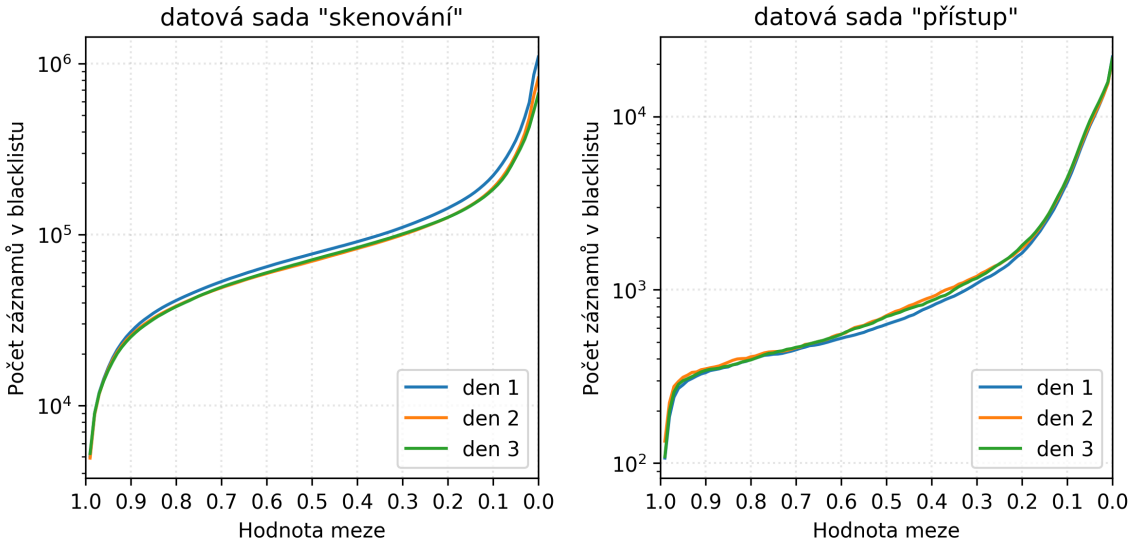
Tato podkapitola je věnována vyhodnocení jednoho z možných použití FMP skóre – generování blacklistů volitelné velikosti. Při tomto použití je na konci každého dne sestaven



Obrázek 9.6: ROC křivky při použití různých skupin vstupních atributů (datová sada *skenování*)



Obrázek 9.7: ROC křivky při použití různých skupin vstupních atributů (datová sada *přístup*)



Obrázek 9.8: Velikost blacklistu v závislosti na hodnotě meze aplikované na FMP skóre. Každá křivka odpovídá blacklistům generovaným pro jeden ze tří dnů.

seznam IP adres s největším FMP skóre (blacklist) a ten je pak používán pro blokování provozu[7] během následujícího dne.

Velikost či restriktivnost blacklistu je definována uživatelem (administrátorem sítě) – buď je použit pevně daný počet IP adres s nejvyšším skóre, nebo jsou použity všechny adresy se skóre vyšším než určitá mez. Jak bylo zmíněno už v úvodu této práce, toto je velká výhoda oproti tradičním blacklistům, kdy je obvykle dáván k dispozici pouze finální seznam škodlivých IP adres, na jehož pravidla tvorby ani použité meze nemá uživatel blacklistu žádný vliv.

Pro vyhodnocení efektivity blacklistů je zde použita metrika *hit-count*. Ta je definovaná (v souladu s [83, 85], viz kap. 3.2) jako počet IP adres na blacklistu, které jsou správně předpovězeny jako škodlivé, tzn. skutečně je během predikčního okna (tj. následujícího dne) detekován útok z dané adresy. V případě použití blacklistu pro blokování provozu je to tedy počet úspěšně zablokovaných útočníků. Zde navíc definujeme metriku *hit-rate*, což je hodnota *hit-count* vydělená velikostí blacklistu. Určuje tedy, jaké procento záznamů v blacklistu se ukázalo jako užitečné.

Je zřejmé, že z hlediska metriky *hit-count* je optimálním způsobem vytvoření blacklistu o velikosti N vždy vybrat takových N IP adres, jejichž pravděpodobnost, že budou v daný čas útočit, je ze všech adres nejvyšší. Pokud tedy FMP skóre dobře aproximuje skutečnou pravděpodobnost budoucího útoku, blacklisty vygenerované podle něj se blíží optimu dosažitelnému s daným množstvím vstupních informací.

Pro experimentální vyhodnocení kvality blacklistů bylo simulováno jejich použití v několika dnech, počet útočníků, které by blacklist zablokoval, byl odvozen na základě hlášení o detekovaných útocích z těchto dní. Byly vybrány tři dny v první polovině prosince 2017, tedy krátce po konci období, z něhož pocházejí data použitá pro trénování modelu, tak jako by tomu bylo při použití metody v praxi. Pro každý den byly vypočítány feature vectory všech adres nahlášených alespoň jednou během předchozího týdne a pomocí modelu natrénovaného v předchozích kapitolách ($GBDT(200, 7)$ se všemi atributy feature vectoru)

[7] Nebo aplikaci rate-limitingu či jiných restriktivních opatření dle potřeb uživatele.

Tabulka 9.9: Porovnání blacklistů různých typů a velikostí pomocí metrik hit-count, hit-rate a procenta zablokovaných útočníků

N	blacklist	T	hit-count	hit-rate	% útočníků
100	FMP	0,99	100	100 %	2,3 %
	$GWOL_1$	–	83	83 %	1,9 %
	$GWOL_7$	–	71	71 %	1,6 %
500	FMP	0,65	443	89 %	10,1 %
	$GWOL_1$	–	236	47 %	5,4 %
	$GWOL_7$	–	233	47 %	5,3 %
2000	FMP	0,18	862	43 %	19,7 %
	$GWOL_1$	–	650	33 %	14,9 %
	$GWOL_7$	–	579	29 %	13,2 %
388444	uceprotect	–	463	0,12 %	10,6 %
8063	bl.de-ssh	–	336	4,2 %	7,2 %
1503	bfb	–	70	4,7 %	1,6 %

bylo každé adrese přiřazeno FMP skóre. Seznam adres pro každý den byl seřazen sestupně podle FMP skóre. Blacklisty jsou pak vytvářeny vždy jako prvních N adres z takového seznamu. Ekvivalentně mohou být na blacklist přidány všechny adresy s FMP skóre větším nebo rovným určité mezní hodnotě. Na obrázku 9.8 je ukázán vztah mezi takovou mezí a délkou blacklistu pro jednotlivé dny a datové sady.

Jednou z výhod FMP skóre je snadná interpretace. Nastavením meze například na 0,9 získáme všechny IP adresy, u nichž je odhadnutá pravděpodobnost škodlivého chování v následujícím dni větší než 90 %. Z grafů lze vyčíst, že v případě *skenování* je takových adres asi 25 000, u datové sady *přístup* jen něco málo přes 300. Uživatel, který bude chtít na základě seznamu IP adres ohodnocených FMP skóre generovat blacklist, může využít podobného grafu pro určení optimální rovnováhy mezi jistotou správné predikce a velikostí blacklistu dle jeho potřeb a možností. Dále je z grafů patrné, že výsledky se v jednotlivých dnech příliš neliší, což potvrzuje, že charakteristiky, ze kterých vychází predikční model, jsou přinejmenším z krátkodobého hlediska poměrně stabilní.

Dále jsou pro hodnocení využita jen data typu *přístup*, protože pokusy o neoprávněný přístup jsou jistě závažnější události než skenování a dává tedy větší smysl zdroje takových aktivit blokovat či aplikovat jiná omezení na související provoz.

Pro vyhodnocení metrik *hit-count* a *hit-rate* byly na základě FMP skóre vytvořeny blacklisty několika různých velikostí. Pro porovnání byly na základě stejných dat (tj. hlášení ze systému Warden) vytvořeny také blacklisty založené na základní metodě GWOL (*global worst offender list*, termín převzat z [83, 85]). V případě této metody se blacklist skládá z těch IP adres, ke kterým bylo za určité předchozí období přijato nejvíce hlášení. V této práci jsou použity dvě varianty lišící se délkou tohoto období – 1 den ($GWOL_1$) a 7 dní ($GWOL_7$). Stejně jako v případě blacklistů založených na FMP skóre lze i v případě GWOL generovat blacklisty různé délky, vždy tedy vzájemně porovnáváme blacklisty se stejným počtem záznamů.

Dále jsou pro porovnání vyhodnoceny tři reálné blacklisty poskytované třetími stranami, konkrétně UCEPROTECT[8], blocklist.de-SSH[9] (*bl.de-ssh*) a BruteForceBlocker[10] (*bfb*). Tyto blacklisty jsou založené na jiných datech. Jejich velikost je pevně daná.

V tabulce 9.9 jsou uvedeny různé metriky pro všechny testované blacklisty. FMP a GWOL blacklisty jsou generovány ve velikostech 100, 500 a 2000 záznamů. Sloupec označený T ukazuje hodnotu meze FMP skóre odpovídající dané velikosti blacklistu. Jinými slovy, FMP blacklisty obsahují vždy ty adresy, které splňují $FMP_{\text{pristup}} \geq T$. Sloupec *hit-count* ukazuje počet adres, které ve dni, pro který byl blacklist připraven, skutečně zaútočily a byly by pomocí blacklistu zablokovány. *Hit-rate* je hodnota *hit-count* vydělená N, tedy procento záznamů v blacklistu, které úspěšně zablokovaly nějaký útok. Všechny hodnoty v tabulce jsou průměrem ze tří testovaných dnů.

Z tabulky je zřejmé, že obecně mají menší blacklisty vyšší hodnoty hit-rate. To je očekávané, protože tyto obsahují jen adresy s největší pravděpodobností budoucích útoků (resp. nejaktivnější v předchozích dnech v případě GWOL). Obzvlášť efektivní jsou FMP blacklisty o délce 100 záznamů, u nichž ve dvou případech v daném dni skutečně zaútočilo všech 100 uvedených adres, ve třetím jich pak bylo 99. Celkově jsou ve všech případech FMP blacklisty výrazně efektivnější než ty vytvořené metodou GWOL.

V průměru bylo v každém dni nahlášeno 4376 různých útočících IP adres. Poslední sloupec v tabulce 9.9 ukazuje, kolik z těchto adres by bylo kterým blacklistem zablokováno. Hodnoty se nezdají být nijak vysoké, je však nutné poznamenat, že přibližně 60 % útočníků v každém dni je „nových“, tzn. nebyli v předchozím týdnu ani jednou detekováni a jejich útoky je tak téměř nemožné předvídat. Maximální dosažitelné procento zablokovaných útočníků je tedy kolem 40 %.

Blacklisty třetích stran se ukázaly být z pohledu *hit-rate* velmi neefektivní, neboť pouze velmi malé procento adres uvedených na blacklistu bylo skutečně detekováno jako zdroj nějakého útoku. To je dáno tím, že tyto blacklisty jsou vytvářeny na základě zcela jiných zdrojů dat a můžou tak uvádět i útočníky, kteří necílí na žádné sítě či protokoly sledované detektory přispívajícími do systému Warden.

Nicméně další analýza ukázala, že pokud není problémem přílišná velikost blacklistů, je výhodné zkombinovat FMP blacklist s těmito blacklisty třetích stran. Seznam sjednocující FMP blacklist s mezí 0,5 (681 záznamů) se všemi třemi blacklisty třetích stran (celkem 397 241 záznamů) dokáže zablokovat 24,1 % útočících IP adres. Je však třeba myslet také na to, že příliš velký blacklist může zvýšit pravděpodobnost zablokování legitimního provozu.

Vyhodnocení lze shrnout tak, že blacklisty vytvořené na základě FMP skóre jsou velmi efektivní. Při stejné velikosti dokáží zablokovat výrazně více útočníků než blacklisty typu GWOL vytvořené na základě stejných dat. I v porovnání s různými blacklisty třetích stran dokáží FMP blacklisty zablokovat srovnatelný či větší počet útočníků, ovšem při mnohem menší velikosti blacklistu a tedy s nižšími nároky na výkon a také s nižší pravděpodobností zablokování legitimního provozu.

[8] http://www.uceprotect.net/en/
[9] https://www.blocklist.de/en/
[10] http://danger.rulez.sk/index.php/bruteforceblocker/

9.4 Využití FMP skóre pro efektivnější obranu proti DDoS útokům

Další možné využití FMP skóre je jako jedno z kritérií pro rozlišení škodlivého a legitimního provozu při obraně proti DDoS útokům. Implementace a vyhodnocení této možnosti použití bylo provedeno v rámci diplomové práce T. Jánského [129], jejímž byl autor této disertační práce konzultantem. Výsledky byly také publikované v konferenčním příspěvku [130]. Tato kapitola je stručným shrnutím těchto prací.

Práce se zabývají obranou proti tzv. objemovým DDoS útokům, tj. takovým, při kterých je cílový server nebo jeho síťové připojení zaplaveno obrovským množstvím požadavků či obecně jakýmkoliv síťovým provozem. Obrana proti takovým útokům obecně spočívá v rozpoznání a zahození škodlivého provozu, tedy toho, který je vygenerován útočníkem a nepředstavuje reálné požadavky uživatelů. Kromě výkonnostních problémů, spojených s nutností filtrace obrovského množství provozu, je klíčovou úlohou schopnost spolehlivě rozpoznat škodlivý provoz od toho legitimního. Není přitom nutné zahodit všechen škodlivý provoz, obvykle stačí celkové množství provozu snížit na určitou úroveň, kterou už je linka schopna bez problémů přenést a server zpracovat, a při tom zahodit co nejméně legitimního provozu. FMP skóre, v [130] nazýváno obecněji jako *reputation score*, je zde použito právě pro zpřesnění metody rozlišující škodlivý a legitimní provoz.

9.4.1 DDoS Mitigation Device (DMD)

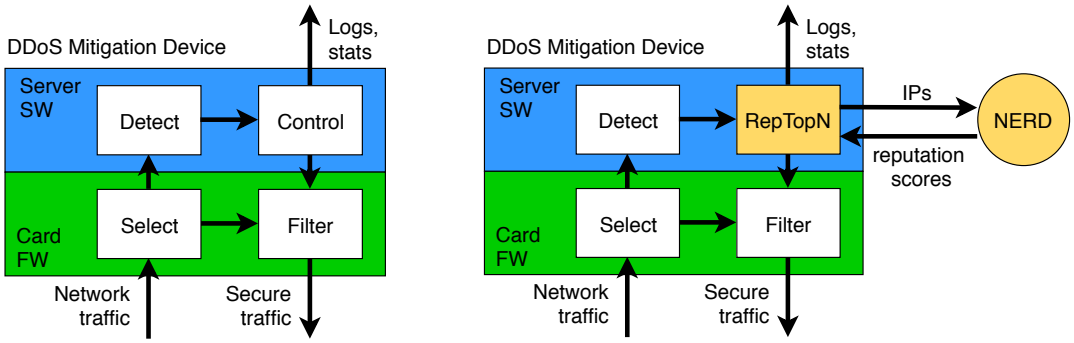
Nová metoda je navrhována a vyhodnocována v kontextu konkrétního zařízení. Jde o tzv. *DDoS Mitigation Device (DMD)*, zařízení vytvořené z běžného serveru a speciální síťové karty s FPGA čipem pro hardwarovou akceleraci filtrování síťového provozu, vyvinuté organizací CESNET. Architektura tohoto zařízení je znázorněna na obrázku 9.10a.

Hardwarová akcelerační karta implementuje přeposílání a filtraci síťového provozu na rychlostech až 100 Gb/s. Při tom měří určité statistiky, které jsou v řídicím software analyzovány a při DDoS útoku jsou na základě nich sestavována pravidla pro filtrování provozu z IP adres identifikovaných jako zdroje útoku[11]. Součástí konfigurace zařízení je seznam chráněných síťových prefixů a meze objemu provozu pro každý z nich (v počtu bitů či paketů za sekundu). Cílem zařízení je pak v případě útoku snížit množství provozu do cílové sítě na nastavenou úroveň.

9.4.2 Algoritmus výběru blokovaných IP adres

Původní algoritmus DMD pro výběr adres, jejichž provoz má být blokován, nazvaný *top-n*, je založen jen na aktuálním objemu provozu posílaného k cíli z jednotlivých zdrojových adres. Zdrojové adresy jsou seřazeny sestupně podle jejich příspěvku k celkovému objemu provozu k cíli a pak jsou adresy od začátku seznamu postupně přidávány do seznamu blokovaných, dokud souhrnný objem provozu těch zbývajících neklesne pod nastavenou mez. Toto měření provozu a nastavování filtračních pravidel se opakuje každou vteřinu. Je tedy blokováno vždy n nejaktivnějších adres, přičemž n se průběžně mění dle potřeby.

[11] Tato práce se zabývá pouze těmi útoky, při nichž nejsou zdrojové adresy náhodně podvrženy. DMD nabízí i několik dalších metod obrany, které lze použít v případě jiných typů útoku, ty však nejsou pro tuto práci relevantní.



(a) Původní architektura

(b) Vylepšení o algoritmus *RepTopN*

Obrázek 9.10: Architektura zařízení DDoS Mitigation Device (DMD) (převzato z [130])

Je však zřejmé, že ne všechny z takto blokovaných adresy musí být vždy součástí útoku, v mnoha případech může být i velmi výrazný zdroj provozu legitimní a přitom provoz z některých útočících adres může být relativně malý.

Vylepšený algoritmus, *RepTopN*, proto přidává do rozhodovacího procesu další informaci, a to reputační skóre jednotlivých IP adres (konkrétně FMP skóre poskytované systémem NERD). Základní myšlenkou je prioritně blokovat IP adresy s vysokým FMP skóre, i když jejich příspěvek k celkovému objemu provozu nemusí být velký, protože u nich je vysoká pravděpodobnost, že jde o zdroje útočného provozu a že tak nebudou zablokováni legitimní uživatelé.

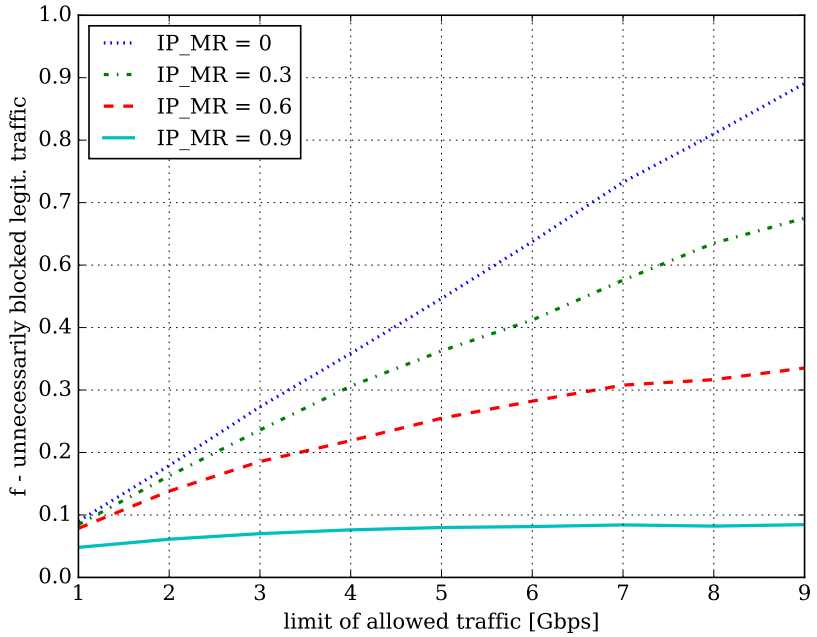
Na obrázku 9.10b je znázorněno rozšíření architektury zařízení DMD. Základ algoritmu je stejný, pravidelně je měřen objem provozu jednotlivých IP adres vůči cíli, navíc je však ke každé takové IP adrese ze systému NERD vždy získáno jeho aktuální FMP skóre[12]. Seznam IP adres je pak seřazen primárně podle tohoto skóre, objem provozu je použit jako sekundární klíč. Následně je opět ze začátku seřazeného seznamu vybráno tolik adres k zablokování, aby celkový objem zbývajícího provozu klesl pod požadovanou mez.

9.4.3 Experimenty

Bylo provedeno několik experimentů za účelem ověření efektivity algoritmu *RepTopN* v porovnání s původním *top-n*, zejména v závislosti na počtu IP adres zapojených do útoku a na množství informací o útočících IP adresách dostupných v systému NERD. Protože reálné DDoS útoky jsou v síti CESNET, k jejímž datům mají autoři přístup, poměrně vzácné, byly pro účely testování útoky pouze simulovány. Jako hodnotící metrika je použito množství legitimního provozu zablokovaného daným algoritmem (které by mělo být co nejmenší) při určité konfiguraci experimentu.

Jedním ze základních parametrů experimentů je procento útočících IP adres, které jsou již známy jako škodlivé, tj. mají již v systému NERD záznam s nenulovým FMP skóre. Důležitou vlastností algoritmu *RepTopN* je, že i v nehorším případě, kdy není o žádné z IP adres zapojených do útoku známa žádná předchozí škodlivá činnost a všechny tedy mají přiřazeno nulové FMP skóre, dojde pouze k degradaci na základní *top-n*, kdy se algoritmus řídí jen naměřeným objemem provozu a dosahuje tedy stejných výsledků jako v původní variantě. Avšak ve všech ostatních případech, za předpokladu, že adresy s nenulovým FMP skóre jsou skutečně s větší pravděpodobností škodlivé, dosahuje *RepTopN* algoritmus vždy

[12]Pokud v systému NERD k dané adrese neexistuje záznam, je hodnota skóre nastavena na 0.



Obrázek 9.11: Efektivita *RepTopN* v závislosti na množství legitimního provozu a podílu škodlivých IP adres s nenulovým FMP skóre (převzato z [130])

lepších výsledků než *top-n*, tzn. dokáže provoz regulovat s menším množstvím zablokovaného legitimního provozu.

Na obrázku 9.11 jsou ukázány výsledky jednoho z experimentů. Při něm bylo množství simulovaného legitimního provozu přibližně rovno nastavené mezi, na kterou má být při útoku celkové množství provozu redukováno. Takový případ je pro algoritmus nejnáročnější, protože aby nebyl zahozen žádný legitimní provoz, musí být správně rozpoznán a zahozen všechen škodlivý provoz, nestačí zahodit jen jeho část, jako v jiných případech. Celkové množství provozu je zde vždy 10 Gb/s. Na vodorovné ose je uvedena hodnota meze a zároveň tedy také množství legitimního provozu, zbývající množství provozu je součástí DDoS útoku. Na svislé ose je vynesen zlomek legitimního provozu, který byl algoritmem zablokován. Hodnota IP_MR vyjadřuje zlomek útočících IP adres, které mají nenulové FMP skóre. V případě IP_MR = 0 je algoritmus ekvivalentní základnímu algoritmu *top-n*. Pokud jsou však alespoň některé útočící adresy již známé jako škodlivé, je vidět, že množství zablokovaného legitimního provozu znatelně klesá. Tím se potvrzuje, že algoritmus *RepTopN*, využívající informace o reputaci IP adres ve formě FMP skóre, skutečně pomáhá rozlišit a zablokovat škodlivý provoz a tím zmírnit dopad DDoS útoku.

Pro podrobný popis nastavení experimentů a pro další výsledky je čtenář odkázán na [130].

Ačkoliv byla metoda *RepTopN* navržena a otestována na konkrétním zařízení, princip využití FMP skóre pro rozlišení škodlivého a legitimního provozu je samozřejmě použitelný obecněji a obdobný postup lze použít i v případě řady jiných algoritmů a zařízení, a to ne nutně jen v kontextu DDoS útoků.

Kapitola 10

Závěr

V této práci byla představena metoda pro číselné vyjádření reputace síťových entit (především IP adres) z hlediska bezpečnostních hrozeb. Toto číslo, nazvané *Future Misbehavior Probability score* či *FMP skóre*, slouží jako shrnutí všech dostupných bezpečnostně relevantních informací o dané entitě a zároveň jako předpověď jejího budoucího chování. Konkrétně FMP skóre vyjadřuje pravděpodobnost, že daná entita bude v příštích 24 hodinách detekována jako zdroj určitého nežádoucího chování, přičemž určení této pravděpodobnosti je prováděno pomocí strojového učení s využitím všech dostupných dat o dané entitě i ostatních „blízkých“ či podobných entitách. Tento obecný princip byl dále konkretizován pro případ ohodnocování škodlivých IPv4 adres a ověřen na reálných datech.

Součástí provedených experimentů bylo vyhodnocení různých variant predikčních modelů. Bylo ukázáno, že všechny testované modely strojového učení (neuronové sítě, lesy rozhodovacích stromů) dokáží dostatečně přesně odhadovat skutečnou pravděpodobnost budoucích hlášení a mohou být tedy využity pro výpočet FMP skóre. Obecně platí, že komplexnější model poskytuje přesnější výsledky, ale rozdíly mezi testovanými variantami byly jen malé. Pro kvalitní predikci je však třeba poměrně velké množství trénovacích dat, řádově alespoň statisíce vzorků předchozích hlášení.

Výsledné FMP skóre může být využito hned několika způsoby. Tím základním je shrnutí dostupných informací o dané entitě do snadno uchopitelného čísla, prezentovaného uživateli, které člověku umožňuje rychle vyhodnotit a porovnat škodlivost jednotlivých entit. Podobným způsobem lze takové číselné ohodnocení využít i strojově, jak bylo ukázáno na příkladu filtrování DDoS útoků v kapitole 9.4.

Při kombinaci s dalšími údaji, jako je například vyhodnocení důležitosti cíle útoku, může být FMP skóre využito i pro prioritizaci incidentů, procesu, jehož cílem je napovědět bezpečnostnímu operátorovi, kterými událostmi se zabývat přednostně.

Další možností využití je vytváření blacklistů s adresami s nejvyšším FMP skóre, tedy s největší pravděpodobností budoucích útoků, které pak lze použít například k blokování provozu těchto adres. Výhodou oproti tradičním blacklistům je možnost zvolit si libovolně velikost takového blacklistu, resp. mezní hodnotu FMP skóre. Experimenty provedené v kapitole 9.3 ukázaly, že blacklisty vytvořené podle FMP skóre dokáží být velmi efektivní z hlediska počtu zablokovaných útočníků vzhledem k velikosti blacklistu.

V porovnání s předchozími pracemi zabývajícími se hodnocením škodlivosti síťových entit je navržená metoda unikátní v tom, že kombinuje výhody všech předchozích přístupů – jde o číselné hodnocení umožňující adresy porovnávat a řadit, pracuje se s jednotlivými adresami namísto celých sítí, přičemž jsou ale zároveň zahrnuty i informace o dalších adresách ve stejné síti, a hodnocení je založeno na explicitní predikci budoucího chování, namísto

pouhého shrnutí předchozích aktivit. Navíc je význam FMP skóre jednoduše interpretovatelný, což usnadňuje jeho používání, případně nastavovaní mezí. Metoda je unikátní i tím, že umožňuje využít prakticky jakákoliv dostupná data, například odhad, zda je adresa dynamicky přidělovaná, či geolokační informace. Předchozí práce vždy vycházely jen z analýzy blacklistů nebo z přijatých hlášení o detekovaných bezpečnostních událostech.

V práci byl také popsán přínos autora v oblasti detekce škodlivého provozu – návrh několika metod detekce škodlivého provozu, významný podíl na návrhu frameworku pro analýzu síťových dat (NEMEA), jeho implementaci, včetně implementace navržených detekčních metod, a nasazení celého systému v reálné síti. Tato činnost pomohla získat velké množství dat o síťových útocích, jejichž analýza pomohla odhalit či ověřit vlastnosti zdrojů škodlivého chování, na jejichž znalosti staví navržená metoda hodnocení reputace. Kromě toho, že byla data z detekčního systému NEMEA využita pro tuto práci, je systém již několik let úspěšně používán v praxi a přispěl i ke vzniku řady akademických prací.

V neposlední řadě byl popsán i systém pokročilé reputační databáze NERD, jenž byl také vytvořen v souvislosti s touto prací. Jeho účelem je jednak získávat dodatečné informace ke škodlivým IP adresám, zároveň slouží jako platforma, pro niž je metoda výpočtu FMP skóre primárně určena.

Během dokončování této práce byl výpočet FMP skóre do systému NERD skutečně implementován. Skóre je zde počítané pro všechny adresy, k nimž tento systém udržuje nějaký záznam, je pravidelně aktualizované a volně dostupné, kdokoliv tak může tato data využít v praxi. Samozřejmostí je i možnost vytvářet blacklisty dle zadaných kritérií a také možnost dotazovat se na skóre IP adres nejen přes webové GUI, ale i programové API, což umožňuje snadnou integraci do jiných systémů.

Na využití tohoto skóre je již připravován systém obrany proti DDoS útokům provozovaný organizací CESNET. Za podobným účelem je plánováno využití FMP skóre i v rámci evropského projektu GN4[1]. Jednou z aktivit tohoto projektu je vývoj nové verze systému Firewall on Demand[2] (FoD), provozovaného organizací GÉANT, v němž by měla být v případě DDoS útoku automaticky navrhována pravidla pro blokování škodlivého provozu. FMP skóre ze systému NERD by mělo být jedním z klíčových kritérií určujících, které adresy či sítě budou při útoku navrženy k blokování [131].

Dále je FMP skóre poskytované systémem NERD využito v evropském projektu PROTECTIVE[3]. V tomto projektu, řešeném konsorciem deseti organizací z akademické i komerční sféry, je vyvíjen nástroj pro sdílení a pokročilou analýzu kyberbezpečnostních informací. FMP skóre je zde využito především jako jedno z kritérií pro prioritizaci hlášení, ale také je ve webovém rozhraní zobrazováno uživateli jako jedna ze základních informací o IP adresách.

Jako další pokračování práce jsou plánovány experimenty s metodami hlubokého učení (deep learning) za účelem zpřesnění predikce. Konkrétně je v plánu vzít jako vstupní data přímo sekvenci předchozích hlášení a využít rekurentních neuronových sítí typu *Long short-term memory* (LSTM). Tyto pokročilejší metody strojového učení by navíc mohly umožnit předpovídat nejen pravděpodobnost výskytu budoucích hlášení, ale i některé jejich vlastnosti, například předpokládaný počet útoků a jejich intenzitu nebo nejpravděpodobnější cíle.

[1] https://www.geant.org/Projects/GEANT_Project_GN4
[2] https://www.geant.org/Networks/Network_Operations/Pages/Firewall-on-Demand.aspx
[3] https://protective-h2020.eu/

Literatura

[1] ENISA: ENISA Threat Landscape Report 2017. Leden 2018, doi:10.2824/967192.
URL https://www.enisa.europa.eu/publications/enisa-threat-landscape-report-2017

[2] Symantec: 2018 Internet Security Threat Report. Březen 2018.
URL https://www.symantec.com/security-center/threat-report

[3] Cisco: Anual Cybersecurity Report. 2018.
URL
https://www.cisco.com/c/en/us/products/security/security-reports.html

[4] Groot, J. D.: The History of Data Breaches. Digital Guardian, listopad 2018, [online, citováno 9. 11. 2018].
URL https://digitalguardian.com/blog/history-data-breaches

[5] World's Biggest Data Breaches. 2018, [online, citováno 9. 11. 2018].
URL http://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/

[6] Ponemon Institute: The Rise of Ransomware. Research Report, leden 2017.
URL https://www.ponemon.org/library/the-rise-of-ransomware

[7] O'Brien, D.: Ransomware 2017. Internet Security Threat Report, Symantec, červenec 2017.
URL https://www.symantec.com/content/dam/symantec/docs/security-center/white-papers/istr-ransomware-2017-en.pdf

[8] Lord, N.: A History of Ransomware Attacks: The Biggest and Worst Ransomware Attacks of All Time. Digital Guardian, duben 2018, [online, citováno 9. 11. 2018].
URL https://digitalguardian.com/blog/history-ransomware-attacks-biggest-and-worst-ransomware-attacks-all-time

[9] O'Gorman, B.: Cryptojacking: A Modern Cash Cow. Internet Security Threat Report, Symantec, září 2018.
URL https://www.symantec.com/content/dam/symantec/docs/security-center/white-papers/istr-cryptojacking-modern-cash-cow-en.pdf

[10] Bloomberg, J.: Top Cyberthreat of 2018: Illicit Cryptomining. Forbes, březen 2018, [online, citováno 9. 11. 2018].
URL https://www.forbes.com/sites/jasonbloomberg/2018/03/04/top-cyberthreat-of-2018-illicit-cryptomining/

[11] Ponemon Institute: Third Annual Study on Exchanging Cyber Threat Intelligence: There Has to Be a Better Way. Research Report, leden 2018.
URL https://ponemonsullivanreport.com/2018/02/third-annual-study-on-exchanging-cyber-threat-intelligence-there-has-to-be-a-better-way/

[12] Collins, M. P.; Shimeall, T. J.; Faber, S.; aj.: Using Uncleanliness to Predict Future Botnet Addresses. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, IMC '07, New York, NY, USA: ACM, 2007, s. 93–104, doi:10.1145/1298306.1298319.

[13] Shue, C. A.; Kalafut, A. J.; Gupta, M.: Abnormally Malicious Autonomous Systems and Their Internet Connectivity. *IEEE/ACM Transactions on Networking*, ročník 20, č. 1, únor 2012: s. 220–230, ISSN 1063-6692, doi:10.1109/TNET.2011.2157699.

[14] van Wanrooij, W.; Pras, A.: Filtering spam from bad neighborhoods. *International Journal of Network Management*, ročník 20, č. 6, 2010: s. 433–444, doi:10.1002/nem.753.

[15] Kořenek, J.: Nové bezpečnostní projekty SIoT a FOCUS. Prezentace, Seminář o bezpečnosti, 2017, slide č. 4.
URL https://www.cesnet.cz/wp-content/uploads/2017/02/siot-focus.pdf

[16] Hofstede, R.; Čeleda, P.; Trammell, B.; aj.: Flow Monitoring Explained: From Packet Capture to Data Analysis With NetFlow and IPFIX. *IEEE Communications Surveys & Tutorials*, ročník 16, č. 4, 2014: s. 2037–2064, ISSN 1553-877X, doi:10.1109/COMST.2014.2321898.

[17] Trammell, B.; Boschi, E.: An introduction to IP flow information export (IPFIX). *IEEE Communications Magazine*, ročník 49, č. 4, duben 2011: s. 89–95, ISSN 0163-6804, doi:10.1109/MCOM.2011.5741152.

[18] Claise, B.; Trammell, B.; Aitken, P.: Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information. RFC 7011, září 2013.

[19] Velan, P.; Jirsík, T.; Čeleda, P.: Design and Evaluation of HTTP Protocol Parsers for IPFIX Measurement. In *Advances in Communication Networking (EUNICE'13)*, LNCS 8115, Springer Berlin Heidelberg, 2013, ISBN 978-3-642-40552-5, s. 136–147.

[20] Velan, P.: *Application-Aware Flow Monitoring*. Dizertační práce, Masarykova univerzita, Fakulta informatiky, Brno, 2018.

[21] Kekely, L.; Kučera, J.; Puš, V.; aj.: Software Defined Monitoring of Application Protocols. *IEEE Transactions on Computers*, ročník 65, č. 2, únor 2016: s. 615–626, ISSN 0018-9340, doi:10.1109/TC.2015.2423668.

[22] Cejka, T.; Bartos, V.; Truxa, L.; aj.: Using Application-Aware Flow Monitoring for SIP Fraud Detection. In *Intelligent Mechanisms for Network Configuration and Security (AIMS 2015)*, LNCS 9122, Springer, 2015, s. 87–99, doi:10.1007/978-3-319-20034-7_10.

[23] Jansky, T.; Cejka, T.; Bartos, V.: Hunting SIP Authentication Attacks Efficiently. In *Security of Networks and Services in an All-Connected World (AIMS 2017)*, LNCS 10356, Springer, 2017, doi:10.1007/978-3-319-60774-0_9.

[24] Bartoš, V.: Heartbleed Detection at CESNET using Extended Flow Monitoring. In *Proceedings of 8th International Scientific Conference on Security and Protection of Information*, 2015, ISSN 2336-5587.

[25] Cejka, T.; Bartos, V.; Svepes, M.; aj.: NEMEA: A Framework for Network Traffic Analysis. In *12th International Conference on Network and Service Management (CNSM 2016)*, IEEE, 2016, s. 195–201, doi:10.1109/CNSM.2016.7818417.

[26] Husák, M.; Velan, P.; Vykopal, J.: Security Monitoring of HTTP Traffic Using Extended Flows. In *Proceedings of the 2015 10th International Conference on Availability, Reliability and Security*, ARES'15, Washington, DC, USA: IEEE Computer Society, 2015, ISBN 978-1-4673-6590-1, s. 258–265, doi:10.1109/ARES.2015.42.

[27] Sperotto, A.; Schaffrath, G.; Sadre, R.; aj.: An Overview of IP Flow-Based Intrusion Detection. *IEEE Communications Surveys & Tutorials*, ročník 12, č. 3, 2010: s. 343–356, ISSN 1553-877X, doi:10.1109/SURV.2010.032210.00054.

[28] Münz, G.: *Traffic Anomaly Detection and Cause Identification Using Flow-Level Measurements*. Dizertační práce, Technische Universität München, 2010.

[29] Golling, M.; Hofstede, R.; Koch, R.: Towards multi-layered intrusion detection in high-speed networks. In *6th International Conference On Cyber Conflict (CyCon 2014)*, June 2014, s. 191–206, doi:10.1109/CYCON.2014.6916403.

[30] Hofstede, R.; Bartoš, V.; Sperotto, A.; aj.: Towards Real-Time Intrusion Detection for NetFlow and IPFIX. In *Proceedings of the 9th International Conference on Network and Service Management (CNSM 2013)*, IEEE, říjen 2013, ISSN 2165-9605, s. 227–234, doi:10.1109/CNSM.2013.6727841.

[31] Galtsev, A. A.; Sukhov, A. M.: Network Attack Detection at Flow Level. In *Proceedings of the 11th International Conference and 4th International Conference on Smart Spaces and Next Generation Wired/Wireless Networking*, NEW2AN'11/ruSMART'11, Springer, 2011, ISBN 978-3-642-22874-2, s. 326–334.

[32] Gao, Y.; Li, Z.; Chen, Y.: A DoS Resilient Flow-level Intrusion Detection Approach for High-speed Networks. In *26th IEEE International Conference on Distributed Computing Systems (ICDCS'06)*, červenec 2006, s. 39–39, doi:10.1109/ICDCS.2006.6.

[33] Cejka, T.; Svepes, M.: Analysis of Vertical Scans Discovered by Naive Detection. In *Management and Security in the Age of Hyperconnectivity (AIMS 2016)*, LNCS 9701, Springer, 2016, s. 165–169, doi:10.1007/978-3-319-39814-3_19.

[34] Huistra, D.: Detecting Reflection Attacks in DNS Flows. In *19th Twente Student Conference on IT*, 2013.

[35] Vykopal, J.; Drašar, M.; Winter, P.: Flow-based brute-force attack detection. *Fraunhofer Research Institution AISEC, Garching near Muenchen*, 2013: s. 41–51.

[36] Hofstede, R.; Hendriks, L.; Sperotto, A.; aj.: SSH Compromise Detection Using NetFlow/IPFIX. *SIGCOMM Computer Communication Review*, ročník 44, č. 5, říjen 2014: s. 20–26, ISSN 0146-4833, doi:10.1145/2677046.2677050.

[37] van der Toorn, O. I.; Hofstede, R.; Jonker, M.; aj.: A first look at HTTP(S) intrusion detection using NetFlow/IPFIX. *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, květen 2015: s. 862–865.

[38] Silveira, F.; Diot, C.; Taft, N.; aj.: ASTUTE: Detecting a Different Class of Traffic Anomalies. In *Proceedings of the ACM SIGCOMM 2010 Conference*, SIGCOMM '10, New York, NY, USA: ACM, 2010, ISBN 978-1-4503-0201-2, s. 267–278, doi:10.1145/1851182.1851215.

[39] Lakhina, A.; Crovella, M.; Diot, C.: Diagnosing Network-wide Traffic Anomalies. *SIGCOMM Comput. Commun. Rev.*, ročník 34, č. 4, srpen 2004: s. 219–230, ISSN 0146-4833, doi:10.1145/1030194.1015492.

[40] Li, X.; Bian, F.; Crovella, M.; aj.: Detection and Identification of Network Anomalies Using Sketch Subspaces. In *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement*, IMC '06, New York, NY, USA: ACM, 2006, ISBN 1-59593-561-4, s. 147–152, doi:10.1145/1177080.1177099.

[41] Carter, K. M.; Lippmann, R. P.; Boyer, S. W.: Temporally Oblivious Anomaly Detection on Large Networks Using Functional Peers. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, IMC '10, New York, NY, USA: ACM, 2010, ISBN 978-1-4503-0483-2, s. 465–471, doi:10.1145/1879141.1879201.

[42] Svoboda, J.; Ghafir, I.; Přenosil, V.: Network Monitoring Approaches: An Overview. *International Journal of Advances in Computer Networks and Its Security (IJCNS)*, ročník 5, č. 2, 2015, ISSN 2250-3757.

[43] Roesch, M.: Snort – Lightweight Intrusion Detection for Networks. In *Proceedings of the 13th USENIX Conference on System Administration*, LISA '99, Berkeley, CA, USA: USENIX Association, 1999, s. 229–238.

[44] Paxson, V.: Bro: A System for Detecting Network Intruders in Real-Time. *Computer Networks*, ročník 31, č. 23, 1999: s. 2435–2463, ISSN 1389-1286, doi:10.1016/S1389-1286(99)00112-7.

[45] Bhuyan, M. H.; Bhattacharyya, D. K.; Kalita, J. K.: Network Anomaly Detection: Methods, Systems and Tools. *IEEE Communications Surveys & Tutorials*, ročník 16, č. 1, 2014: s. 303–336, ISSN 1553-877X, doi:10.1109/SURV.2013.052213.00046.

[46] Spitzner, L.: *Honeypots: Tracking Hackers*. Addison Wesley, 2003, ISBN 0-321-10895-7.

[47] Mairh, A.; Barik, D.; Verma, K.; aj.: Honeypot in Network Security: A Survey. In *Proceedings of the 2011 International Conference on Communication, Computing & Security*, ICCCS'11, New York, NY, USA: ACM, 2011, ISBN 978-1-4503-0464-1, s. 600–605, doi:10.1145/1947940.1948065.

[48] Baykara, M.; Das, R.: A Survey on Potential Applications of Honeypot Technology in Intrusion Detection Systems. *International Journal of Computer Networks and Applications (IJCNA)*, ročník 2, č. 5, 2015, ISSN 2395-0455.

[49] ENISA: Actionable Information for Security Incident Response. Listopad 2014, doi:10.2824/38111.

[50] Goodwin, C.; Nicholas, J. P.: A framework for cybersecurity information sharing and risk reduction. Microsoft, 2015.

[51] Dandurand, L.; Serrano, O. S.: Towards improved cyber security information sharing. In *5th International Conference on Cyber Conflict (CYCON 2013)*, NATO CCD COE Publications, červen 2013, ISSN 2325-5374.

[52] Serrano, O.; Dandurand, L.; Brown, S.: On the Design of a Cyber Security Data Sharing System. In *Proceedings of the 2014 ACM Workshop on Information Sharing & Collaborative Security (WISCS)*, ACM, 2014, s. 61–69, doi:10.1145/2663876.2663882.

[53] Thomas, K.; Amira, R.; Ben-Yoash, A.; aj.: The Abuse Sharing Economy: Understanding the Limits of Threat Exchanges. In *Research in Attacks, Intrusions, and Defenses (RAID)*, LNCS 9854, Springer, 2016, s. 143–164, doi:10.1007/978-3-319-45719-2_7.

[54] ENISA: Standards and tools for exchange and processing of actionable information. Listopad 2014, doi:10.2824/37776.

[55] Steinberger, J.; Sperotto, A.; Golling, M.; aj.: How to exchange security events? Overview and evaluation of formats and protocols. In *Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management (IM 2015)*, IEEE Computer Society, 5 2015, s. 261–269, doi:10.1109/INM.2015.7140300.

[56] Sauerwein, C.; Sillaber, C.; Mussmann, A.; aj.: Threat Intelligence Sharing Platforms: An Exploratory Study of Software Vendors and Research Perspectives. In *Proceedings der 13. Internationalen Tagung Wirtschaftsinformatik (WI 2017)*, 2017, s. 837–851.

[57] Steinberger, J.; Sperotto, A.; Baier, H.; aj.: Collaborative Attack Mitigation and Response: A survey. In *Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management (IM 2015)*, IEEE Computer Society, 5 2015, s. 910–913, doi:10.1109/INM.2015.7140407.

[58] Fisk, G.; Ardi, C.; Pickett, N.; aj.: Privacy Principles for Sharing Cyber Security Data. In *2015 IEEE Security and Privacy Workshops*, květen 2015, s. 193–197, doi:10.1109/SPW.2015.23.

[59] Stupka, V.; Horák, M.; Husák, M.: Protection of personal data in security alert sharing platforms. In *Proceedings of the 12th International Conference on Availability, Reliability and Security*, ACM, 2017, ISBN 978-1-4503-5257-4, s. "65:1"–"65:8", doi:10.1145/3098954.3105822.

[60] Wagner, C.; Dulaunoy, A.; Wagener, G.; aj.: 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*, ACM, 2016, s. 49–56.

[61] Kacha, P.; Kostenec, M.; Kropacova, A.: Warden 3: Security Event Exchange Redesign. In *19th International Conference on Computers: Recent Advances in Computer Science*, 2015.

[62] Kacha, P.; Kostenec, M.; Kropacova, A.: Warden 3: Internet Threat Sharing Platform. *International Journal of Computers*, ročník 10, 2016, ISSN 1998-4308.

[63] Bartoš, V.: Analysis of alerts reported to Warden. Technická zpráva 1/2016, CESNET, únor 2016.

[64] Kácha, P.: IDEA: Designing the Data Model for Security Event Exchange. In *17th International Conference on Computers: Recent Advances in Computer Science*, 2013.

[65] Valeur, F.; Vigna, G.; Kruegel, C.; aj.: A Comprehensive Approach to Intrusion Detection Alert Correlation. *IEEE Transactions On Dependable and Secure Computing*, ročník 1, č. 3, červenec 2004: s. 146–169, ISSN 1545-5971.

[66] Salah, S.; Maciá-Fernández, G.; Díaz-Verdejo, J. E.: A model-based survey of alert correlation techniques. *Computer Networks*, ročník 57, č. 5, 2013: s. 1289–1317, ISSN 1389-1286, doi:10.1016/j.comnet.2012.10.022.

[67] Pouget, F.; Dacier, M.: Alert correlation: Review of the state of the art. Technická zpráva RR-03-093, Institut Eurecom, 12 2003.
URL http://www.eurecom.fr/publication/1271

[68] Mirheidari, S. A.; Arshad, S.; Jalili, R.: Alert Correlation Algorithms: A Survey and Taxonomy. In *Cyberspace Safety and Security*, LNCS 8300, Springer, 2013, ISBN 978-3-319-03584-0, s. 183–197, doi:10.1007/978-3-319-03584-0_14.

[69] Sadoddin, R.; Ghorbani, A.: Alert Correlation Survey: Framework and Techniques. In *Proceedings of the 2006 International Conference on Privacy, Security and Trust (PST'06): Bridge the Gap Between PST Technologies and Business Services*, ACM, 2006, ISBN 1-59593-604-1, s. 37:1–37:10, doi:10.1145/1501434.1501479.

[70] Alsubhi, K.; Aib, I.; Boutaba, R.: FuzMet: a fuzzy-logic based alert prioritization engine for intrusion detection systems. *International Journal of Network Management*, ročník 22, č. 4, září 2011: s. 263–284, ISSN 1099-1190, doi:10.1002/nem.804.

[71] Wallin, S.; Leijon, V.; Landén, L.: Statistical analysis and prioritisation of alarms in mobile networks. *International Journal of Business Intelligence and Data Mining*, ročník 4, č. 1, 2009: s. 4–21, ISSN 1743-8195, doi:10.1504/IJBIDM.2009.025408.

[72] Zomlot, L.; Sundaramurthy, S. C.; Luo, K.; aj.: Prioritizing Intrusion Analysis Using Dempster-Shafer Theory. In *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence*, AISec'11, ACM, 2011, ISBN 978-1-4503-1003-1, s. 59–70, doi:10.1145/2046684.2046694.

[73] PROTECTIVE: Meta-alerts ranking and prioritisation mechanisms report. Výzkumná zpráva, srpen 2017.
URL https://protective-h2020.eu/wp-content/uploads/2018/01/PROTECTIVE-D3.2-E-0817-Meta-Alerts-Ranking-and-Prioritisation-Mechanisms-Report.pdf

[74] Moura, G. C. M.; Sadre, R.; Pras, A.: Internet Bad Neighborhoods: the Spam Case. In *7th International Conference on Network and Services Management (CNSM 2011)*, IEEE, říjen 2011, ISBN 978-1-4577-1588-4, s. 56–63.

[75] Moura, G. C. M.; Sadre, R.; Sperotto, A.; aj.: Internet Bad Neighborhoods Aggregation. In *Proceedings of IEEE/IFIP Network Operations and Management Symposium (NOMS 2012)*, IEEE, duben 2012, ISBN 978-1-4673-0269-2, s. 343–350, doi:10.1109/NOMS.2012.6211917.

[76] Moura, G. C. M.; Sperotto, A.; Sadre, R.; aj.: Evaluating Third-Party Bad Neighborhood Blacklists for Spam Detection. In *Proceedings of IFIP/IEEE International Symposium on Integrated Network Management 2013*, IEEE, květen 2013, ISBN 978-1-4673-5229-1, s. 252–259.

[77] Moura, G. C. M.: *Internet Bad Neighborhoods*. Dizertační práce, University of Twente, Nizozemsko, březen 2013, doi:10.3990/1.9789036534604.

[78] Moura, G. C. M.; Sadre, R.; Pras, A.: Taking on Internet Bad Neighborhoods. In *Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS 2014)*, IEEE, květen 2014, ISBN 978-1-4799-0913-1, doi:10.1109/NOMS.2014.6838284.

[79] Moura, G. C. M.; Sadre, R.; Pras, A.: Internet Bad Neighborhoods Temporal Behavior. In *Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS 2014)*, IEEE, may 2014, ISBN 978-1-4799-0913-1, doi:10.1109/NOMS.2014.6838306.

[80] Zhang, J.; Chivukula, A.; Bailey, M.; aj.: Characterization of Blacklists and Tainted Network Traffic. In *Passive and Active Measurement*, LNCS 7799, Springer Berlin Heidelberg, 2013, s. 218–228, doi:10.1007/978-3-642-36516-4_22.

[81] Wahid, A.: Estimating the Internet malicious host population while preserving privacy. Disertační práce, The University of Melbourne, 2013.

[82] Bartoš, V.; Žádník, M.: An Analysis of Correlations of Intrusion Alerts in an NREN. In *19th International Workshop on Computer-Aided Modeling Analysis and Design of Communication Links and Networks (CAMAD)*, IEEE, prosinec 2014, s. 305–309.

[83] Zhang, J.; Porras, P.; Ullrich, J.: Highly Predictive Blacklisting. In *Proceedings of the 17th Conference on Security Symposium*, SS'08, Berkeley, CA, USA: USENIX Association, 2008, s. 107–122.

[84] Soldo, F.; Le, A.; Markopoulou, A.: Predictive Blacklisting as an Implicit Recommendation System. In *2010 Proceedings IEEE INFOCOM*, March 2010, ISSN 0743-166X, doi:10.1109/INFCOM.2010.5461982.

[85] Soldo, F.; Le, A.; Markopoulou, A.: Blacklisting Recommendation System: Using Spatio-Temporal Patterns to Predict Future Attacks. *IEEE Journal on Selected Areas in Communications*, ročník 29, 08 2011: s. 1423–1437.

[86] Ma, X.; Zhu, J.; Wan, Z.; aj.: Honeynet-based collaborative defense using improved highly predictive blacklisting algorithm. In *2010 8th World Congress on Intelligent Control and Automation*, červenec 2010, s. 1283–1288, doi:10.1109/WCICA.2010.5554909.

[87] Freudiger, J.; De Cristofaro, E.; Brito, A. E.: Controlled Data Sharing for Collaborative Predictive Blacklisting. In *Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA'15)*, LNCS 9148, Springer, 2015, ISBN 978-3-319-20550-2, s. 327–349, doi:10.1007/978-3-319-20550-2_17.

[88] Melis, L.; Pyrgelis, A.; Cristofaro, E. D.: Building and Measuring Privacy-Preserving Predictive Blacklists. arXiv e-print, 2015.
URL http://arxiv.org/abs/1512.04114v4

[89] Husák, M.; Kašpar, J.: Towards Predicting Cyber Attacks Using Information Exchange and Data Mining. In *2018 14th International Wireless Communications Mobile Computing Conference (IWCMC)*, červen 2018, ISSN 2376-6506, s. 536–541, doi:10.1109/IWCMC.2018.8450512.

[90] Dulaunoy, A.; Wagener, G.; Iklody, A.; aj.: An Indicator Scoring Method for MISP Platforms. In *The Networking Conference TNC'18*, GÉANT, 2018.

[91] Iklody, A.; Wagener, G.; Dulaunoy, A.; aj.: Decaying Indicators of Compromise. arXiv e-print, 2018.
URL http://arxiv.org/abs/1803.11052v1

[92] Ye, N.; Chen, Q.; Borror, C. M.: EWMA forecast of normal system activity for computer intrusion detection. *IEEE Transactions on Reliability*, ročník 53, č. 4, prosinec 2004: s. 557–566, ISSN 0018-9529, doi:10.1109/TR.2004.837705.

[93] Viinikka, J.; Debar, H.; Mé, L.; aj.: Processing Intrusion Detection Alert Aggregates with Time Series Modeling. *Information Fusion*, ročník 10, č. 4, říjen 2009: s. 312–324, ISSN 1566-2535, doi:10.1016/j.inffus.2009.01.003.

[94] Ling, C. X.; Sheng, V. S.: Class Imbalance Problem. In *Encyclopedia of Machine Learning*, editace C. Sammut; G. I. Webb, Boston, MA: Springer US, 2010, ISBN 978-0-387-30164-8, s. 171–171, doi:10.1007/978-0-387-30164-8_110.

[95] Class Imbalance Problem. 2013, [online, citováno 30. 9. 2018].
URL http://www.chioka.in/class-imbalance-problem/

[96] Chawla, N. V.; Bowyer, K. W.; Hall, L. O.; aj.: SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, ročník 16, 2002: s. 321–357.

[97] Dal Pozzolo, A.; Caelen, O.; Johnson, R. A.; aj.: Calibrating Probability with Undersampling for Unbalanced Classification. In *IEEE Symposium Series on Computational Intelligence*, IEEE, 2015, s. 159–166.

[98] Havránek, J.: Exportér síťových toků s podporou aplikačních informací. Bakalářská práce, ČVUT, 2017.

[99] Švepeš, M.: Systém pro konfiguraci a monitorování distribuovaného systému NEMEA. Bakalářská práce, ČVUT, 2014.

[100] Židek, M.: Unifikované konfigurační rozhraní pro NEMEA kolektory. Diplomová práce, ČVUT, 2018.

[101] Enns, R.; Bjorklund, M.; Schoenwaelder, J.; aj.: Network Configuration Protocol (NETCONF). RFC 6241, June 2011.

[102] Bartoš, V.; Žádník, M.; Čejka, T.: Nemea: Framework for stream-wise analysis of network traffic. CESNET technical report 6/2013.

[103] Gil, T. M.; Poletto, M.: MULTOPS: A Data-structure for Bandwidth Attack Detection. In *Proceedings of the 10th Conference on USENIX Security Symposium – Volume 10*, SSYM'01, Berkeley, CA, USA: USENIX Association, 2001.

[104] Hellemons, L.; Hendriks, L.; Hofstede, R.; aj.: SSHCure: A flow-based SSH intrusion detection system. In *Dependable Networks and Services (AIMS 2012)*, LNCS 7279, Springer, 2012, s. 86–97, doi:10.1007/978-3-642-30633-4_11.

[105] Čejka, T.; Rosa, Z.; Kubátová, H.: Stream-wise Detection of Surreptitious Traffic over DNS. In *Proc. of 19th IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, IEEE, 2014, doi:10.1109/CAMAD.2014.7033254.

[106] Cejka, T.; Bodó, R.; Kubatova, H.: Nemea: Searching for Botnet Footprints. In *The 3rd Prague Embedded Systems Workshop (PESW2015)*, 2015.

[107] US-CERT: UDP-Based Amplification Attacks. [online, citováno 11. 9. 2018]. URL https://www.us-cert.gov/ncas/alerts/TA14-017A

[108] Šabík, E.: Detekce těžení kryptoměn pomocí analýzy dat o IP tocích. Diplomová práce, VUT v Brně, 2017.

[109] Rosa, Z.; Čejka, T.; Žádník, M.; aj.: Building a feedback loop to capture evidence of network incidents. In *12th International Conference on Network and Service Management (CNSM 2016)*, IEEE, říjen 2016, ISSN 2165-963X, s. 292–296, doi:10.1109/CNSM.2016.7818435.

[110] Švepeš, M.; Čejka, T.: Making Flow-Based Security Detection Parallel. In *Security of Networks and Services in an All-Connected World (AIMS 2017)*, LNCS 10356, Springer, 2017, s. 3–15, doi:10.1007/978-3-319-60774-0_1.

[111] Čejka, T.; Žádník, M.: Preserving Relations in Parallel Flow Data Processing. In *Security of Networks and Services in an All-Connected World (AIMS 2017)*, LNCS 10356, Springer, 2017, s. 153–156, doi:10.1007/978-3-319-60774-0_14.

[112] Cejka, T.; Robledo, A.: Detecting Spoofed Time in NTP Traffic. In *The 4th Prague Embedded Systems Workshop (PESW2016)*, 2016.

[113] Švepeš, M.; Čejka, T.: Overload-resistant Network Traffic Analysis. In *The 4th Prague Embedded Systems Workshop (PESW2016)*, 2016.

[114] Čejka, T.; Švepeš, M.; Viktorin, J.: Gateway for IoT Security. In *The 5th Prague Embedded Systems Workshop (PESW2017)*, 2017.

[115] Šuster, F.; Čejka, T.: Stream-wise adaptive blacklist filter based on flow data. In *Proceedings of the 6th Prague Embedded Systems Workshop (PESW2018)*, 2018, s. 38–39.

[116] Slabihoudek, M.; Čejka, T.: Stream-wise Aggregation of Flow Data. In *Proceedings of the 6th Prague Embedded Systems Workshop (PESW2018)*, 2018, str. 37.

[117] Bartoš, V.; Kořenek, J.: Evaluating Reputation of Internet Entities. In *IFIP International Conference on Autonomous Infrastructure, Management and Security (AIMS'16)*, LNCS 9701, Springer, 2016, s. 132–136, doi:10.1007/978-3-319-39814-3_13.

[118] Bartoš, V.: Creating a Network Reputation Database. Poster, TNC'16 conference, 2016.

[119] Panjwani, S.; Tan, S.; Jarrin, K. M.; aj.: An experimental evaluation to determine if port scans are precursors to an attack. In *International Conference on Dependable Systems and Networks (DSN'05)*, IEEE, June 2005, ISSN 1530-0889, s. 602–611, doi:10.1109/DSN.2005.18.

[120] Paganini, P.: More than 900k routers of Deutsche Telekom German users went offline. Security Affairs, listopad 2016, [online, citováno 18. 7. 2018].
URL https://securityaffairs.co/wordpress/53871/iot/deutsche-telekom-hack.html

[121] ESET: "Petya" Ransomware: What we know now. červen 2017, [online, citováno 18. 7. 2018].
URL https://www.eset.com/us/about/newsroom/corporate-blog/petya-ransomware-what-we-know-now/

[122] List of countries by IPv4 address allocation. Wikipedia, [online, citováno 18. 7. 2018].
URL https://en.wikipedia.org/wiki/List_of_countries_by_IPv4_address_allocation

[123] Pitacairnovy ostrovy. Wikipedia, [online, citováno 18. 7. 2018].
URL https://cs.wikipedia.org/wiki/Pitcairnovy_ostrovy

[124] Durumeric, Z.; Wustrow, E.; Halderman, J. A.: ZMap: Fast Internet-Wide Scanning and its Security Applications. In *Proceedings of the 22nd USENIX conference on Security*, USENIX Association, 2013, s. 605–620.

[125] Theano Development Team: Theano: A Python framework for fast computation of mathematical expressions. arXiv e-print, květen 2016.
URL http://arxiv.org/abs/1605.02688

[126] Chen, T.; Guestrin, C.: XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2016, s. 785–794.

[127] Pedregosa, F.; Varoquaux, G.; Gramfort, A.; aj.: Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, ročník 12, 2011: s. 2825–2830.

[128] Fawcett, T.: ROC Graphs: Notes and Practical Considerations for Researchers. Technická zpráva, HP Laboratories, Palo Alto, CA, USA, 2004.

[129] Jánský, T.: Informovaná mitigace DDoS útoků na základě reputace. Diplomová práce, ČVUT, 2018.

[130] Jánský, T.; Čejka, T.; Žádník, M.; aj.: Augmented DDoS Mitigation with Reputation Scores. In *Proceedings of the 13th International Conference on Availability, Reliability and Security*, ARES 2018, New York, NY, USA: ACM, 2018, doi:10.1145/3230833.3233279.

[131] Schmitz, D.; Čejka, T.; Bartoš, V.; aj.: Distributed Denial of Service Mitigation v1.0 Pilot. GN4-2 deliverable D8.3, GÉANT, 2017.
URL https://www.geant.org/Projects/GEANT_Project_GN4/deliverables/D8.3_Distributed-Denial-of-Service-Mitigation-v1.0-Pilot.pdf

Příloha A

Seznam atributů feature vectoru

(tabulka následuje na několika dalších stránkách)

č.	Název	Popis	Rozsah	Transf.
1	`scan_alerts_1d`	Počet hlášení typu *skenování* o dané adrese přijatých v posledním dni (24 hod).	$[0, \infty)$	$log(x+1)$
2	`scan_conns_1d`	Celkový počet pokusů o spojení (objem útoku) v hlášeních typu *skenování* o dané adrese přijatých v posledním dni (24 hod).	$[0, \infty)$	$log(x+1)$
3	`scan_nodes_1d`	Počet různých detektorů (hodnot `Node[-1].Name` v IDEA zprávě) v hlášeních typu *skenování* o dané adrese přijatých v posledním dni (24 hod).	$[0, \infty)$	$log(x+1)$
4	`scan_alerts_7d`	Počet hlášení typu *skenování* o dané adrese přijatých v posledních 7 dnech.	$[0, \infty)$	$log(x+1)$
5	`scan_conns_7d`	Celkový počet pokusů o spojení (objem útoku) v hlášeních typu *skenování* o dané adrese přijatých v posledních 7 dnech.	$[0, \infty)$	$log(x+1)$
6	`scan_nodes_7d`	Počet různých detektorů (hodnot `Node[-1].Name` v IDEA zprávě) v hlášeních typu *skenování* o dané adrese přijatých v posledních 7 dnech.	$[0, \infty)$	$log(x+1)$
7	`scan_alerts_ewma`	Plovoucí průměr EWMA z počtu hlášení za den za posledních 7 dní. Počítána jsou pouze hlášení typu *skenování* o dané adrese. Parametr (tzv. *smoothing factor*) $\alpha = 0.25$.	$[0, \infty)$	$log(x+1)$
8	`scan_conns_ewma`	Plovoucí průměr EWMA z počtu pokusů o spojení za den za posledních 7 dní. Počítána jsou pouze spojení v hlášeních typu *skenování* o dané adrese. Parametr $\alpha = 0.25$.	$[0, \infty)$	$log(x+1)$
9	`scan_binalerts_ewma`	Plovoucí průměr EWMA z hodnot b za posledních 7 dní, kde $b = 1$ pokud bylo v daném dni přijato alespoň jedno hlášení, jinak $b = 0$. Uvažována jsou pouze hlášení typu *skenování* o dané adrese. Parametr $\alpha = 0.25$.	$[0, 1]$	—
10	`scan_last_alert_age`	Počet dnů (reálné číslo) mezi posledním přijatým hlášením a časem predikce (t_0). Uvažována jsou pouze hlášení typu *skenování* o dané adrese přijatá v posledních 7 dnech.	$[0, 7]$	$exp(-x)$
11	`scan_avg_interval`	Průměrná délka intervalu (v počtu dnů) mezi po sobě jdoucími hlášeními v předchozích sedmi dnech. Nekonečno, pokud byla přijata méně než dvě hlášení. Uvažována jsou pouze hlášení typu *skenování* o dané adrese.	$[0, \infty)$	$exp(-x)$
12	`scan_avg_interval`	Medián délek intervalů (v počtu dnů) mezi po sobě jdoucími hlášeními v předchozích sedmi dnech. Nekonečno, pokud byla přijata méně než dvě hlášení. Uvažována jsou pouze hlášení typu *skenování* o dané adrese.	$[0, \infty)$	$exp(-x)$
13	`prefix_scan_alerts_1d`	Počet hlášení typu *skenování* o adresách ve stejném /24 prefixu přijatých v posledním dni (24 hod).	$[0, \infty)$	$log(x+1)$
14	`prefix_scan_conns_1d`	Celkový počet pokusů o spojení (objem útoku) v hlášeních typu *skenování* o adresách ve stejném /24 prefixu přijatých v posledním dni (24 hod).	$[0, \infty)$	$log(x+1)$

č.	Název	Popis	Rozsah	Transf.
15	`prefix_scan_conns_1d`	Počet různých IP adres v hlášeních typu *skenování* o adresách ve stejném /24 prefixu přijatých v posledním dni (24 hod).	$[0, 256]$	$log(x+1)$
16	`prefix_scan_nodes_1d`	Počet různých detektorů (hodnot `Node[-1].Name` v IDEA zprávě) v hlášeních typu *skenování* o adresách ve stejném /24 prefixu přijatých v posledním dni (24 hod).	$[0, \infty)$	$log(x+1)$
17	`prefix_scan_alerts_7d`	Počet hlášení typu *skenování* o adresách ve stejném /24 prefixu přijatých v posledních 7 dnech.	$[0, \infty)$	$log(x+1)$
18	`prefix_scan_conns_7d`	Celkový počet pokusů o spojení (objem útoku) v hlášeních typu *skenování* o adresách ve stejném /24 prefixu přijatých v posledních 7 dnech.	$[0, \infty)$	$log(x+1)$
19	`prefix_scan_conns_7d`	Počet různých IP adres v hlášeních typu *skenování* o adresách ve stejném /24 prefixu přijatých v posledních 7 dnech.	$[0, 256]$	$log(x+1)$
20	`prefix_scan_nodes_7d`	Počet různých detektorů (hodnot `Node[-1].Name` v IDEA zprávě) v hlášeních typu *skenování* o adresách ve stejném /24 prefixu přijatých v posledních 7 dnech.	$[0, \infty)$	$log(x+1)$
21	`prefix_scan_alerts_ewma`	Plovoucí průměr EWMA z počtu hlášení za den za posledních 7 dní. Počítána jsou pouze hlášení typu *skenování* o adresách ve stejném /24 prefixu. Parametr (tzv. *smoothing factor*) $\alpha = 0.25$.	$[0, \infty)$	$log(x+1)$
22	`prefix_scan_conns_ewma`	Plovoucí průměr EWMA z počtu pokusů o spojení za den za posledních 7 dní. Počítána jsou pouze spojení v hlášeních typu *skenování* o adresách ve stejném /24 prefixu. Parametr $\alpha = 0.25$.	$[0, \infty)$	$log(x+1)$
23	`prefix_scan_binalerts_ewma`	Plovoucí průměr EWMA z hodnot b za posledních 7 dní, kde $b = 1$ pokud bylo v daném dni přijato alespoň jedno hlášení, jinak $b = 0$. Uvažována jsou pouze hlášení typu *skenování* o adresách ve stejném /24 prefixu. Parametr $\alpha = 0.25$.	$[0, 1]$	—
24	`access_alerts_1d`	Počet hlášení typu *přístup* o dané adrese přijatých v posledním dni (24 hod).	$[0, \infty)$	$log(x+1)$
25	`access_conns_1d`	Celkový počet pokusů o spojení (objem útoku) v hlášeních typu *přístup* o dané adrese přijatých v posledním dni (24 hod).	$[0, \infty)$	$log(x+1)$
26	`access_nodes_1d`	Počet různých detektorů (hodnot `Node[-1].Name` v IDEA zprávě) v hlášeních typu *přístup* o dané adrese přijatých v posledním dni (24 hod).	$[0, \infty)$	$log(x+1)$
27	`access_alerts_7d`	Počet hlášení typu *přístup* o dané adrese přijatých v posledních 7 dnech.	$[0, \infty)$	$log(x+1)$
28	`access_conns_7d`	Celkový počet pokusů o spojení (objem útoku) v hlášeních typu *přístup* o dané adrese přijatých v posledních 7 dnech.	$[0, \infty)$	$log(x+1)$
29	`access_nodes_7d`	Počet různých detektorů (hodnot `Node[-1].Name` v IDEA zprávě) v hlášeních typu *přístup* o dané adrese přijatých v posledních 7 dnech.	$[0, \infty)$	$log(x+1)$

č.	Název	Popis	Rozsah	Transf.
30	`access_alerts_ewma`	Plovoucí průměr EWMA z počtu hlášení za den za posledních 7 dní. Počítána jsou pouze hlášení typu *přístup* o dané adrese. Parametr (tzv. *smoothing factor*) $\alpha = 0.25$.	$[0, \infty)$	$log(x+1)$
31	`access_conns_ewma`	Plovoucí průměr EWMA z počtu pokusů o spojení za den za posledních 7 dní. Počítána jsou pouze spojení v hlášeních typu *přístup* o dané adrese. Parametr $\alpha = 0.25$.	$[0, \infty)$	$log(x+1)$
32	`access_binalerts_ewma`	Plovoucí průměr EWMA z hodnot b za posledních 7 dní, kde $b = 1$ pokud bylo v daném dni přijato alespoň jedno hlášení, jinak $b = 0$. Uvažována jsou pouze hlášení typu *přístup* o dané adrese. Parametr $\alpha = 0.25$.	$[0, 1]$	—
33	`access_last_alert_age`	Počet dnů (reálné číslo) mezi posledním přijatým hlášením a časem predikce (t_0). Uvažována jsou pouze hlášení typu *přístup* o dané adrese přijatá v posledních 7 dnech.	$[0, 7]$	$exp(-x)$
34	`access_avg_interval`	Průměrná délka intervalu (v počtu dnů) mezi po sobě jdoucími hlášeními v předchozích sedmi dnech. Nekonečno, pokud byla přijata méně než dvě hlášení. Uvažována jsou pouze hlášení typu *přístup* o dané adrese.	$[0, \infty)$	$exp(-x)$
35	`access_avg_interval`	Medián délek intervalů (v počtu dnů) mezi po sobě jdoucími hlášeními v předchozích sedmi dnech. Nekonečno, pokud byla přijata méně než dvě hlášení. Uvažována jsou pouze hlášení typu *přístup* o dané adrese.	$[0, \infty)$	$exp(-x)$
36	`prefix_access_alerts_1d`	Počet hlášení typu *přístup* o adresách ve stejném /24 prefixu přijatých v posledním dni (24 hod).	$[0, \infty)$	$log(x+1)$
37	`prefix_access_conns_1d`	Celkový počet pokusů o spojení (objem útoku) v hlášeních typu *přístup* o adresách ve stejném /24 prefixu přijatých v posledním dni (24 hod).	$[0, \infty)$	$log(x+1)$
38	`prefix_access_conns_1d`	Počet různých IP adres v hlášeních typu *přístup* o adresách ve stejném /24 prefixu přijatých v posledním dni (24 hod).	$[0, 256]$	$log(x+1)$
39	`prefix_access_nodes_1d`	Počet různých detektorů (hodnot `Node[-1].Name` v IDEA zprávě) v hlášeních typu *přístup* o adresách ve stejném /24 prefixu přijatých v posledním dni (24 hod).	$[0, \infty)$	$log(x+1)$
40	`prefix_access_alerts_7d`	Počet hlášení typu *přístup* o adresách ve stejném /24 prefixu přijatých v posledních 7 dnech.	$[0, \infty)$	$log(x+1)$
41	`prefix_access_conns_7d`	Celkový počet pokusů o spojení (objem útoku) v hlášeních typu *přístup* o adresách ve stejném /24 prefixu přijatých v posledních 7 dnech.	$[0, \infty)$	$log(x+1)$
42	`prefix_access_conns_7d`	Počet různých IP adres v hlášeních typu *přístup* o adresách ve stejném /24 prefixu přijatých v posledních 7 dnech.	$[0, 256]$	$log(x+1)$

č.	Název	Popis	Rozsah	Transf.
43	`prefix_access_nodes_7d`	Počet různých detektorů (hodnot `Node[-1].Name` v IDEA zprávě) v hlášeních typu *přístup* o adresách ve stejném /24 prefixu přijatých v posledních 7 dnech.	$[0, \infty)$	$log(x+1)$
44	`prefix_access_alerts_ewma`	Plovoucí průměr EWMA z počtu hlášení za den za posledních 7 dní. Počítána jsou pouze hlášení typu *přístup* o adresách ve stejném /24 prefixu. Parametr (tzv. *smoothing factor*) $\alpha = 0.25$.	$[0, \infty)$	$log(x+1)$
45	`prefix_access_conns_ewma`	Plovoucí průměr EWMA z počtu pokusů o spojení za den za posledních 7 dní. Počítána jsou pouze spojení v hlášeních typu *přístup* o adresách ve stejném /24 prefixu. Parametr $\alpha = 0.25$.	$[0, \infty)$	$log(x+1)$
46	`prefix_access_binalerts_ewma`	Plovoucí průměr EWMA z hodnot b za posledních 7 dní, kde $b = 1$ pokud bylo v daném dni přijato alespoň jedno hlášení, jinak $b = 0$. Uvažována jsou pouze hlášení typu *přístup* o adresách ve stejném /24 prefixu. Parametr $\alpha = 0.25$.	$[0, 1]$	—
47	`blocklist_de_ssh`	1 pokud je adresa na SSH blacklistu služby blocklist.de, jinak 0. Zdroj blacklistu: https://lists.blocklist.de/lists/ssh.txt	$\{0, 1\}$	—
48	`uceprotect`	1 pokud je adresa na blacklistu UCEPROTECT (level 1), jinak 0. Blacklist je denně stahován promocí *rsync* z neveřejné adresy.	$\{0, 1\}$	—
49	`sorbs-dul`	1 pokud je adresa na blacklistu SORBS „dul“ (seznam dynamicky přidělovaných IP rozsahů), jinak 0. Blacklist je dotazován prostřednictvím DNSBL na adrese `dnsbl.sorbs.net`.	$\{0, 1\}$	—
50	`spamhaus-pbl`	1 pokud je adresa na blacklistu PBL společnosti Spamhaus, jinak 0. Blacklist je dotazován prostřednictvím DNSBL na adrese `zen.spamhaus.org`.	$\{0, 1\}$	—
51	`spamhaus-pbl-isp`	1 pokud je adresa na blacklistu PBL-ISP společnosti Spamhaus, jinak 0. Blacklist je dotazován prostřednictvím DNSBL na adrese `zen.spamhaus.org`.	$\{0, 1\}$	—
52	`spamhaus-xbl-cbl`	1 pokud je adresa na blacklistu XBL-CBL společnosti Spamhaus, jinak 0. Blacklist je dotazován prostřednictvím DNSBL na adrese `zen.spamhaus.org`.	$\{0, 1\}$	—
53	`hostname_exists`	1 pokud se prostřednictvím DNS dotazu typu PTR podařilo zjistit doménové jméno přiřazené dané adrese, jinak 0.	$\{0, 1\}$	—
54	`dynamic_static`	1 pokud zjištěné doménové jméno odpovídá regulárnímu výrazu `\bdyn(amic)?|pool|dial-?up\b` (jde tedy pravděpodobně o dynamicky přidělovanou adresu), -1 pokud odpovídá regulárnímu výrazu `\bstatic\b` (jde tedy pravděpodobně o staticky přidělovanou adresu), jinak 0.	$\{-1, 0, 1\}$	—

č.	Název	Popis	Rozsah	Transf.
55	`dsl`	1 pokud zjištěné doménové jméno odpovídá regulárnímu výrazu `\b(broad(band)?|[avx]dsl|cable)\b` (jde tedy pravděpodobně o DSL přípojku), jinak 0.	$\{0, 1\}$	—
56	`ip_in_hostname`	1 pokud zjištěné doménové jméno obsahuje alespoň dva oktety dané IP adresy (tzn. doménové jméno je pravděpodobně automaticky generované z IP adresy), jinak 0.	$\{0, 1\}$	—
57	`ctry_badness`	Poměr počtu škodlivých IP adres vůči všem IP adresám patřícím do stejné země, jako vyhodnocovaná adresa. Jako škodlivé jsou uvažovány ty adresy, k nimž bylo v posledním týdnu přijato alespoň jedno hlášení libovolného typu.	$[0, 1]$	—
58	`asn_badness`	Poměr počtu škodlivých IP adres vůči všem IP adresám patřícím do stejného ASN, jako vyhodnocovaná adresa. Jako škodlivé jsou uvažovány ty adresy, k nimž bylo v posledním týdnu přijato alespoň jedno hlášení libovolného typu.	$[0, 1]$	—