Univerzita Hradec Králové

Fakulta informatiky a managementu

Katedra informatiky a kvantitativních metod

Automatizace v síti poskytovatele internetu

Diplomová práce

Autor práce: Bc. Jindřich Vrba, DiS.
Studijní obor: Aplikovaná Informatika

Vedoucí práce: Ing. Pavel Kříž, Ph.D.

Hradec Králové 2022

Prohlášení

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a s použitím uvedené literatury

.....

Jindřich Vrba

28. dubna 2022

Poděkování

Rád bych poděkoval vedoucímu práce Ing. Pavlu Křížovi, Ph.D. za odborné vedení, trpělivost a podnětné připomínky při vedení mé diplomové práce. Dále bych chtěl poděkovat rodině a přátelům za podporu a trpělivost při mém studiu. Mé díky patří také Stanislavu Toufarovi za korekturu.

Anotace

Tato diplomová práce se zabývá automatizací v síti poskytovatele internetu, konkrétně přetížením rádiových spojů a síťovými útoky. Dané problémy jsou rozebrány teoreticky a jsou představeny jejich aktuální možnosti řešení. Důležitý cíl této práce je implementace aplikací, které tyto problémy řeší. Pro eliminaci přetížení rádiových spojů je vytvořena aplikace, která v případě detekce takové situace, rozděluje dostupnou šíři pásma mezi uživatele pomocí řízeného shapingu a zajišťuje tak jednotlivým uživatelům přijatelnou odezvu. Pro detekci síťových útoků je implementována aplikace, která sleduje hlavičky datové komunikace v rámci celé sítě poskytovatele. Tato data vyhodnocuje, detekuje útoky, které buď blokuje nebo na ně upozorňuje obsluhu. Aplikace jsou napsány v jazyce Python a vytvořené řešení běží na systému GNU/Linux u poskytovatele internetu.

Annotation

Title: Automation in Internet Service Provider's network

This diploma thesis deals with automation in the network of an internet provider, specifically congestion of radio links and network attacks. The given problems are analyzed theoretically and their current possible solutions are presented. An important goal of this work is the implementation of applications that solve these problems. To eliminate radio link congestion, an application is created, which, if such a situation is detected, distributes the available bandwidth among the users by means of traffic shaping and thus ensures an acceptable response for the individual users. An application that monitors data communication headers within the provider's entire network is implemented to detect network attacks. It evaluates this data, detects attacks, either blocks them or notifies the operator. The applications are written in Python and the resulting solution runs on GNU/Linux at internet provider's network.

Klíčová slova

automatizace, ISP, bezdrátové sítě, WiFi, shaping, rušení, zahlcení, síťové útoky, zabezpečení, NetFlow, Python, Linux

Keywords

automation, ISP, wireless networks, WiFi, shaping, interference, congestion, network attacks, security, NetFlow, Python, Linux

Citace

VRBA, Jindřich. *Automatizace v síti poskytovatele internetu.* Hradec Králové, 2022. Diplomová práce. Univerzita Hradec Králové, Fakulta informatiky a managementu. Vedoucí práce Ing. Pavel Kříž, Ph.D.

Obsah

Seznam obrázků

Seznam tabulek

1 Úvod

Internet se čím dál více stává nedílnou součástí našich životů. Používáme jej v práci i soukromém životě, na komunikaci, vyhledávání informací, zábavu, zálohování dat, aktualizace a na mnoho dalších činností. Internet obecně využíváme stále více, a tak i požadavky na počítačové sítě stále rostou a to jak z ohledu přenosové kapacity tak kvality.

K internetu připojujeme mnoho zařízení, která by bez něj ani nemohla naplno sloužit svému účelu. Chytré telefony, hodinky a další osobní zařízení, kamerové systémy, lednice a jiná vybavení domácnosti, automobily, letadla, lékařské zařízení apod. Na uživatele a všechna jejich zařízení na dnešním internetu číhá spoustu nástrah a nebezpečí, před kterými se musí chránit.

V této práci se na problém podíváme z pohledu poskytovatele internetu – *Internet Service Provider* (*ISP*). Budeme se zabývat tím, jak je možné pomocí automatizace zlepšit kvalitu služeb dodávaných koncovým zákazníkům. Práce je rozdělena na dvě části, v první z nich se budeme zabývat detekcí a eliminací přetížení rádiových spojů, v druhé pak detekcí síťových útoků. Obě části budou mít za cíl vytvořit vlastní řešení těchto problémů. Společný rys těchto řešení bude nasazení automatizace s využitím operačního systému *Linux*, jazyka *Python* a zařízení v síti poskytovatele.

Na bezdrátových sítích budeme zkoumat, jak v reálném prostředí dodat co nejlepší službu zákazníkovi. Především nelicencované bezdrátové spoje mění během dne svoji kapacitu, při tom jak se mění jejich využití uživateli a rušení z okolí. Představíme situaci, možná řešení problému a vlastní řešení, které přetížení rádiových spojů detekuje a v rámci parametrů tarifu také eliminuje pomocí řízeného shapingu provozu s cílem přijatelné odezvy.

V detekci útoků se budeme zabývat tím, jaké síťové útoky uživatelům a jejich zařízením hrozí, představíme existující metody jejich detekce a možnosti, jak jim lze zabránit. Budeme se zabývat především útoky, které jsou detekovatelné bez nahlížení do obsahu paketů, tedy jen z jejich hlaviček, množství a frekvence navazovaných spojení, počtu protistran a podobně. Představíme vlastní řešení, které dokáže útoky z internetu detekovat a provoz z útočících zdrojových IP adres blokovat. Vyvinuté řešení bude zároveň detekovat útoky z napadených zařízení ve vnitřní síti.

2 Cíl práce

Cílem této práce je prozkoumat možnosti automatizace v síti *ISP*, které by vedly ke zlepšení dodávaných služeb zákazníkům. První část se bude věnovat detekci a eliminaci přetížení bezdrátových spojů, druhá pak síťovým útokům.

Dané problémy budou rozebrány teoreticky a budou představeny i aktuální možnosti řešení těchto problémů. Hlavní cíl této práce je vytvořit aplikace, které tyto problémy budou řešit.

Pro eliminaci přetížení bezdrátových sítí je cílem vytvořit aplikaci, která bude schopna přetížení detekovat a problém řešit prostřednictvím rozdělení dostupné šířky pásma mezi uživatele pomocí řízeného shapingu, čímž zajistí uživatelům přijatelnou odezvu.

Další praktický cíl je vytvořit aplikaci, která bude detekovat síťové útoky z internetu v rámci celé sítě *ISP* a dokáže těmto útokům zamezit, aniž by to mělo vliv na legitimní komunikaci zákazníků. Tato aplikace by zároveň měla detekovat útoky z vnitřní sítě a pomáhat tak technikům sítě při odhalování problematických míst.

3 Detekce a eliminace přetížení rádiových spojů

ISP se musejí vypořádat kromě dostatečně silné konektivity na páteřních linkách také s tzv. poslední mílí, tedy s posledním úsekem nejblíže k uživateli internetu. Tato poslední míle může být realizována například optickým vláknem, metalicky a nebo také bezdrátově. Pro bezdrátové připojení je možné použít jak licencovaná, tak i volná pásma. Za volná pásma se sice neplatí licenční poplatky, může je ale využívat kdokoliv, a tak je nutné zde počítat i se vzájemným rušením.

U jakékoliv technologie musí *ISP* dbát na dostatečné dimenzování kapacity těchto linek tak, aby dokázal v jakékoliv situaci uspokojit požadavky uživatelů na kapacitu, alespoň dle jejich garantovaných rychlostí. Pro optická a metalická vedení a bezdrátové spoje v licencovaných pásmech je situace relativně jednoduchá, kapacita dané linky je dána a tak je jen potřeba dbát na to, aby součet garantovaných rychlostí zákazníků v dané oblasti nepřekročil kapacitu dané technologie.

Horší je situace u bezdrátových technologií ve volných pásmech, kde kapacita linky pevně dána není, neboť se mění v závislosti na rušení z okolí. Toto rušení může být kromě jiného způsobeno různými elektronickými zařízeními, odrazem, nebo bezdrátovým přenosem na stejné nebo blízké frekvenci jiného uživatele, ať už připojeného od stejného nebo jiného *ISP*.

Dostupná kapacita pro jednotlivého uživatele se tedy na sdílených linkách může výrazně měnit. Asi nejvýraznější je tento stav ve městech, neboť zde je mnoho *ISP* i uživatelů internetu a tedy i větší rušení. V době největšího vytížení se uživatelé volných frekvencí ruší navzájem a najít volný kanál je prakticky nemožné.

V zájmu poskytovatele je, aby se zákazníci vzájemně nerušili a měli dostupnou co nejvyšší kapacitu. Jak toho ale v zarušených oblastech dosáhnout?

Autoři z německé univerzity ve své studii Interference of Simulated IEEE 802.11 Links with Directional Antennas [21] zjistili, že i v případě směrových antén ve venkovských oblastech bývá hlavním problémem rušení, často způsobené vyzařováním postranními laloky, které se v modelech většinou zanedbává. Toto rušení pak samozřejmě výrazným způsobem ovlivňuje přenosovou rychlost a kvalitu linky.

Další autoři popisují problematiku rušení jako hlavní důvod chybovosti a zhoršování *Quality of Service* (*QoS*), především na bezdrátových sítích ve volných pásmech, jako důsledek nasazování stále většího množství vzájemně se překrývajících sítí. [27]

Další článek se zabývá vlivem rušení na chování *Wireless Mesh Network* (*WMN*) sítí, autoři dochází k tomu, že rušení je jeden z nejvýraznějších problémů bezdrátových sítí, způsobujících výraznou degradaci přenosu, nízké hodnoty *Signal to Interference and Noise Ratio* (*SINR*), velmi vysoké zpoždění a ztrátu dat. [26]

S těmito problémy se na bezdrátových sítích potýkají *ISP* obecně, nejvýraznější je pak tato situace v místech, kde je bezdrátových sítí vysoká koncentrace. *ISP* se snaží umísťovat svoje vysílače tak, aby nerušili jiné své vysílače, jejich konkurenti se snaží o to samé, ale jelikož *ISP* je na daném místě mnoho, ruší se navzájem a tak se snaží přejít na jiné frekvence, které nejsou tolik vytížené. Přechodem na jinou frekvenci pak dochází ke změně, na kterou reagují další *ISP* a spouští se tak nové kolečko změn frekvencí. Někteří *ISP* přecházejí na jiné nelicencované bezdrátové technologie na jiných frekvencích, jiní na licencované, další se snaží o zajištění co nejhustší sítě vysílačů, tak aby se zákazníci připojovali na kratší vzdálenosti. Do toho všeho však přichází problémy s odrazem signálu a vyzařování postranními laloky.

Často je složitá situace i na vysílacích bodech. Antény pro připojení zákazníků chtějí všichni *ISP* umístit na co nejlépe viditelné místo pro co nejširší okolí. Takové místa bývají střechy panelových domů nebo jiných vysokých budov, komíny apod. Takových míst bývá jen omezené množství, někdy dokonce jen jedno jediné ideální. Více *ISP* je pak nuceno sdílet jedno místo o které se spolu musí nějak rozumně rozdělit, aby mohli všichni poskytovat zamýšlené služby.

3.1 Analýza problému

Uživatelé požadují kvalitní vysokorychlostní přenos, kterého ale není ve všech případech možné jednoduše dosáhnout z důvodů popsaných v předchozí kapitole. Jaké jsou tedy možnosti, aby co nejlépe dosáhli požadované kvality?

Jedna z možností, jak se vyvarovat rušení, je nevysílat na stejné frekvenci, tedy zvolit jiný kanál. Práce *Analysis and Experimental Verification of Frequency-Based Interference Avoidance Mechanisms in IEEE 802.15.4* [27] mimo jiné zkoumá různé způsoby výběru

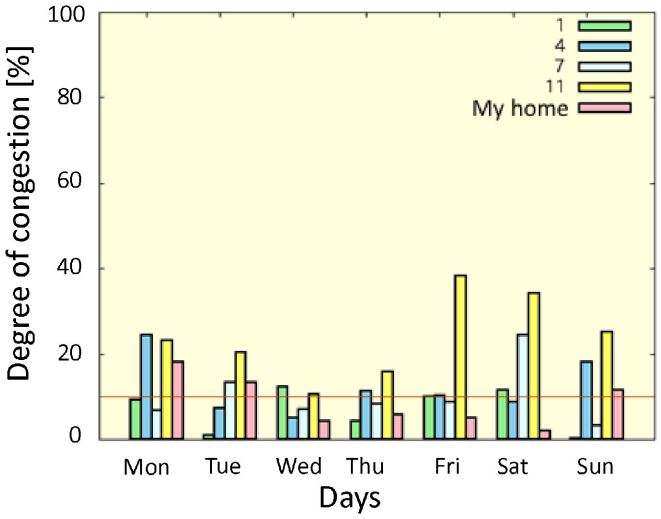
vhodného kanálu. Autoři tyto metody testují v reálném prostředí a porovnávají dosažené výsledky. Dále představují vlastní vyvinutou metodu, kterou prověřují nejen pro jednorázové změny kanálu, ale i pro přepínání kanálu v reálném čase.

Dalším důležitým prvkem je výběr správné přenosové rychlosti dané technologie. Tento výběr silně závisí na kvalitě linky a její stabilitě. Od doby, kdy standard IEEE 802.11 představil podporu více přenosových rychlostí, proběhlo mnoho výzkumů v oblasti algoritmů výběru vhodné přenosové rychlosti. Tyto algoritmy pomáhají dosáhnout lepší kvality spojení na bezdrátových sítích. Málokdy bývají specifikovány přímo ve standardu a jsou tak většinou nechány na implementaci samotným výrobcem. [23]

Další článek se zabývá nebývalou úrovní zarušení ve volných pásmech především v hustě zalidněných oblastech. Jako pomocnou ruku pro podporu optimalizace využití bezdrátových sítí autoři používají mapu bezdrátových sítí (*Radio Environment Maps*). S pomocí této mapy detekují překrývající se spoje a mezery v pokrytí. Vhodnou strategií pak tyto informace využívají ke změně používaného kanálu na vysílacích bodech. Optimalizují tak přiřazení jednotlivých klientů mezi dostupné vysílací body. Za cenu snížení celkové propustnosti sítě tak dosahují zlepšení na nejvíce přetížených bodech. K nasazení této metody je potřeba relativně hustá síť senzorů, neboť každý vysílací bod potřebuje i svůj senzor, což zvyšuje cenu řešení. [13]

Další autoři se zabývají domácími sítěmi na frekvenci 2,4 GHz. Jako nedostačující hodnotí automatické nastavení kanálu v běžně dostupných zařízeních, provádějící výběr kanálu jen dle *Received Signal Strength Indication* (*RSSI*) bez ohledu na množství skutečného provozu, navíc zjišťující danou situaci jen jednorázově při startu zařízení. Zabývají se především, jak běžnému uživateli doporučit vhodné nastavení kanálu pomocí systému, který by danou oblast monitoroval 24 hodin denně. Příklad detekovaných informací je na obrázku 1. Autoři se dále zabývají detekcí streamování videa z důvodu očekávaného dlouhodobého vytížení daného kanálu v porovnání např. s běžným surfování po webu. Oblast monitorují pomocí levného jednodeskového počítače s bezdrátovým síťovým adaptérem v módu monitor. Stav linky hodnotí především podle příznaků retry v hlavičkách rámců. [18]

Další práce se zabývá zajištěním spolehlivosti Real-Time *Wireless Fidelity* (*WiFi*). Autoři očekávají rušení jak z *WiFi* sítí, tak i mimo ně. Rušení mimo *WiFi* sítě řeší přechodem na nižší přenosové rychlosti, případě opakováním vysílání. Rušení z *WiFi* sítí zde navrhují



Obrázek 1: Graf zahlcení kanálů v jednotlivých dnech. [18]

řešit pomocí virtual carrier sensing, tedy pozdržení vysílání v běžných *WiFi* sítích pomocí *network allocation vector* (*NAV*). Nevyužitý čas pro vlastní vysílání pak zpátky umožní využít běžným *WiFi* sítím. [28]

Některé z těchto metod, např. výběr vhodného kanálu a správné přenosové rychlosti, *ISP* již dnes běžně používají. Pro využití metody s mapou bezdrátových sítí je potřeba mnoho senzorů a je tak příliš nákladné.

V dalších kapitolách této práce představuji vlastní řešení problému, které funguje na principu detekce přetížení a eliminace pomocí řízeného shapingu, které nepotřebuje další nákladná zařízení. Toto řešení by mohlo doplnit metody, které se již používají.

3.2 Nové řešení pomocí řízeného shapingu provozu

Představa o tom jak řešit popisovaný problém lze ve zkratce popsat následujícími body:

- Detekovat zarušení nebo zahlcení bezdrátových vysílačů.
- Snížit rychlost uživatelů, kteří se na rušení nejvíce podílí.
- Následně detekovat zvýšení dostupné kapacity a postupně zvyšovat rychlost uživatelům.

Pro detekci zarušení nebo zahlcení bezdrátových vysílačů vychází toto navrhované řešení z předpokladu, že daná situace se projeví zvýšenou odezvou, případně až ztrátou dat. Tyto výchylky bude nutné detekovat, abychom na ně mohli reagovat, neboť budeme předpokládat, že každá taková výchylka znamená, že se uživatel v daném místě snaží využít větší množství kapacity, než je zde aktuálně dostupné. Odezvu budeme měřit jako dobu mezi vysláním *Internet Control Message Protocol* (*ICMP*) *Echo* a návratem *ICMP Echo Reply*, neboli budeme zjišťovat *Round-Trip Time* (*RTT*).

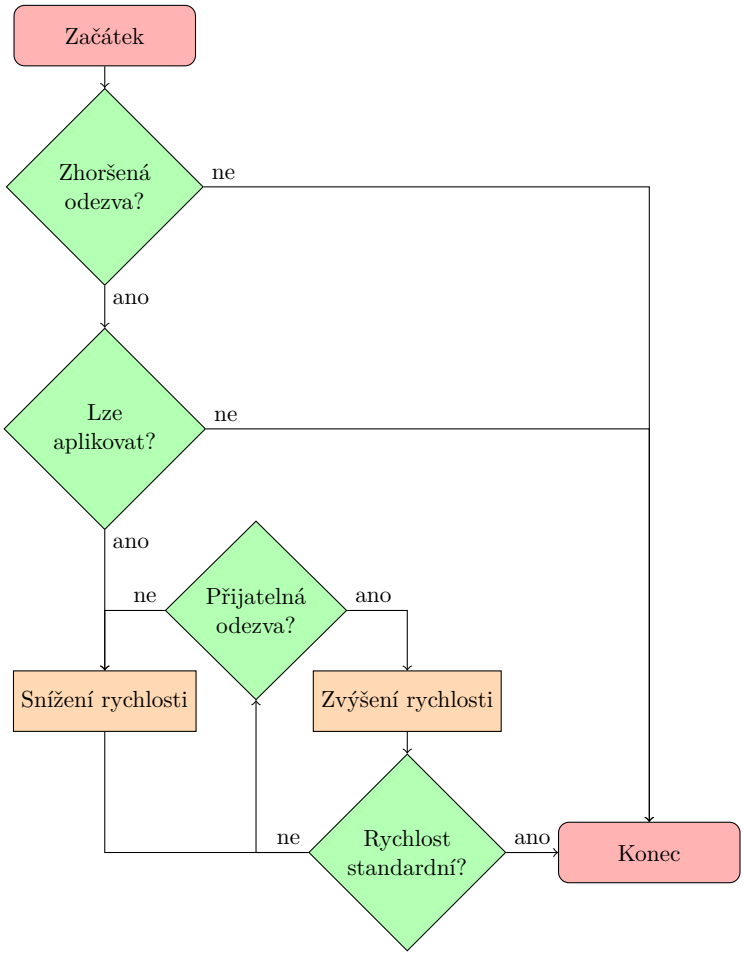
Detekované zvýšení odezvy z důvodu přetížení bezdrátového spoje je nutné odlišit od zvýšení odezvy z důvodu přepojení na záložní trasu. Zde si pomůžeme sledováním směrodatné odchylky *RTT*. Při přetížení spoje budeme očekávat kolísání *RTT*, naproti tomu přepojení na stabilní záložní trasu by se mělo projevit jen jednorázovou změnou *RTT* a dále nekolísat. Hodnota směrodatné odchylky v případě kolísání hodnot bude výrazně vyšší.

Získané údaje ze sledování *RTT* budou zaznamenávány pomocí *Round Robin Database Tool* (*RRDTool*), což je výkonný nástroj pro pravidelné zaznamenávání údajů v čase, jejich zpracování a vytváření grafů. Databáze *Round Robin* má stále stejnou velikost, neboť obsahuje stále stejné množství záznamů – při vložení nového záznamu se nejstarší odstraní. Z pravidelně zaznamenávaných dat lze počítat např. minima, maxima či průměry a tyto hodnoty ukládat pro delší časové období. [19]

Některá data se budou ukládat též do *MariaDB*, což je relační databáze, která vznikla jako svobodná alternativa k *MySQL*.

U uživatelů, kde bude detekováno největší přetížení, bude snižována jim přidělená rychlost tak, aby k přetížení nedocházelo. Jakmile se podaří dostat odezvu do běžných hodnot daného uživatele, budou se přidělené rychlosti zákazníků opět navyšovat. Takto pomocí snižování a navyšování přidělených rychlostí se bude hledat optimální nastavení pro danou chvíli tak, aby odezva zůstala v normálu, ale zároveň aby uživatelé dostali přidělenu co možná nejvyšší rychlost až do svého maxima. Graficky je toto znázorněno ve vývojovém diagramu na obrázku 2.

Tento řízený shaping musí být samozřejmě v mezích uživatelova tarifu, tedy pokud má uživatel např. zakoupen tarif s garantovanou rychlostí, nelze o řízeném shapingu vůbec uvažovat.



Obrázek 2: Vývojový diagram – shaping jednotlivých uživatelů. [autor]

Uživatelem zakoupený tarif má mimo jiné dánu maximální, běžně dostupnou a minimální rychlost, kterou *ISP* popisuje ve specifikaci služeb. Tyto rychlosti jsou v českém prostředí regulovány Českým telekomunikačním úřadem. Od 1.1.2021 je platné všeobecné oprávnění č. VO-S/1/08.2020-9 [10], kterým se stanoví podmínky k poskytování služeb elektronických komunikací. Pro služby přístupu k internetu v pevném místě platí, že běžně dostupná rychlost odpovídá alespoň 60 % hodnoty rychlosti inzerované a je dostupná v 95 % času během jednoho kalendářního dne. Minimální rychlost pak odpovídá alespoň 30 % hodnoty rychlosti inzerované.

V připravovaném řízeném shapingu se přidělená rychlost bude udržovat v mezích ohraničených zdola běžně dostupnou rychlostí, tedy alespoň 60 % hodnoty rychlosti inzerované, a shora maximální rychlosti, tedy 100 % hodnoty rychlosti inzerované. Pokud nebude

možné zajistit pomocí řízeného shapingu přijatelnou odezvu, nebude se rychlost dále snižovat a bude se předpokládat, že se jedná o poruchu, která potřebuje zásah technika.

Očekávaný přínos tohoto řešení je stabilní odezva pro uživatele a rychlá reakce na měnící se podmínky v zarušených bezdrátových sítích. Aplikace nemá za cíl zkoumat pouze přetížení bezdrátového spojení v úseku „poslední míle“, ale jejím účelem je reagovat i na přetížení kteréhokoliv spoje po cestě. Teoreticky by mohla být prospěšná i v případě licencovaných pásem, např. při silném letním dešti, nepřesně mířící anténě po vichřici, nebo podobné situaci. Takové použití ale nebylo testováno.

3.3 Implementace řešení

Popisované řešení je realizováno pomocí skriptu v *Pythonu* spouštěném pravidelně plánovačem úloh *Cron* na *Linux* serveru. Skript kontroluje odezvu jednotlivých uživatelů a pokud zjistí zhoršení odezvy oproti běžnému stavu, navrhne vhodnou úpravy rychlosti, kterou realizuje. Informace o provedených úpravách zapisuje do *MariaDB* databáze. U již evidovaných uživatelů kontroluje odezvu také a podle situace zvyšuje, či snižuje přiřazené rychlosti a to až do doby, kdy je rychlost uživatele nastavena na jeho maximální rychlost.

Informace o použití vytvořené aplikace je možné najít v souboru `README.md` v GIT repozitáři (odkazy viz. příloha B na straně 63) nebo ZIP archivu přiloženém k této práci.

Základní informace, které skript potřebuje k řízenému shapingu, jsou přidělené maximální a garantované rychlosti odesílání a přijímání dat. Jako garantovaná rychlost je volena vyšší hodnota z garantované rychlosti dle služby zákazníka a 60 % maximální rychlosti, tak aby skript zasahoval jen v hodnotách, které jsou dle nařízení VO-S/1/08.2020-9 [10] v pořádku.

Pro rozhodování potřebuje aplikace též statistiky odesílání a přijímání dat každého uživatele. Informace se získávají v případě IPv4 přímo z routerů, v případě IPv6 prostřednictvím *NetFlow* z routerů a tyto údaje se zapisují do *Round Robin* databáze. Tyto informace jsou pak v aplikaci využívány v případě zhoršení odezvy u uživatele k určení, jestli je smysluplné aplikovat u daného uživatele řízený shaping. Pokud by aplikování shapingu u daného uživatele nemělo přinést kýžený efekt, nerealizuje se a šetří se tak výkon.

K rozhodování je dále potřeba pravidelně získávat informace o odezvě uživatelů. K tomu účelu je použita aplikace *SmokePing*, která pomocí programu *fping* zjišťuje *RTT* mezi

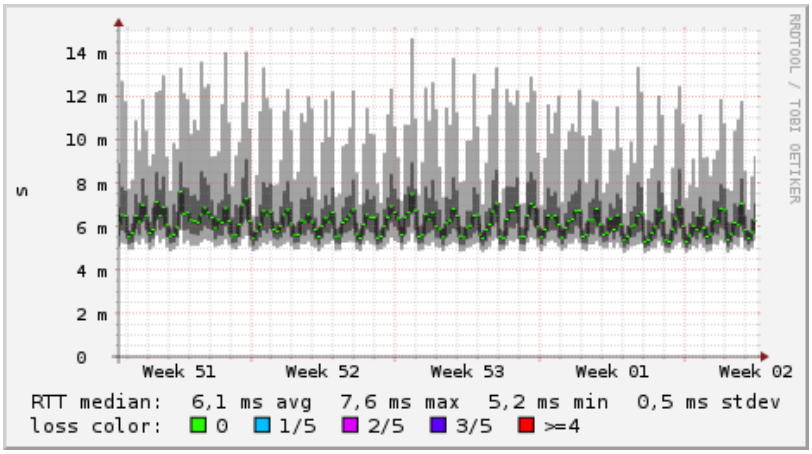
vysláním *ICMP Echo* a návratem *ICMP Echo Reply*. Získané údaje jsou zaznamenávány do *Round Robin* databáze. Tyto údaje získané pomocí *SmokePing* se průběžně zapisují i do *MariaDB* databáze, kam se průběžně též zapisují průměrné hodnoty *RTT* každého uživatele za 24 hodin a 7 dní zpětně. Vyvinutá aplikace pak může tyto hodnoty efektivně získávat a porovnávat.

Aplikace standardně porovnává aktuální hodnotu odezvy s denní průměrnou hodnotou. Pokud však denní průměrná hodnota není dostupná, nebo je nečekaně vysoká, použije se k porovnání týdenní průměr. Vychází se ze situace, že ideální je porovnávat odezvu s průměrnou denní hodnotou. V případě poruchy u zákazníka však může být denní statistika příliš vychýlená a v takových situacích je k dispozici relativně rozumná hodnota týdenního průměru. Díky tomu může aplikace hned po vyřešení problému u zákazníka pokračovat v činnosti a nemusí čekat celý den na naplnění denních statistik rozumnými hodnotami.

Pro účely aplikace, ale i pro přehled o stavu sítě, je sledována odezva všech uživatelů pravidelně každou minutu. Každému uživateli je odesíláno 6 požadavků *ICMP Echo* o velikosti 1300 bajtů s minimálním rozestupem 4 sekundy, z čehož odezva prvního z nich se ignoruje, kvůli možné rozkolísanosti. Získané údaje se zaznamenávají do *Round Robin* databáze, kde se takto udržuje 1440 časových polí záznamů, což odpovídá jednomu dni.

Jak bylo popsáno výše, dostupné jsou i informace o průměrných hodnotách za poslední týden, tedy je zřejmé, že i tyto informace musí být někde zaznamenány. To umožňuje též *RRDTool*. Hodnoty, vyčítané každou minutu, jsou ukládány do jednoho *Round Robin* archivu. Z několika těchto za sebou jdoucích hodnot jsou počítány minimum, průměr a maximum. Tyto hodnoty jsou ukládány do dalšího *Round Robin* archivu a tímto způsobem se tak získávají delší časové období záznamů s nižší přesností, což pro starší data stačí. Tento princip je možné opakovat vícekrát a tak získat např. statistiky týdenní, měsíční a roční. Pro ukázku, jak vypadá takto vygenerovaný měsíční graf *RTT* viz. obrázek 3. Roční data jsou pak vypočítané hodnoty z každého celého dne, což nám pro přehled za poslední rok zcela postačuje a datově zabírá naprosto minimální prostor oproti situaci, kdy by byly udržovány údaje za každou minutu po dobu jednoho roku.

Všechny tyto výpočty jsou konfigurovatelné a *RRDTool* umožňuje nepřeberné množství dalších operací s daty, např. v aplikaci používané výpočty mediánu nebo směrodatné odchylky se provádí přímo pomocí *RRDTool*. Slušelo by se podotknout, že hodnoty, se kterými



Obrázek 3: Ukázka měsíčního grafu *RTT* zaznamenaná pomocí *RRDTool*. [autor]

pracuje implementovaná aplikace, je medián z popisovaných pěti hodnot získaných pomocí požadavku *ICMP Echo*.

Odezva zákazníka je standardně ovlivněna mírou využití jeho přidělené maximální rychlosti, tedy pokud se využití blíží 100 % přidělené rychlosti, zvyšuje se i *RTT*. Takto naměřená hodnota *RTT* ale není ta správná potřebná pro detekci přetížení bezdrátového spoje, neboť cílem není zjišťovat, zda se zákazník blíží své maximální přidělené rychlosti, cílem je detekovat kvalitu spojení. Toto je řešeno pravidly ve firewallu na zařízeních, která realizují shaping, aby daná *ICMP* komunikace ze serverů, které detekci provádějí, nebyla ovlivněna shapingem. Jiná *ICMP* komunikace takto ovlivněna není a zákazník tak pomocí příkazu ping může sledovat odezvu ovlivněnou svým provozem, jak by asi očekával.

Pokud aktuální odezva zákazníka výrazněji překročí průměrnou hodnotu, je snížena rychlost zákazníka. Aby byla reakce co nejrychlejší a zároveň přiměřená, není snížení rychlosti o pevnou hodnotu, ale sníží se poměrově, dle zhoršení. Obdobně opačný stav, tedy zlepšení odezvy, má za následek zvýšení rychlosti poměrově, dle zlepšení.

Při praktickém testování se došlo k optimálním hraničním hodnotám 2,5 násobek pro snížení rychlosti a 1,2 násobek pro zvýšení rychlosti. Tím se myslí, že pokud se odezva zhorší o více než 2,5 násobek oproti průměrné hodnotě, dojde ke snížení rychlosti. Pokud se zlepší pod 1,2 násobek oproti průměrné hodnotě, dojde naopak ke zvýšení rychlosti. Pro hodnoty aktuální odezvy mezi 1,2 a 2,5 násobku oproti průměrné hodnotě zůstává shaping nezměněn. Tato oblast je cíleně relativně široká, neboť změny shapingu jsou časově náročné.

Pro samotný shaping se využívá skript v *Pythonu* na jiném serveru, kterému se předávají hodnoty k nastavení a on již sám zjišťuje, na kterých zařízeních je potřeba shaping upravit a co přesně musí nastavit.

K běhu skriptu pro eliminaci přetížení se v *MariaDB* databázi eviduje seznam zákaznických přípojek s upravenou rychlostí, a to především jejich aktuální nastavená rychlost odesílání a přijímání dat. Pro účely testování aplikace pro každou takto evidovanou zákaznickou přípojku také zaznamenává počátek řízení shapingu, poslední změnu shapingu a počet provedených změn. Tyto informace by bylo později možné použít i pro informování techniků o nutnosti zásahu, například pokud je nutné řídit shaping pro daného zákazníka delší než nastavenou dobu, nebo pokud je nastavená rychlost nižší než stanovená mez, případně v kombinaci s časem.

4 Detekce síťových útoků

Domácnosti a firmy k internetu připojují stále více zařízení a dávno už to nejsou jen osobní počítače, *Network Attached Storage* (*NAS*) zařízení a mobilní telefony. Přibyly také kamerové systémy nebo zařízení *Internet of Things* (z angličtiny „*internet věcí*“, *IoT*). Všechna tato zařízení jsou pod trvalým a stále sílícím tlakem v podobě neustále vznikajících nových virů, trojských koní a červů. Útočníci mají zájem na uživatele skrze tato zařízení zaútočit a na jejich úkor se obohatit, případně jejich zařízení využít pro svůj botnet.

Nebezpečnost virtuálního prostředí zároveň posiluje fakt, že uživatelé často spoléhají, že jejich bezpečnost je zajištěna někým jiným, nějakou autoritou. Ať už státem nebo nějakou soukromou organizací. [16, str. 407]

4.1 Kyberkriminalita

V současném internetu se již běžně setkáváme s kyberkriminalitou. Je považována za nový druh kriminality, ale část zařazená do této skupiny až tak nová není, je to běžná kriminalita, jen zasazená do digitálního prostředí, kde lze různá protiprávní jednání (podvody, krádeže, šikanu aj.) provádět často lépe, rychleji a efektivněji. [15, str. 181]

V posledních letech je kyberkriminalita na vzestupu a představuje celosvětový problém. Europol ve své zprávě [4] z roku 2014 uvádí, že kyberkriminalita stojí globální ekonomiky přibližně 300 miliard $ ročně. Od útoků prováděných jednotlivci pro zábavu či dokázání si technických schopnosti překonání překážek došlo k posunu k činnosti páchané především profesionály nebo organizovanými skupinami s cílem výdělku. [15, str. 183]

Ještě než se začneme dále věnovat zabezpečení sítě, zamysleme se ještě nad pohnutkami, které vedou útočníky ke kyberkriminalitě. Takto totiž bude lépe zřejmé s čím se potýkáme a lépe asi i odhadneme, jak se tento stav může vyvinout v budoucnu. Důvody vzestupu kyberkriminality jsou pochopitelné a spojené se třemi aspekty [15, str. 183]:

1. Závislost společnosti na internetu (resp. nabízených službách, technologiích aj.).
2. Kyberkriminalita se stala výnosným globálním byznysem.

3. Minimální gramotnost některých uživatelů využívajících informační a komunikační technologie.

Další pohled na situaci by mohla dokreslit tabulka 1, která nastiňuje, co pachatelé riskují a co mohou získat v případě phishing útoku v porovnání s fyzickým bankovním přepadením. Phishing sice není přesně ten typ útoku, který bychom z pohledu *ISP* dokázali přímo detekovat, ale je možné detekovat a případně zabránit útokům, které připravují pole pro pozdější phishing útok.

Tabulka 1: Statistiky FBI – porovnání běžného „bankovního přepadení“ s jednáním, které má povahu phishing útoku [15, str. 182]

Parametr	Průměrné ozbrojené přepadení	Průměrný kybernetický útok
Riziko	Pachatel riskuje, že bude zraněn či zabit.	Bez rizika fyzické újmy
Zisk	Průměrně 3–5 tisíc USD.	Průměrně 50–500 tisíc USD.
Pravděpodobnost dopadení	Dopadeno 50–60 % útočníků.	Dopadeno cca 10 % útočníků.
Pravděpodobnost odsouzení	Odsouzeno 95 % dopadených útočníků.	Z dopadených útočníků dojde k soudnímu projednávání pouze u 15 % útočníků a z nich je odsouzeno jen 50 %.
Trest	Průměrně 5–6 let, pokud pachatel při loupeži nikoho nezranil.	Průměrně 2–4 roky.

U kybernetického útoku se nepříjemně zvyšují možnosti, jaké množství osob lze útokem poškodit. Je těžko představitelné, jak jinak než prostřednictvím kybernetického útoku by bylo možné poškodit tisíce nebo miliony osob. Možnosti se zvyšují zároveň s tím, jak roste počet připojených počítačových systémů do kyberprostoru. Existují různá grafická znázornění probíhajících útoků[1]. [15, str. 182–183]

Varovná situace vzniká s připojováním *IoT* zařízení, jako jsou domácí spotřebiče nebo třeba automobily. Aktuálně jich je k síti připojeno několik desítek miliard a stále masivně přibývají. Otázka je, jak dobře jsou taková zařízení zabezpečena. Případně, pokud jsou dobře zabezpečena nyní, jak dobře obstojí v polovině nebo na konci své životnosti. Mnoho výrobců *IoT* zařízení má za cíl především co nejrychleji vyrobit takováto zařízení, aby je mohli uvézt na trh, a zabezpečení je pro ně podružné. [15, str. 184–185, 202]

Mnoho příkladů zneužití těchto zařízení známe již nyní, například v roce 2014 byla součástí botnetu lednice, jež rozeslala více než 750 000 e-mailů, které měly povahu spamu. Existují

[1] např. https://cybermap.kaspersky.com nebo https://map.lookingglasscyber.com

ale i mnohem nepříjemnější důsledky pro běžné uživatele, než je rozesílání spamu. Když už jsme u lednice, tak například útočníkem ovládaná teplota lednice, nebo nákup nesmyslného množství potravin, což lze pomocí „chytré“ lednice též provést, určitě není něco, po čem bychom toužili. Asi si ani nechceme představovat, co by útočníci mohli provádět s k síti připojeným kardiostimulátorem, autem či letadlem. [15, str. 184–185, 202]

S rozvojem různých služeb postavených na principu *as-a-service* jsou na různých undergroundových a darknet fórech nabízeny služby, které lze označit jako *Crime-as-a-service*. Ty nabízejí nepřeberné množství různých komodit a služeb, jaké lze v kyberprostoru využít nebo získat [15, str. 183–184]:

- *Research-as-a-service*[2]
- *Crimeware-as-a-service*[3]
- *Infrastructure-as-a-service*[4]
- *Hacking-as-a-service*[5]
- *Data-as-a-service*[6]
- *Spam-as-a-service*[7]
- *Ransomware-as-a-service*[8]

4.1.1 Síťová bezpečnost

Kvalitní šifrování nám zajistí utajení a důvěrnost dat, tedy ochranu před neautorizovaným únikem informací. Autentizace nám pak umožňuje ověřit totožnost protistrany, tedy ověřit,

[2] *Research-as-a-service* spočívá v průzkumu zranitelností cílového počítačového systému, či software. [15, str. 184]

[3] „Služba *crimeware-as-a-service* nabízí celou řadu aktivit od prostého prodeje malware, přes jeho „úpravu na míru“, dále pak dodávání exploitů (zranitelností) aj.“[15, str. 184]

[4] „*Infrastructure-as-a-service* pak představuje nabídku fyzických či virtuálních počítačových systémů (botnety, hostingové služby, pronájmy sítí aj.).“[15, str. 184]

[5] „*Hacking-as-a-service* v sobě může zahrnovat prosté prolomení přístupových údajů k e-mailu, účtu na sociální síti aj. až po profesionální a sofistikované útoky na vybranou oběť. Do této oblasti pak může spadat např. i provedení útoků typu *DoS* a *DDoS*.“[15, str. 184]

[6] Služba *data-as-a-service* nabízí nejžádanější komoditu, kterou jsou právě data. Jedná se např. o přístupové údaje k různým účtům, bankovním účtům, údaje z platebních karet, nebo informace o osobách (bydliště, data narození, telefonní čísla, e-maily aj.).[15, str. 184]

[7] Prostřednictvím *spam-as-a-service* je možné si objednat a zaplatit spamovou kampaň.[15, str. 184]

[8] *Ransomware-as-a-service* umožňuje objednat si napadení uživatelů pomocí ransomware za účelem jejich vydírání.[15, str. 221]

že druhá strana je skutečně tím, kým tvrdí, že je. Systém veřejných a privátních klíčů nám pak umožní ověřit integritu dat, tedy že přijatá zpráva je totožná se zprávou vyslanou. Bezpečnost sítě, to ale není jen šifrování a autentizace. [20, str. 487]

Existuje mnoho způsobů jak lze cílový systém kompromitovat, například skrze uhodnutí hesla, napadení virem nebo trojským koněm, díky softwarové chybě, prostřednictvím spustitelného programu nebo skriptu, nebo například vložením kódu s pomocí neošetřeného vstupu. [25, str. 662]

Významné jsou také zneužití nultého dne (*Zero Day Exploits*). Jedná se o zneužití chyb, na které ještě neexistuje oprava. To může být v době, kdy chyba softwaru ještě není veřejně známá, nebo také v situaci, kdy už oprava v podobě aktualizace daného softwaru vyšla, ale ještě není na daném systému aplikována. [25, str. 662]

Existují různé databáze zranitelností, hojně využívaná je *Common Vulnerabilities and Exposures* (*CVE*), kterou spravuje společnost MITRE Corporation pro divizi národní kybernetické bezpečnosti (National Cyber Security Division) ministerstva vnitřní bezpečnosti Spojených států amerických, lze ji nalézt na https://cve.mitre.org . Využívá se jako referenční zdroj pro popis zranitelností, jejich závažnosti a stupně rizika. [25, str. 664]

4.1.2 Malware

Internet se stal pro mnoho dnešních uživatelů a organizací nezbytným. Mnoho z nás jej využívá pro vyhledávání informací, ke komunikaci, propagaci, vzdálenému přístupu k různým zařízením, zábavě a nepřebernému množství dalších činností. Na internetu však nenajdeme pouze obsah, za kterým jsme přišli, ale číhá zde na nás také nebezpečí.

Naše zařízení může napadnout malware (z anglického malicious software, tedy škodlivý software), který může mít různé cíle. Například poškození našeho softwarového vybavení nebo znepřístupnění našich dat a následné vydírání. Může nainstalovat *spyware* (špehovací software) a pomocí něj sledovat naši aktivitu na internetu, odchytávat hesla, osobní údaje, sledovat nás pomocí webkamery nebo odposlouchávat pomocí mikrofonu. Naše zařízení může také zařadit mezi další tisíce podobně napadených zařízení, známých jako botnet. Botnety pak mohou sloužit například k rozesílání spamu nebo *Distributed Denial of Service* (*DDoS*). [17, str. 64]

Malware se v dnešní době sám replikuje. Jakmile je napaden nějaký hostitel, snaží se z tohoto hostitele napadnout další. Může se šířit buď jako vir, nebo červ. Rozdíl spočívá v tom, že vir k nákaze daného hostitele potřebuje interakci uživatele, červ se dokáže šířit bez přispění uživatele. Příkladem šíření viru je napadení prostřednictvím otevření nebezpečného e-mailu uživatelem. Příkladem šíření červa je napadení skrze provozování zranitelné síťové aplikace. [17, str. 64]

4.1.3 Botnet

O botnetu jsme se již v předchozích kapitolách zmínili. Botnet je síť botů (zkrácenina slova robot), která vykonává činnosti dle zadání botmastera[9]. Bot je pak na dálku ovládaný počítačový systém. [15, str. 193–204]

Počáteční myšlenka pro vznik botnetu byla využít (pro legální účely) výkon počítačů v době, kdy nevykonávají žádnou práci a použít je na distribuované výpočty. Při takovém rozdělení práce je možné získat výkon převyšující výkon superpočítače. To samozřejmě nezůstalo bez povšimnutí a využití k nelegálním činnostem přišlo také. [15, str. 193–204]

Některé známé botnety sdružují stovky, jiné klidně miliony botů. Pro svoji činnost mohou využívat servery nebo běžné osobní počítače. Některé botnety se specializují například na chytré telefony, které dnes disponují dostatečným výkonem a díky jejich nižšímu zabezpečení oproti běžným počítačům bývá jednodušší nainstalovat do těchto zařízení malware a získat tak nad nimi kontrolu. Pro botnet jsou využitelné i například zařízení IoT. [15, str. 193–204]

Důvod k používání botnetu je při síťových útocích především rozdělení útoků mezi mnoho botů. Pokud by se útočilo jen z jednoho místa, byť by to bylo ze superpočítače s velmi rychlým připojením k internetu, bylo by velmi jednoduché útok detekovat a zablokovat. Pokud se však útočí z mnoha míst a každý bot se na útoku podílí jen malým množstvím operací, jsou jednotlivě těžko detekovatelní a mají tak větší šanci ve své nelegální činnosti uspět.

Smyslem botnetu ale nemusí být jen síla daná množstvím botů. Pro botmastera je výhodné mít členy svého botnetu rozmístěny v různých sítích ve světě. Pokud bude potřebovat

[9] Botmaster je vlastník nebo správce botnetu.

napadnout konkrétní systém, může pak využít i botů v nějaké části sítě, ze které by mohlo být jednodušší cílový systém napadnout. Může také proskenovat pro něj zajímavé síťové služby ze všech svých botů a odhalit tak například, že z nějaké sítě služba dostupná je, přestože z jiných je blokována. Například z důvodů geografické blokace, kdy cíl útoku nechce svoje služby poskytovat do jiné země světa. Nebo pokud je vstup na dané služby otevřen třeba jen pro pobočky firmy, která je cílem útoku.

Útočníkům v tomto smyslu někdy nechtěně pomáhají i samotní správci systémů, když si zjednodušují práci při vytváření pravidel firewallu a používají univerzálně masky /8 (255.0.0.0), /16 (255.255.0.0) nebo /24 (255.255.255.0) i v případech kdy je ve skutečnosti maska jiná. Práce s těmito maskami je sice příjemná a přehledná, neboť určují adresu sítě po celých oktetech, ale správci tak otevírají serverové služby mnohem širšímu okruhu sítí, mnohdy i z jiné země. Útočníci o těchto praktikách, kterými si nedůslední správci sítí usnadňují práci, vědí a rádi je použijí ke svému užitku. Jelikož mají výběr z mnoha strojů v internetu, které jsou součástí jejich botnetu, budou se snažit prolomit cílovou síť i ze sítí adresně blízko této cílové a mají tak větší šanci na úspěch.

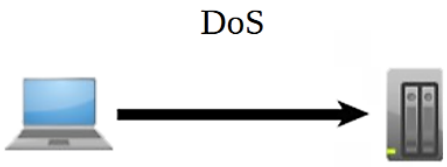
4.1.4 DoS a DDoS

Jak zkratka *Denial of Service* napovídá, cílem *Denial of Service* (*DoS*) útoku je znepřístupnění služby. Pro uživatele se server napadený *DoS* útokem projevuje vysokou odezvou, občasnými výpadky, nebo úplnou nedostupností služby. [15, str. 296]

Většina *DoS* útoků spadá do jedné z těchto kategorií [17, str. 65]:

- *Útok na chybu zabezpečení.* Odstavení síťové služby nebo operačního systému cílového hostitele pomocí vhodně zvolené komunikace. To může mít za následek zastavení služby nebo celého hostitele.
- *Zahlcení šířky pásma.* Využití celé šíře pásma cílového hostitele. To může mít za následek až úplné zabránění oprávněným paketům dostat se k cíli nebo doručení odpovědi zpět.
- *Zahlcení připojení.* Zahlcení mnoha polootevřenými nebo plně otevřenými TCP připojeními. Hostitel tak není schopen přijímat legitimní připojení.

V případě *DoS* útoku je zdroj útoku jen jeden (viz. obrázek 4). Pro *DoS* se přímo nabízí situace, kde je pro útočníka jednodušší vyslat požadavek, než pro oběť jej zpracovat. Útoky také využívají záměrných změn v hlavičkách paketů, a to zdrojových adres a portů pro obcházení nastavených filtrů, nebo dalších hodnot v hlavičce IP datagramu, i rezervovaných nebo nepoužívaných. [20, str. 493]



Obrázek 4: DoS útok. [15, str. 296]

SYN flooding je typickým případem zahlcení připojení, kdy útočník zasílá oběti mnoho *SYN* požadavků o spojení. Oběť pak odpovídá zprávou *SYN-ACK*, na něž již nedostává odpověď, a tedy žádné potvrzení *ACK* neobdrží. Přibývají polootevřená spojení, až oběť začne odmítat další nová spojení, tedy i oprávněné žádosti o navázání spojení. [20, str. 493–494]

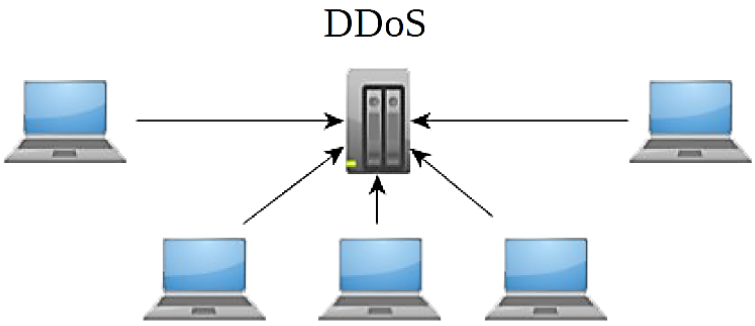
Ve většině hlavních operačních systémů je dnes používána účinná obrana proti *SYN floodingu*, známá jako *SYN cookies* (RFC 4987). Bez této obrany server po přijetí *SYN* přiděluje a inicializuje danému připojení proměnné a buffery, následně klientovi odpovídá zprávou *SYN-ACK*, na kterou klient odpovídá zprávou *ACK*, čímž je dokončen *třícestný handshake* a spojení je úspěšně navázáno. Pokud klient neodešle zprávu *ACK*, server zbytečně udržuje přidělené prostředky pro daného klienta a *SYN flooding* pak nakonec všechny prostředky serveru vyčerpá. V případě použití *SYN cookies* server nepřiděluje prostředky ihned po přijetí *SYN*, ale až po dokončení *třícestného handshaku*. Zpola otevřená spojení tak nezahltí systém a *SYN flooding* tím ztrácí pro útočníky na významu a ti se proto zaměřují především na zahlcení sítí a spojů, jimiž jsou oběti připojeny. [17, str. 212] [20, str. 496]

Pro zahlcení šířky pásma bývá někdy použit *ping flood*, útočník vysílá na cíl velké množství *ICMP Echo* paketů a oběť odpovídá pomocí *ICMP Echo Reply*.

DoS útoku bez využití spoofingu (viz dále) je relativně jednoduché se ubránit, neboť stačí zablokovat jen tento jeden zdroj útoku. Při útoku na zahlcení šířky pásma ani nemusí být jeden zdroj schopen zahltit celou šíři pásma jednoho hostitele, především pokud má hostitel

šíři pásma velmi vysokou. Navíc útok tohoto typu může být detekován a zablokován již někde po cestě, kde je šíře pásma ještě relativně vysoká.

V případě *DDoS* útoku (viz. obrázek 5) pak může útočník využít mnoho zdrojů, obrana proti nim je tak mnohem obtížnější. Pro zahlcení je pak nutné vygenerovat v součtu vyšší provoz než je šířka pásma hostitele. [17, str. 65] [15, str. 296]



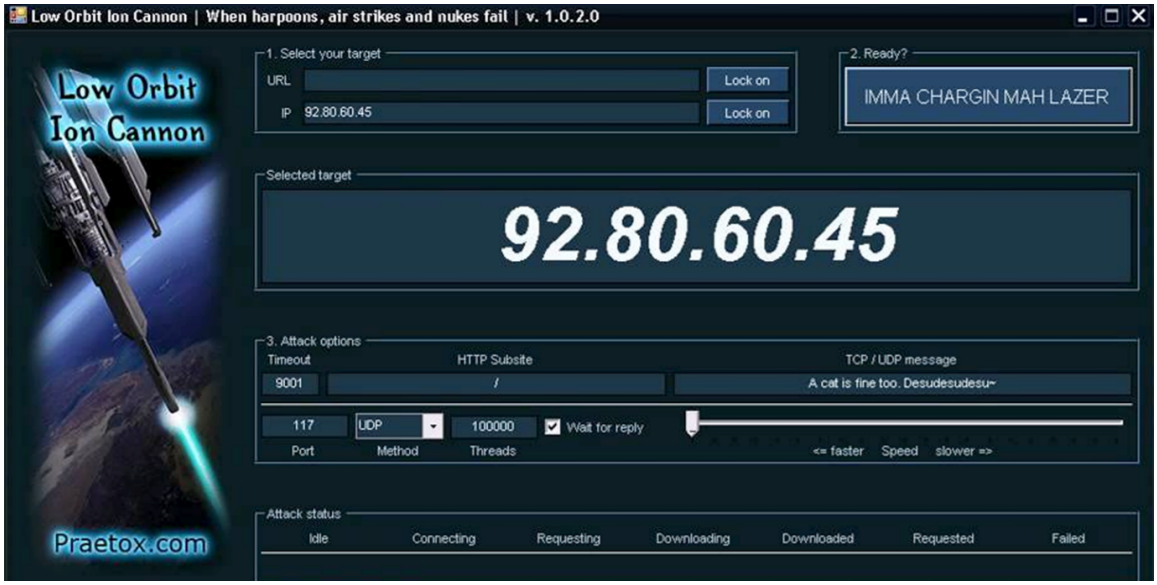
Obrázek 5: DDoS útok. [15, str. 296]

K provedení *DDoS* útoku nemusí být nutné použít jen botnet, ale existují i situace, kdy se mnoho skutečných uživatelů ve stejnou dobu připojuje ke stejné službě. Může to být například z důvodu vyhlášení slevové akce na nějakém webu. Ať už skutečné, vyhlášené provozovatelem webu, nebo domnělé, inzerované velkému okruhu uživatelů útočníky. Existují též akce, kdy se útočníci svolávají na konkrétní čas a společně se pak například opětovně přihlašuji a odhlašují, aby cílený systém přetížili. Ke stejnému účelu využili přívrženci hnutí Anonymous open-source aplikaci LOIC (Low Orbit Ion Cannon), standardně určenou pro zátěžové testování sítě (viz. obrázek 6). [15, str. 296–300]

4.1.5 Spoofing

DoS a *DDoS* popsané výše útočníci kombinují s falšováním zdrojové adresy, neboli *spoofingem*. V takovém případě samozřejmě útočník nemůže dostat odpověď na svůj dotaz, ale to v případě tohoto útoku ani nepotřebuje.

V případě *SYN flooding* tak útočník vysílá SYN požadavky se zfalšovanou zdrojovou adresou a oběť odpovídá zprávou SYN-ACK na tyto falešné adresy, až vyčerpá svoje prostředky



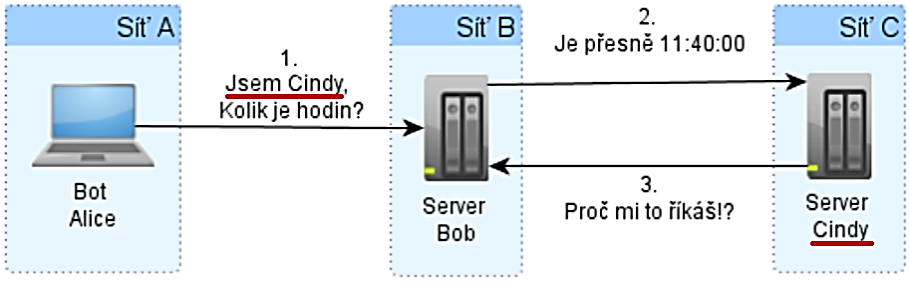
Obrázek 6: Aplikace LOIC (Low Orbit Ion Cannon) distribuovaná přívrženci hnutí Anonymous jako prostředek pro *DDoS* útok. [15, str. 299]

a začne odmítat i oprávněné žádosti o navázání spojení. To se děje v případě, že nemá ochranu proti SYN floodingu.

Útočník se prostřednictvím *spoofingu* může schovávat pod mnoha různými zdrojovými adresami, klidně náhodně generovanými, a oběť se tak nemůže jednoduše bránit tím, že by odmítala komunikaci z konkrétní IP adresy. Jsou ale i situace, kdy útočník používá jen jednu zdrojovou IP adresu, případně několik cíleně vybraných.

Ať už ve variantě *DoS* nebo *DDoS* se k vedení útoku pro zahlcení šířky pásma zneužívají takzvané reflektory (viz. obrázek 7), tedy zařízení na která útočníci odesílají zprávy, kde zfalšovaná zdrojová adresa je cíl útoku. Takové útoky se někdy označují jako *Distributed Reflected Denial of Service* (*DRDoS*) [15, str. 297]. Reflektorů se využívá především, pokud je zpráva s odpovědí mnohem větší než zpráva s dotazem. V takovém případě útočníkovi stačí posílat mnohem menší množství dat, než jakým poté zahltí reflektor oběť. I samotný reflektor pak může být částečně též obětí, ale samozřejmě ovlivněn mnohem menším datovým provozem, než cílová oběť na kterou je směrován útok prostřednictvím mnoha reflektorů. [20, str. 491, 495]

Tomuto útoku prostřednictvím reflektoru se říká též zesilovací (amplificated) útok, právě kvůli danému pákovému efektu, kdy se prostřednictvím reflektoru útok od útočníka směrem



Obrázek 7: Využití reflektoru při *DoS* útoku – reflektor je zde Server Bob. [15, str. 297]

na cíl násobí. Tento útok se týká především služeb na protokolu UDP, z těch nejznámějších například *Network Time Protocol* (*NTP*) nebo *Domain Name System* (*DNS*). Pákové efekty mohou být velmi výrazné, existují i služby které mají poměr vyšší než 100 násobek [22]. To prakticky znamená, že útočník prostřednictvím datového toku 100 Mbit na reflektor může docílit datového toku více než 10 Gbit z reflektoru na oběť.

Útočníci se také mohou snažit při spoofingu využít IP adresu, která se pro cíl bude jevit důvěryhodně. Tedy najít například IP rozsah podnikové sítě nebo jiné pobočky, která by přístup mohla mít povolen a tedy by služba nemusela být patřičně chráněna. Při zfalšování zdrojové IP adresy útočník samozřejmě nemůže získat odpověď na svůj požadavek. To ale útočníkovi nemusí vadit. [20, str. 491]

Spoofingu by bylo možné velmi výrazně zamezit, pokud by každý *ISP* alespoň neumožnil odcházet z jeho sítě paketům se zdrojovou unicastovou IP adresou, která nepatří do jeho rozsahu.

4.1.6 Man in the middle

Každý *ISP* mimo to, že chrání svoje zákazníky, musí chránit též svoje síťová zařízení. Jakékoliv napadené síťové zařízení bývá výraznější problém pro bezpečnost sítě *ISP* než napadené zákaznické zařízení. Například přes routery nebo switche prochází komunikace mnoha zákazníků a je tak pro útočníky zneužitelné nejen pro předchozí popsané útoky, ale také pro útok *Man in the Middle* (z angličtiny „*člověk uprostřed*“, *MITM*).

Napadená síťová zařízení nemusí být využita jen ke sledování nebo úpravě komunikace přes ně procházející, ale také k úplnému jeho odstavení a tím znefunkčnění připojení k inter-

netu pro všechny za ním připojené, pokud tedy není dobře vyřešeno zálohování, případně i multihoming. Prostřednictvím napadení routerů též může dojít k včlenění nepravdivých informací do routovacích tabulek a tím opět buď znefunkčnění spojení mezi některými dvěma stranami, případně nasměrování na zákeřné stroje, které komunikaci převezmou. [20, str. 498]

Proniknout na síťová zařízení lze mimo jiné odhalením hesla. To lze například hrubou silou opakovanými pokusy o uhodnutí hesla, nebo odposlechnutím při použití nešifrovaných protokolů jako je *Telnet*. Další možností je prostřednictvím napadení stroje administrátora například pomocí trojského koně. [20, str. 492]

Mohlo by se zdát, že jediný problém zabezpečení je použít silné heslo a to nevyzradit. Což je sice důležité, ale není to jediná cesta jak získat přístup do zařízení. Tou může být například neopravená bezpečnostní chyba na zařízení, nebo důsledek podcenění fyzické bezpečnosti.

4.1.7 Další útoky

Krátce se ještě zmíníme o zabezpečení *Local Area Network* (z angličtiny „*lokální síť*“, *LAN*), neboť to musí *ISP* řešit samozřejmě také. Z principu věci bude nutné ochránit především správnou funkci *Address Resolution Protocol* (*ARP*) a *Dynamic Host Configuration Protocol* (*DHCP*). Útočník může spustit *Rogue DHCP*, odpovídat tak na *DHCP* dotazy ostatních uživatelů a tím jim buď znefunkčnit internetové připojení nebo je připojit přes svoje zařízení, čímž může realizovat útok *MITM*. Obrana proti tomuto útoku je možná pomocí konfigurace *DHCP snooping* na switchi, která umožní provozovat *DHCP* server jen na konkrétních portech switche dle konfigurace. Znefunkčnit připojení k internetu lze též pomocí *DHCP Starvation*, kdy útočník vysílá velké množství *DHCP* požadavků s podvrženými MAC adresami, čímž vyčerpá přiřazovaný IP rozsah, bránit se lze pomocí konfigurace *Port Security* na switchi. Útok *MITM* lze realizovat též pomocí *ARP Cache Poisoning*, obrana proti tomuto útoku už rozhodně není tak jednoduchá jako v případě *Rogue DHCP*. [16, str. 429–438]

Zdaleka ne všechny útoky byly výše popsány, ale byl to výčet především útoků, které nás zajímají z pohledu ochrany zákazníků *ISP*, tedy především útoky na zákazníky, které je schopen *ISP* alespoň částečně podchytit. Více nepopisuji ani například útoky na servery

DNS, jako otrávení záznamů v cache, kterým se *ISP* musí samozřejmě také zabývat, ale týkají se komunikace klient – server, které se v této práci hlouběji nevěnuji. Více se v této práci nevěnuji ani fyzické bezpečnosti nebo sociálnímu inženýrství.

Jak vidno z předchozího výčtu, který asi nikdy nemůže být úplný, neboť nové typy útoků stále vznikají, útoků je nepřeberná řada. Některé více, jiné méně, ohrožují nebo znepříjemňují práci nejen uživatelům, ale i *ISP*. Než řešit pouze následky, případně čekat na ukončení útoku, je pro *ISP* chytřejší útoky blokovat, situaci řešit a mít připraveny metody jak následky minimalizovat. Zároveň chráněná síť nebývá pro útočníky tolik lákavá, naopak „útočníci prohledávají častěji bloky adres, kde očekávají více slabě zabezpečených systémů“ [20, str. 496].

Na druhou stranu, je třeba zaměřit se především na nákladově efektivní ochranu, vytvářet 100 % dokonalé zabezpečení není obecně ani možné [20, str. 490].

4.1.8 Registr WHOIS

Útoky byly popsány v předchozích kapitolách a jejich detekcí se budeme zabývat dále. Bylo by ale také vhodné popsat, jakým způsobem probíhá ohlášení útoku, když je detekován, nebo napadenému způsobuje problémy. Zde nám nejde vyloženě o to, nahlašovat útok na policii, ale spíše, co nejrychleji se zbavit útoku přímo u zdroje, tím, že nahlásíme jinému *ISP*, že má v síti útočící zařízení.

Co tedy může napadený dělat, když jediné, co by mohlo být vodítkem k útočníkovi je zdrojová IP adresa útoku? A kde zjistit, komu IP adresa patří a komu daný útok nahlásit? Nejen k tomu slouží registr *WHOIS*, kde lze dohledat registrátora dané IP adresy a e-mailový *abuse* kontakt pro tuto IP adresu. Na tento *abuse* kontakt je pak možné zaslat stížnost na daný útok. Příklad takových stížností je v příloze C na straně 64. Posílat stížnost má samozřejmě smysl jen v případě, že se nejedná o spoofing.

Jak je to s *WHOIS* databází a registrátory? Hlavní správu nad IP adresním rozsahem má *Internet Assigned Numbers Authority* (*IANA*), která adresní rozsahy rozděluje organizacím označovaným jako *Regional Internet Registry* (*RIR*), jednotliví regionální registrátoři jsou [25, str. 472–473] [15, str. 136–138] [5]:

1. AFRINIC – „Africká“ oblast – vznik 2005
2. APNIC – „Asijsko pacifická“ oblast – vznik 1993
3. ARIN – „Severo-americká“ oblast – vznik 1997
4. LACNIC – „Jiho-americká“ oblast – vznik 2001
5. RIPE NCC – „Euro-asijská“ oblast – vznik 1992

RIR v těchto pěti teritoriích (viz mapa na obrázku 8) přidělují síťový rozsah dále jednotlivým *Local Internet Registry* (*LIR*) a mimo jiné provozují službu *WHOIS*, což je označení pro databázi, v níž jsou evidovány údaje o držitelích IP adres. *LIR* bývá zpravidla *ISP* (v českém právu se označuje jako poskytovatel služeb informační společnosti), provozovatel datového centra, nebo větší organizace. *LIR* může svůj rozsah dále poskytnout jiným subjektům nebo využít sám. [15, str. 136–138]

Obrázek 8: Regionální internetoví registrátoři. [5]

Každý *RIR* eviduje informace o přiřazených síťových rozsazích a každý *LIR* je pak povinen evidovat informace o jím přiřazených IP rozsazích. U větších bloků adres bývá uveden přímo uživatel daného rozsahu, tady například firma nebo organizace. U menších bloků adres je uveden jen *LIR*, ale ten je samozřejmě schopen dohledat konkrétního uživatele dané IP adresy. V každém případě by měl být uveden *abuse* kontakt, kde je možné nahlásit kybernetický útok.

Nahlašovat detekované útoky na *abuse* kontakt lze i automaticky, například prostřednictvím software *Fail2ban* pro *POSIX* systémy, který chrání například před záplavovými útoky nebo útoky hrubou silou.

4.2 Analýza problému

Mohlo by se zdát, že z hlediska *ISP* bude ochrana uživatelů mimo jeho zájem. Vždyť proč by měl *ISP* zasahovat do komunikace svých zákazníků a nějakou komunikaci blokovat? Každý zákazník by měl nejlépe sám vědět, kterou komunikaci chce povolit a která je z jeho pohledu závadná a takovou blokovat.

To by platilo v situaci, kdy uživatelé jsou odborníci, kteří se v problému perfektně orientují. To však ve velkém měřítku není pravda. Zákazníci *ISP* jsou převážně jen uživatelé, kteří chtějí používat internet. Stejně tak zařízení uvnitř své vnitřní sítě chtějí mít veřejně dostupné proto, že se na něj chtějí vzdáleně připojovat (ať už je to kamerový server, *NAS*, chytrá domácnost atd.), nebo potřebují, aby se na jejich zařízení mohla připojit třetí strana (vyčítání statistik z kotlů, klimatizací apod.). Takoví uživatelé často neumí nastavit firewall a případně ani netuší, že otevřením přístupu z celého světa na všechny služby ve vnitřní síti se vystavují nebezpečí proniknutí útočníků do jejich domácnosti.

V zájmu *ISP* samozřejmě je, aby uživatelé v jeho síti byli chráněni, neměli napadené zařízení a neměli do internetu otevřené služby, jejichž prostřednictvím by mohli jako reflektor poskytovat možnost útočníkům směrovat útoky dále. Mohlo by to ale skončit prostým konstatováním, kdy by pro to *ISP* dále nic nedělal. Proč tedy má smysl investovat čas a prostředky do zabezpečení a blokaci útoků, ať už z internetu do vnitřní sítě nebo obráceným směrem?

V zájmu *ISP* je primárně dodávat kvalitní službu svým uživatelům. Je celkem logické, že pro to potřebuje mít ochráněna svoje zařízení, která jsou pro dodávání služby zásadní. Potřebuje ale též ochránit své uživatele především proti takovým útokům, které by ovlivňovaly dodávané služby.

V zájmu *ISP* také je, aby nebyla napadena zařízení uživatele. Jedním z důsledků napadení by mohlo být, že se zařízení uživatele bude chovat nestandardně, což může vézt ke zhoršení kvality dodávané služby. Horší je fakt, že napadené zařízení může útočit dále do vnitřní sítě,

ať už na zařízení *ISP*, další zákazníky, nebo do internetu. Útočící IP adresa pak může být postiženými protistranami v internetu zablokována, nebo se dostane na černé listiny, a opět uživatel je tím výsledně omezen. V situaci sdílení veřejných IPv4 adres pomocí *Network Address Translation* (z angličtiny „*překlad síťových adres*“, *NAT*) takto napadený zákazník neovlivňuje jen sám sebe, ale všechny zákazníky se kterými danou IPv4 adresu sdílí. Když už k situaci dojde, vyjednávání pro odstranění blokace bývají zdlouhavá, pracná, někdy nemusí být ani možná, a tak je lepší takovým situacím předcházet.

ISP zároveň nechce zhoršovat reputaci svých IP adres nebo celého autonomního systému, opět by to mělo vliv na jeho zákazníky. V neposlední řadě je dobré nebýt v hledáčku útočníků, kterým by síť daného *ISP* mohla přijít jako zajímavý cíl nebo prostředník pro jejich činnost.

Když se problém z pozice *ISP* rozumně uchopí, může využít znalosti svých specialistů. Navíc může nahlížet na informace v celé síti a odhalit tak problémy, které pro jednotlivé zákazníky již nemusí být zjistitelné, byť by se v problému perfektně orientovali. K blokaci je ale nutné přistupovat s pokorou, nevěřit v neomylnost a dát jednotlivým zákazníkům také možnost, aby těmito pravidly nebyli omezeni, pokud jim nevyhovují. V takovém případě je však logické, že se takový zákazník musí postarat o zabezpečení sám, tak aby neohrožoval ostatní.

Z pohledu *ISP* je tedy vhodné připravit potenciálním útočníkům do cesty další překážky v podobě zesílené ochranu sítě a zvýšení laťky, kterou musí útočníci překonat. A tak když už se nepodaří jejich útokům úplně zamezit, alespoň jim jejich útok znesnadnit. Docílit by tím bylo možné alespoň toho, že se přestanou zajímat o danou síť a obrátí se směrem ke snazším kořistem. [25, str. 662]

Pro zabezpečení zařízení na síti je potřeba co nejvíce snížit útočný povrch zařízení. Tedy uzavřít co nejvíce cest k zařízení a minimalizovat tak na co možná nejnižší míru riziko napadení. To například znamená nenechat do světa zbytečně otevřeny služby nebo porty, které se nevyužívají. Pokud by však i přes podobná opatření bylo zařízení napadeno, minimalizace dále nemá žádný účinek, tedy například nijak nesnižuje důsledky napadení. [25, str. 668]

4.2.1 Intrusion Detection System

Jeden z možných způsobů, jak detekovat útoky, je pomocí *Intrusion Detection System* (z angličtiny „*systém detekce průniku*“, *IDS*). Filtrovat provoz pak pomocí *Intrusion Prevention System* (z angličtiny „*systém prevence průniku*“, *IPS*). Tento nástroj kontroluje nejen hlavičky paketů, ale provádí i jejich hloubkovou inspekci, tedy sleduje i aplikační data. V případě, že nalezne podezřelý paket nebo sérii paketů, může takovou aktivitu označit za podezřelou a odeslat správci sítě upozornění. Případně rovnou zakročit a zabránit takovým paketům ve vstupu do vnitřní sítě. *IDS* sleduje podezřelé chování v síťovém provozu a lze jej tak využít k detekci velkého množství útoků, mimo jiné skenování portů, chybnou identifikaci uživatelů, zaplavení šířky pásma útoky *DoS*, útokům na zranitelnost operačního systému nebo na zranitelnost aplikací. [17, str. 567–569] [20, str. 489]

Systémy *IDS* se rozlišují na dvě skupiny: systémy na bázi signatur a systémy na bázi anomálií. Systémy na bázi signatur porovnávají pakety se signaturami ve své databázi. Mohou tak porovnat pakety nebo sérii paketů s již známými útoky. Signaturami se má na mysli informace o paketu, např. zdrojovém a cílovém portu, typu protokolu, nebo specifické posloupnosti bitů v obsahu paketu. Může popisovat i sérii paketů. [17, str. 567–569] [20, str. 489]

Systémy na bázi anomálií vytváří na základě pozorování běžné komunikace komunikační profil a poté hledají toky, které jsou statisticky neobvyklé. Nespoléhají se tak na informace o dříve známých útocích a mohou zachytit i nové dosud neznámé útoky. Rozlišovat mezi běžnou a statisticky neobvyklou komunikací je však náročný úkol a tak není divu, že systémy na bázi signatur převažují. [17, str. 567–569] [20, str. 489]

4.2.2 Detekce anomálií

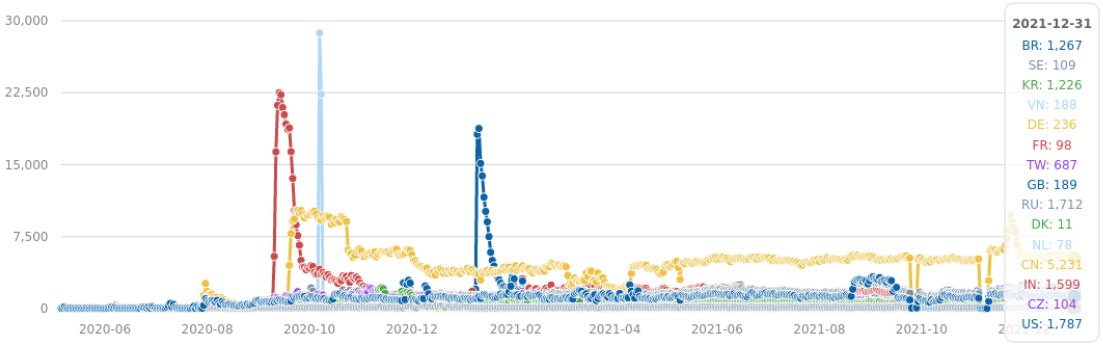
Indičtí autoři se ve své práci [24] zabývají útoky *DDoS*, které považují za jednu z největších hrozeb pro internet. Vidí potřebu obranných mechanismů, které by dokázaly síť ochránit. Pro tento účel navrhují vytvořit distribuovaný systém ochrany, kde by každý *ISP* detekoval útoky na svých routerech, tyto informace by zpracovával a sdílel s ostatními *ISP*. Díky této spolupráci by pak společně mohli blokovat útoky velmi efektivně.

Pro detekci anomálií v síťovém provozu sledují entropii pro měření náhodnosti různých dat z probíhajících toků jako je zdrojová nebo cílová IP adresa, port, celkový počet paketů apod. V běžném stavu očekávají entropii relativně stabilní, v případě *DDoS* útoku pak může některá hodnota dominovat, což vede ke snížení entropie. Pokud taková situace nastane, snaží se vyhodnotit, zda se jedná o *DDoS*. Pokud ano, je útok zablokován a informace o útoku je sdílena s dalšími sítěmi, aby mohli útok též blokovat. [24]

4.2.3 Turris Sentinel

Velmi zajímavý počin s ohledem na kybernetickou bezpečnost jsou routery Turris od sdružení CZ.NIC. Co pro nás bude především zajímavé, je systém pro detekci a ochranu před kybernetickými útoky *Turris Sentinel*. Tento systém detekuje na všech zapojených routerech Turris kybernetické útoky, informace o útocích odesílá na centrálu, kde se hodnotí jejich nebezpečnost a jakmile překročí stanovenou hranici, je daná útočící IP adresa přidána na seznam (*Sentinel greylist*) k blokaci. Zařízení si tyto seznamy IP adres automaticky stahují a blokují jejich přístup do vnitřní sítě v reálném čase pomocí dynamického firewallu *DynFW*. [9]

Co všechno dokáže detekce hrozeb systému *Turris Sentinel* sledovat? Sleduje záznamy z firewallu, informace z *minipot* a také z *SSH Honeypot*. Získané informace se využívají pro různé bezpečnostní analýzy Turris týmem a národním CSIRT České republiky CSIRT.CZ. Na grafu na obrázku 9 je zanesen vývoj útočníků dle země získaný pomocí údajů z *Turris Sentinel*. [9]



Obrázek 9: Graf útočníků dle země získaný pomocí údajů z *Turris Sentinel* [8].

Ze záznamů z firewallu je sledováno, která IP adresa se snaží připojit na jaký port. Ke každé podezřelé IP adrese je zaznamenáno hodnocení v závislosti na typu útoku. [9]

Turris minipot (zkrácenina ze slov minimal honeypot) je odlehčený *honeypot*, běžící na službách *Telnet*, *Hypertext Transfer Protocol* (*HTTP*), *File Transfer Protocol* (*FTP*) a *Simple Mail Transfer Protocol* (*SMTP*). Minipot emuluje chování daného protokolu a umí tedy jen minimum toho, co tyto protokoly běžně umí, tak aby to stačilo pro útočníkův dojem, že komunikuje se skutečnou službou. Na každou snahu o přihlášení na těchto protokolech navrací minipot „incorrect password“, zadané kombinace přihlašovacích jmen a hesel spolu s IP adresou pak zaznamenává. [9]

Pro běh *Secure Shell* (*SSH*) *Honeypot* se využívá projekt Cowrie[10]. Ten neběží přímo na routerech Turris, ale na serveru u CZ.NIC, na který se pomocí proxy směruje provoz z routerů Turris s aktivovanou službou *Honeypot as a Service* (*HaaS*). *SSH Honeypot* simuluje operační systém a umožní útočníkovi přihlásit se pomocí *Telnet* nebo *SSH*, spouštět příkazy a stahovat malware. Zaznamenávají se přihlašovací údaje a chování útočníka. Použité příkazy jsou analyzovány a stejně tak stažený malware. [9]

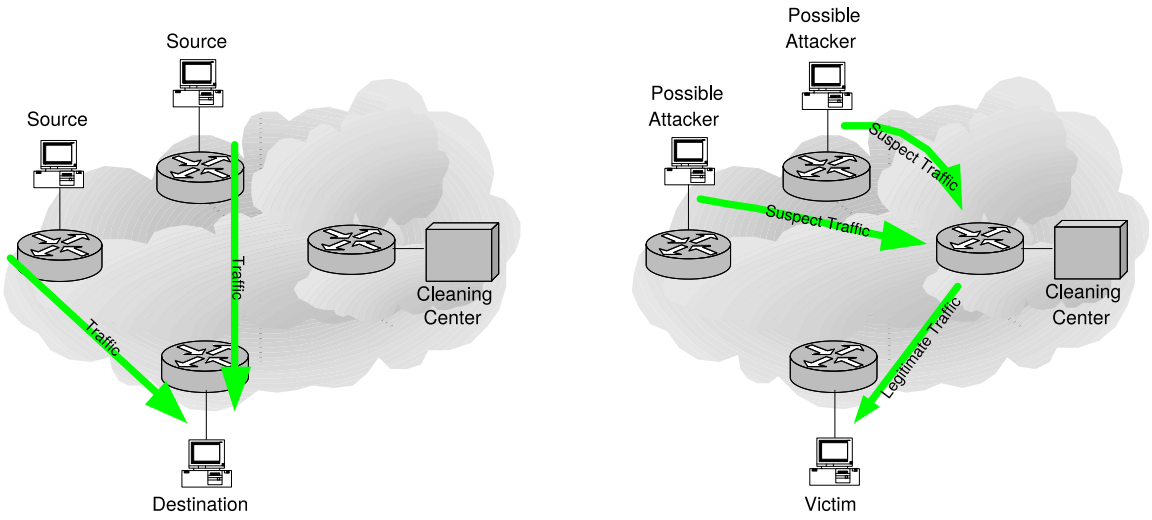
Zařízení s nainstalovaným *DynFW* klientem mohou získávat informace o aktuálních změnách v *Sentinel greylist*, nemusí to být dokonce ani Turris zařízení. Zároveň je každý den zveřejňován soubor se seznamem *Sentinel greylist* ve formátu *Comma Separated Values* (*CSV*), který měl v době psaní tohoto textu 5643 záznamů. [9]

4.2.4 Scrubbing Center

Scrubbing Center případně Cleaning Center je další možnost jak vyčistit datový provoz od záplavových útoků. Přes Scrubbing Center standardně datový provoz neprochází, ale je na něj nasměrován až v situaci, kdy je detekován útok (viz obrázek 10). Scrubbing Center pak datový provoz čistí o *DDoS* útoky a dále posílá jen legitimní provoz. Scrubbing centrum často nabízí poskytovatelé tranzitní konektivity jako službu pro *ISP*.

Výhoda této služby je, že se data filtrují už na páteřní síti, kde bývá dostatek konektivity. Nemusí se tak kompletně zahazovat provoz směřující na cílovou IP adresu útoku, a tak i cíl útoku může díky vyčištění provozu dále fungovat. [12]

[10] https://github.com/cowrie/cowrie



Obrázek 10: Scrubbing Center – toky dat. Vlevo běžná situace bez využití Scrubbing Center. Vpravo při čištění dat pomocí Scrubbing Center. [12]

Mezi nevýhody tohoto řešení patří, že v momentě přesměrování provozu přes Scrubbing centrum se zvýší latence a může vznikat *jitter*, nebo se změnit pořadí paketů, případně dojít ke ztrátě dat. Scrubbing centrum také v momentě přepojení na něj, s ohledem na to, že sleduje jen provoz v jednom směru, nedokáže vždy odlišit mezi legitimním provozem a útokem, a tak jsou některá spojení donucena spojit se znovu, což může vézt ke snížení kvality dodávané služby koncovému uživateli. Scrubbing centrum čistí jen příchozí datový provoz z internetu, nemohou tedy zajišťovat blokování útoků z vnitřní sítě do internetu. [11] [12]

4.2.5 Zhodnocení

IDS a *IPS* je naprosto ideální pro účely ochrany firemní sítě, kde chceme síť nejen jednorázově zabezpečit, ale i dále průběžně sledovat anomálie a mít celkový přehled o stavu firemní sítě. Pro účely zabezpečení zákazníků *ISP* ovšem není ideální kvůli své náročnosti na výpočetní výkon, neboť při potřebě kontrolovat desítky gigabitů dat by bylo nutné nasadit mnoho *IDS* senzorů. Zároveň hloubková inspekce paketů a tedy i sledování aplikačních dat zákazníků je minimálně nevhodná.

Detekce anomálií dle indických autorů je zajímavá především jako námět, jak by bylo možné ochranu proti *DDoS* řešit. Je zatím spíše teoretická a potřebuje ještě projít vývojem v praxi,

která prověří schopnosti této myšlenky. Princip sdílení detekovaných dat mezi *ISP* je pak chytrý nápad, který funkční detekci a blokaci dokáže dostat ještě o kus dále.

Turris Sentinel od CZ.NIC je pak metoda, zpracovaná nejen teoreticky, ale uvedená i do praxe. Každý router provádí detekci a informace o nebezpečných IP adresách sdílí s ostatními Turris routery. Kouzlo tohoto řešení je i v nasazení mnoha těchto routerů v různých sítích, kde fungují jako sondy a dodávají tak hodnotné informace o probíhajících útocích. *Turris Sentinel* se jeví jako dokonalé řešení pro koncové uživatele. Uživatel je chráněn, podílí se na detekci a vše si může nakonfigurovat na vyladěném Turris routeru.

Scrubbing centrum je výborné pro *ISP* jako záložní řešení pro situace, kdy dojde k akutnímu problému, který by už výrazněji ovlivňoval chod sítě *ISP*. Datový provoz dokáže vyčistit od masivních útoků a *ISP* a jeho zákazníci tak dále fungují. Není ale určené pro trvalé nasazení, tedy nemůže řešit neustále se vyskytující množství menších útoků, které proudí v podstatě neustále. Scrubbing centrum je zároveň určeno jen na čištění příchozí komunikace.

Zmíněná řešení mají každé jiné zaměření a také své klady a zápory. Pro běh u *ISP* žádné z nich nesplňuje všechny požadavky, které bychom potřebovali. Chtěli bychom detekovat útoky jak směrem z internetu do vnitřní sítě, tak i útoky směrem od zákazníků *ISP* do internetu, oboje bez nahlížení do aplikačních dat. Chtěli bychom mít možnost sledovat provoz všech zákazníků a to nezávisle na použitém routeru zákazníka. Dále bychom chtěli, aby detekce byla pokud možno nenáročná na výkon a nebylo nutné nakupovat drahé vybavení. Blokovat konkrétní IP adresy bychom chtěli jen v situaci, kdy tato blokace nebude mít vliv na žádný legitimní provoz.

Přestože se ani *Turris Sentinel* nejeví jako řešení pro naše účely, bylo by zajímavé se výsledky z tohoto projektu blíže zabývat. Informace o detekovaných útočících IP adresách z tohoto projektu jsou volně dostupné, což by umožňovalo prověřit, jak hodně by byla tato data vhodná pro blokaci útoků z internetu i pro účely *ISP*. Případně by bylo možné tato data využít jako zdroj potenciálních IP k blokaci a vyfiltrovat ty, které *ISP* považuje za vhodné pro blokaci i pro svoje účely. Zároveň by mohlo být zajímavé prověřit, zda by výsledky z dále představené detekce nemohly být použitelné i jako zdroj pro *Turris Sentinel*.

4.3 Nové řešení

Prozkoumali jsme bitevní pole kybernetických útoků a popsali, jaké mají útočníci možnosti, důvody a cíle. Zabývali se, proč má *ISP* zájem v ochraně svých uživatelů a kterým útokům by bylo možné na úrovni *ISP* čelit. Představili jsme také existující možnosti řešení tohoto problému. Nyní se budeme věnovat navrhovanému řešení, které spočívá v kompletním sledování síťového provozu, jeho vyhodnocováním a blokací problematických zdrojů.

Oproti detekci útoků u koncového zákazníka má *ISP* velkou výhodu v tom, že dokáže vyhodnocovat informace nejen o navazovaných spojeních daného zákazníka, ale o všech spojeních v celé své síti. Díky tomu může získat mnohem lepší přehled o situaci v síti a lépe odhalit útoky.

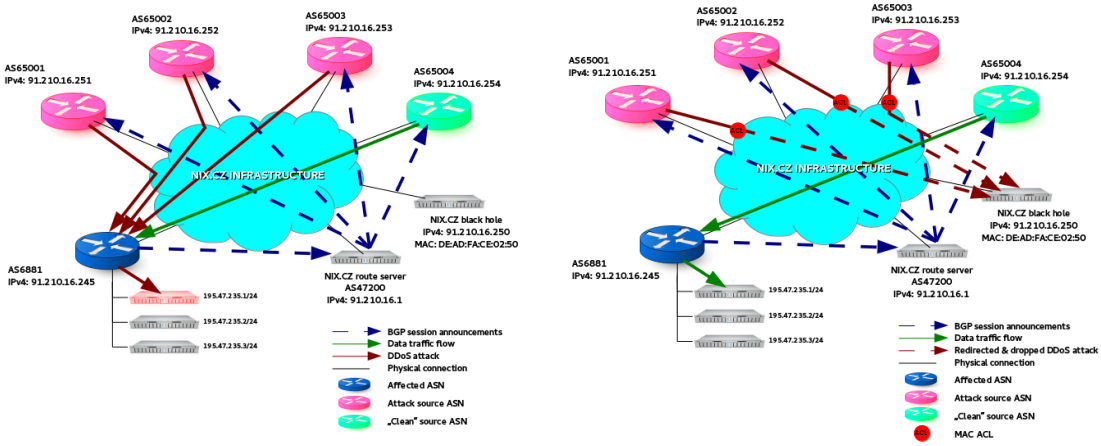
Další plus pro *ISP* je, že má informace o provozu směrem ke koncovému zákazníkovi už na páteřních bodech v jeho síti. Na první pohled by se zdálo, že provoz z internetu směrem k zákazníkovi bude na páteřním vstupu z internetu tentýž jako v místě předání konektivity u zákazníka (pomiňme, že má *ISP* více konektivit). V případě, kdy se bavíme o *DoS* a *DDoS* tomu tak ale být nemusí.

Dejme tomu, že má zákazník přípojku o rychlosti 200 Mbit, ale přitom je na něj směrován *DDoS* útok o velikosti 15 Gbit. Pokud má *ISP* páteřní vstup z internetu 40 Gbit, putuje zde ještě celý daný útok. Někde dále směrem k danému zákazníkovi už má *ISP* propoj jen 10 Gbit a tedy odsud dále putuje maximálně daných 10 Gbit. Samotná technologie u zákazníka může být gigabitový ethernet, tedy zde už dále putuje jen 1 Gbit a zákazníkovi se pak předává shapovaný provoz dle jeho služby 200 Mbit (ve skutečnosti se bude shaping provádět pravděpodobně dříve).

Zákazník by tedy při detekci na své přípojce sice zjistil, že se jedná o útok na zahlcení šířky pásma, ale již by nedokázal zjistit k jak velkému útoku dochází a nemohl by útoku ani efektivně čelit. V situaci, kdy tento útok dokáže detekovat *ISP* na páteřní lince a velikost útoku nepřekračuje šířku linky, může *ISP* provést blokaci přímo zde. Pokud útok již překračuje možnosti jeho linky, může využít možností *Remotely Triggered Black Hole* (*RTBH*)[11]

[11] *RTBH* je technika, jak prostřednictvím BGP protokolu definovat, který provoz (ze kterých IP adres) není legitimní a má být zahozen již o úroveň výše, aby tak nezatěžoval vlastní přístupovou linku [7].

a nebo se přímo obrátit na svého poskytovatele konektivity s žádostí o zablokování útoku na jeho infrastruktuře. Útok a jeho blokace prostřednictvím *RTBH* je na obrázku 11.



Obrázek 11: Využití *RTBH*: Vlevo je znázorněn *DDoS* útok, vpravo pak jeho blokace pomocí *RTBH*. [14]

V čem má naopak zákazník výhodu oproti *ISP*, je fakt, že ví, jaké služby plánuje využívat a provozovat. Může tedy otevřít do světa jen služby na konkrétních portech, případně některé služby otevřít jen pro konkrétní IP adresy v internetu.

Další výhoda zákazníka je, že může reagovat i na obsah paketů, které mu přichází. Navrhované řešení takové možnosti nemá a z dané komunikace má přístup pouze k hlavičkám paketů. *ISP* ani nepřísluší nahlížet do datového obsahu paketů zákazníků, šifrování takovouto možnost navíc efektivně znemožňuje.

Cílem navrhovaného řešení bude detekovat útoky z internetu ale i z vnitřní sítě směrem do internetu. Pro detekci se bude sledovat kompletní síťový provoz. Pro tento účel se využijí stávající zařízení a nebude tak nutné nakupovat a nasazovat další vybavení. Pro odhalení útoků z internetu se bude sledovat komunikace na vstupních rozhraních jednotlivých konektivit. Pro útoky z vnitřní sítě pak na hlavních bodech, kde se provádí *NAT*. Probíhající komunikace se na daných zařízení bude sledovat prostřednictvím NetFlow, což zároveň znamená, že se budou sledovat jen hlavičky paketů a ne aplikační data.

NetFlow data se budou odesílat na centrální kolektor, kde se budou i vyhodnocovat. Hledat se budou již známé vzorky problematického chování a také neobvyklé výchylky. Při odhalení nového typu útoku je možné vytvořit další filtr pro jeho detekci.

Na útoky z internetu se bude reagovat primárně automaticky. Po detekci útoku a prověření, že blokace útočící IP adresy neovlivní legitimní komunikaci dojde k blokaci takovýchto IP adres. Na útoky z vnitřní sítě bude reagováno upozorněním síťovým technikům *ISP*, kteří s pomocí těchto informací dále vyhodnotí a vyřeší situaci.

Blokace útočící IP adresy při automatizované reakci nebude trvalá, ale jen na omezenou dobu, která se bude prodlužovat v závislosti na opakovaných detekcích útoků z této IP adresy. Maximální doba blokace od posledního útoku je taktéž shora omezená.

Navrhované řešení bude detekovat skenování dostupných IP adres, nebo otevřených portů na protokolech TCP i UDP, *SYN flooding* a útoky na zahlcení šířky pásma. To především jako obecné útoky, bez rozlišení na jaké konkrétní služby se útočí. Detekovat se budou také útoky přímo na konkrétní služby, například *SSH*, *Telnet* nebo služby elektronické pošty. Bude se detekovat i TCP *Null*, *FIN* a *Xmas Tree* skenování, *ICMP flooding*, nebo využití zařízení zákazníka jako reflektor pro *DRDoS*. Útoky se budou detekovat na IPv4 i IPv6 protokolech.

4.3.1 NetFlow

Pro záznam navazovaných spojení bude využito *NetFlow*, což je protokol původně navržený firmou Cisco Systems pro efektivní monitorování síťového provozu na routerech a switchích této značky, nyní jej podporuje i mnoho síťových zařízení dalších značek.

NetFlow architektura se skládá z exportérů a kolektoru. Jako *NetFlow* exportéry se používají routery a switche v síti, případně sondy, které sledují procházející komunikaci. Tyto exportéry sledují statistiky o spojeních a ty pak odesílají na *NetFlow* kolektor jako *NetFlow* data.

NetFlow kolektor je pak zařízení s velkým úložným prostorem, kde se *NetFlow* data po stanovenou dobu ukládají. *NetFlow* data obsahují informace o zdrojové a cílové IP adrese, portu, začátku a ukončení spojení, množství přenesených bytů a paketů a také další informace. [1]

4.4 Implementace řešení

Pro sběr statistik o síťových tocích na páteřních linkách se využívají páteřní BGP routery Cisco. Statistiky se sbírají na všech konektivitách a propojeních s dalšími *ISP*. Dále se tato data sbírají na všech rozhraních směrem k zákazníkům za routery provádějícími *NAT*. Zde se ke sběru využívají routery od firmy Mikrotik.

Jako centrální kolektor je využit *GNU/Linux* server, na kterém je pro každý *NetFlow* exportér spuštěna instance démonu *nfcapd* (*NetFlow Capture Daemon*), která *NetFlow* data sbírá a ukládá na pevný disk.

Posbírané *NetFlow* data se využívají nejen k detekci útoků, ale dále například k vytváření grafů toků jednotlivých zákazníků nebo k uchovávání provozních a lokalizačních údajů pro vyšetřování trestné činnosti dané zákonem.

Na pevném disku jsou *NetFlow* data rozdělena do adresářů dle exportérů, dále do adresářů dle dní a nakonec do souborů dle času. *NetFlow* data se používají k více účelům, je nutné neztrácet informace o zdroji, ze kterého pocházejí a moci data po expiraci mazat. Například data z 30.11.2021 z doby od 12:30 do 12:40 z exportéru Letná jsou uloženy v souboru:

```
/netflow/Letna/2021-11-30/nfcapd.202111301230
```

Pro účely dané zákonem je potřeba *NetFlow* data uchovávat po dobu půl roku a je tedy nutné uchovávat jich velké množství. Pro archivaci je logicky vhodné data komprimovat, pro jejich zpracování však nikoliv, jelikož to zvyšuje nároky na výkon a prodlužuje dobu jejich zpracování. Jako vhodná varianta se tak jeví nekomprimovat data hned při jejich sběru, ale až později po tom, co už k nim není potřeba intenzivně přistupovat. Jejich zpracování je díky tomu efektivní a kompresi lze provádět dávkově v nočních hodinách, kdy se server využívá minimálně. Realizace je velmi jednoduchá, stačí na to triviální skript pouštěný plánovačem *Cron*:

```
#zpracovat včerejší data
datum=$(date "+%Y-%m-%d" -d "-1 day")
#komprimovat bz2 (-J 2) ve všech adresářích s NetFlow
```

```
find /netflow/ -mindepth 2 -maxdepth 2 -type d -name $datum \
-exec nfdump -R {} -J 2 \;
```

V adresáři `/netflow` jsou data ze všech exportérů, odsud se zálohují a provádí se jejich expirace. Aby při vytvoření nového exportéru nebylo nutné zasahovat do skriptu pro detekci útoků a evidovat, které exportéry jsou konektivita nebo vnitřní síť, vytváří se symbolický odkaz dle příslušnosti do adresářů `/netflow-konektivity` nebo `/netflow-zakaznicke`. Skripty pro detekci útoků pak čtou z těchto adresářů dle příslušného typu.

```
ln -s /netflow/Letna/ /netflow-zakaznicke/
ln -s /netflow/NIX/ /netflow-konektivity/
```

O detekce útoků se stará *Python* skript. Středobodem tohoto skriptu je funkce getStatNFData s parametry filtr, agregační klíč, a nepovinné parametry pořadí a minimum.

```
def getStatNFData(filtr, agreg, poradi='flows', minimum=None):
```

Tato funkce se stará o vyčtení agregovaných údajů z *NetFlow* dat ze všech patřičných exportérů. Dle zadaného filtru se určují požadované údaje. Agregační klíč určuje, dle jaké hodnoty se budou seskupovat záznamy, obvykle se používá srcip nebo dstip, případně jejich varianty po provedení *NAT* – nsrcip a ndstip. Záznamy je možné seskupovat i dle jiných údajů uložených v *NetFlow* datech, například srcport nebo dstport. Parametr pořadí určuje řazení záznamů, standardně se využívá flow – pořadí dle počtu toků, je možné využít též bytes – pořadí dle přenesených dat. Parametr minimum pak zajistí, aby funkce vracela jen ty záznamy, které po agregaci obsahují údaje alespoň o takovém množství toků nebo množství dat, tedy dle proměnné v parametru pořadí. Funkce vrací seznam slovníků s daty, klíčem slovníku jsou hodnota klíče (tedy např. zdrojová IP adresa při agregačním klíči srcip) a počet nalezených toků. Příklad získaných dat může vypadat například takto:

```
[{'val': '198.51.100.7', 'fl': '368'}, {'val': '2001:db8::27', 'fl': '78'}]
```

Funkce getStatNFData slouží jako prostředník ke získání agregovaných dat z *nfdump*. *Nfdump* je aplikace určená ke zpracování sesbíraných *NetFlow* dat. Má podobnou syntaxi

filtru jako *tcpdump*. Dokáže zobrazovat *NetFlow* data, vytvářet statistiky o tocích, bytech nebo paketech. Dokáže také sestavovat agregované statistiky. Umí více výstupních formátů, včetně *CSV* pro další strojové zpracování. [6]

V následujících kapitolách bude popsána detekce útoků z vnitřní sítě a detekce útoků z internetu následovaná blokací útoků z internetu. Zvlášť je popsána detekce a blokace rozesílání nevyžádané pošty, kde nestačí metody popsané v předchozích detekcích a je potřeba reagovat okamžitě na vzniklou situaci. Další informace o použití vytvořených aplikací je možné najít v souboru `README.md` v GIT repozitáři (odkazy viz. příloha B na straně 63) nebo ZIP archivu přiloženém k této práci.

Detekce útoků z vnitřní sítě a detekce útoků z internetu mají každá svá specifika a vyhodnocují se z jinde získaných dat a tak danou detekci vykonává i jiný skript. Co mají obě detekce společné, je spouštění detekce po rotaci souborů s *NetFlow* daty, tedy po dokončení a uzavření souborů se statistikami. Časy rotace jsou však rozdílné kvůli různým požadavkům na dobu reakce. V případě detekce útoků z internetu je to každou minutu, neboť se výsledky této detekce používají především k automatické blokaci. V případě detekce útoků z vnitřní sítě pak každých 10 minut, zde detekované problémy prověřuje a vyhodnocuje obsluha.

4.4.1 Detekce útoků z vnitřní sítě

Detekce útoků z vnitřní sítě se provádí z *NetFlow* dat posbíraných na hlavních bodech na rozhraních směrem k zákazníkům. Nalezneme zde:

1. Komunikaci na lokálních IPv4 adresách dle RFC1918 zákazníků, kteří nemají přiděleny vlastní veřejné IPv4 adresy. Jako příklad je v této práci použit privátní rozsah `10.0.0.0/8`.

2. Komunikaci na veřejných IPv4 adresách zákazníků, kteří mají přímo přidělenu veřejnou IPv4 adresu nebo síťový rozsah, a tedy jim *ISP* neprovádí *NAT*. Jako příklad je použit dokumentační rozsah `198.51.100.0/24`.

3. Komunikaci na veřejných IPv6 adresách, kde mají zákazníci přiřazen svůj prefix /56, případně /48. Jako příklad je používám dokumentační rozsah `2001:db8::/32`.

Všechny tyto varianty pokrývá konstanta `SRC_ISP` ze skriptu `detekce.py`, která pomocí syntaxe *nfdump* specifikuje síťové rozsahy využité na straně zákazníků u daného *ISP*.

```
SRC_ISP="(src net 10.0.0.0/8 or src net 198.51.100.0/24 \
or src net 2001:db8::/32)"
```

Především v případě IPv4 komunikace není automatem jednoduše rozpoznatelné, zda se jedná o komunikaci jen jednoho uživatele, nebo například až celé organizace. Daný zákazník totiž sám na svém zařízení provádí *NAT*, který za jednu IP adresu může schovat neomezené množství IP adres. To detekci ztěžuje. Cílem detekce útoků z vnitřní sítě však není provádět automatické blokace, ale informovat správce *ISP* sítě, kteří hlouběji vyhodnotí situaci a podniknou patřičné kroky.

Jak probíhá detekce útoků si ukážeme na příkladu. Naším cílem bude odhalit skenování otevřených portů *SSH* z vnitřní sítě. Budeme hledat lokální IP adresy, které otevírají podezřele velké množství spojení na *SSH* port, kde buď nedostávají žádnou odpověď zpět nebo na příchozí odpověď nereagují a tedy nedokončí *třícestný handshake*. K získání těchto informací slouží následující kód:

```
getStatNFData("proto tcp and dst port 22 and flags S and not flags A \
    and SRC_ISP", "srcip", minimum=60*TIME/10)
```

Tedy hledáme komunikaci na protokolu TCP, kde je cílová služba *SSH* (port 22). Dále podmínkou definujeme, že hledáme takové toky, kde byla odeslána zpráva SYN, ale ne zpráva ACK. Zajímají nás toky, které pochází z vnitřní sítě (tu jsme si nadefinovali již dříve v konstantě `SRC_ISP`). Data agregujeme dle zdrojové IP adresy (srcip). Parametr minimum udává, že se budeme zajímat jen o IP adresy, které otevírají popisovaná spojení v průměru častěji než jednou za 10 sekund.

Pro správce, který má útoky řešit, by bylo u takto nalezených IP adres ještě vhodné vědět, na kolik IP adres se takto neúspěšně snažily připojit. To zjistíme následujícím příkazem:

```
getStatNFData("proto tcp and dst port 22 and flags S and not flags A \
    and src ip %s", "dstip")
```

Na místo `%s` dosadíme jednotlivé nalezené lokální IP adresy z příkladu výše. Jelikož agregační funkce zde je cílová IP adresa (dstip), zjistíme tímto dotazem všechny IP adresy na které se daná lokální IP adresa snažila otevřít daná spojení, jejichž počet nás zajímá.

Podobným způsobem, jako je skenování *SSH*, budeme provádět i detekci útoku na *Telnet* (TCP port 23), poštovní služby (TCP porty 25, 465 a 587), nebo služby používané na platformě Mikrotik, jako je Winbox (TCP port 8291), Mikrotik API (TCP port 8728) a Mikrotik API-SSL (TCP port 8729). Nevynecháme ani oblíbený cíl, službu Microsoft DS síťového protokolu SMB (port 445).

Budeme též detekovat útok z WS-Discovery (UDP port 3702), kde by mohli útočníci využít zařízení našich zákazníků jako reflektor a dosáhnout tak velmi vysokých pákových efektů[12]. Na dotazy z internetu na tomto portu reagují jen zařízení s chybnou implementací, neboť protokol byl navržen, aby reagoval pouze na multicast adrese 239.255.255.250 určené pro *LAN* a ne na unicast adrese [22].

Všechny výše zmíněné detekce jsou příklady specifických detekcí pro konkrétní služby, které mohou být použity pro přesnější detekci v závislosti na funkci daného protokolu. Nastíněným způsobem lze relativně jednoduše připsat další detekce.

Středobodem detekce útoků z vnitřní sítě však budou obecnější detekce, tak aby pokrývaly pokud možno všechny možné varianty. Pro TCP protokol je připravena detekce *SYN scan*, nastíněna detekce *Null scan*, *FIN scan* a *Xmas Tree scan*. Pro UDP protokol detekce skenování nebo útoku. Pro *ICMP* detekce zaměřená jak na toky, tak na množství dat. Nebude chybět ani detekce na protokolech mimo TCP, UDP a *ICMP*. Všechny tyto detekce útoků jsou popsány v následující kapitole „Detekce útoků z internetu“ ve složitější variantě, kde dochází i k rozhodování pro následnou automatickou blokaci.

4.4.2 Detekce útoků z internetu

Detekce útoků z internetu se provádí z *NetFlow* dat posbíraných na páteřních rozhraních s konektivitami. Zde se nevyskytuje žádná komunikace na lokálních IP adresách, ale všechny zdrojové i cílové IP adresy zde jsou veřejné. Konstanta `DST_ISP` s celým rozsahem IP adres ve skriptu `detekce_internet.py` je v našem případě definována takto:

[12]Na protokolu WS-Discovery bylo naměřeno až 15 300 % pákového efektu při využití reflektoru [22].

```
DST_ISP="(dst net 198.51.100.0/24 or dst net 2001:db8::/32)"
```

Dále jsou ještě nadefinovány konstanty `DST_SNAT` a `SRC_SNAT` s rozsahy *ISP*, které se využívají jen pro *source NAT*, tedy *NAT* zdrojové IP adresy. *Destination NAT*, tedy *NAT* cílové IP adresy, se u této skupiny IP adres neaplikuje. To se děje u IP adres pro běžné koncové zákazníky, kteří nemají přiřazenu vlastní veřejnou IP adresu a kterým se tak při připojování na služby v internetu musí provést *source NAT* jejich lokální IP adresy na sdílenou veřejnou IP. Také se definují konstanty `DST_NOUSE` a `SRC_NOUSE` s rozsahy, které aktuálně nejsou využity. Jak `SNAT`, tak `NOUSE` rozsahy se využijí dále při některých detekcích, kdy se z IP adres z těchto rozsahů očekává jiná odpověď na dotazy z internetu, než v případě, kdy se jedná o IP adresu, na které je provozován například server, jehož služby jsou z internetu dostupné.

```
DST_SNAT="(dst net 198.51.100.64/22 or dst net 198.51.100.128/22)"
SRC_SNAT="(src net 198.51.100.64/22 or src net 198.51.100.128/22)"
DST_NOUSE="(dst net 198.51.100.192/22 or dst net 2001:db8:f000::/36)"
SRC_NOUSE="(src net 198.51.100.192/22 or src net 2001:db8:f000::/36)"
```

Při jednotlivých detekcích se hledají IP adresy v internetu, které provádí daný útok. Než se přistoupí k blokaci takto nalezených IP adres, je nutné provést různé kontroly, aby se blokací nezpůsobilo více problémů, než užitku. V případě mnoha typů útoků si nemůžeme být jisti, zda není zdrojová IP adresa falšována. Pokud budeme slepě blokovat každou IP adresu, ze které budeme útok detekovat, sice útok odfiltrujeme, ale s tím můžeme zablokovat i legitimní komunikaci. Útočníci by takového jednání mohli zneužít a k zamezení komunikace mezi naší sítí a konkrétní IP adresou v internetu by jim v takové případě stačilo zfalšovat svoji zdrojovou IP adresu a použít místo své skutečné právě tuto konkrétní. Proto před blokací IP adresy budeme kontrolovat, jestli s danou IP adresou probíhá z naší sítě i jiná komunikace, než detekovaný útok. Budeme vycházet z předpokladu, že pokud jinou komunikaci než útoky nenajdeme, je možné přistoupit k blokaci.

Po detekci útoku a prověření, že s danou IP adresou neprobíhá legitimní komunikace, je daná IP adresa zaznamenána do databázové tabulky `net_blokace`. Navržená struktura databázové tabulky je v tabulce 2. K dané IP zaznamenáme též aktuální čas do sloupce

`blokace_od` a aktuální čas inkrementovaný o jednu hodinu do sloupce `blokace_do`. Pokud už záznam s danou IP v tabulce existoval, je jen aktualizován a to tak, že je prodloužen čas `blokace_do` o jednu hodinu. Tedy při každém běhu detekce jsou nalezené útočící IP adresy přidány na hodinu do blokace, nebo jim je jejich blokace o hodinu prodloužena. Kolik druhů útoků provádějí v této metodě nehraje roli. Pokud už je blokace IP adresy na delší dobu než jeden den, dále se čas blokace neprodlužuje a zároveň se takové IP adresy ani neprověřují na další útoky a šetří se tak výkon.

Tabulka 2: Návrh databázové tabulky net_blokace

net_blokace		
`IP`	varbinary(16)	«PK»
`maska`	tinyint(4) unsigned	«PK»
`blokace_od`	datetime	výchozí: aktuální čas
`blokace_do`	datetime	

SYN flooding / scan

Pro detekci SYN flooding nebo SYN scan budeme prověřovat datové toky z IP adres, kde najdeme v průměru více než jeden datový tok za sekundu (proto `minimum=TIME*60`), které odpovídají stavu, že v daném spojení byl vyslán SYN požadavek, ale nedošlo v něm k odesláni ACK, tedy nebyl dokončen *třícestný handshake*:

```
getStatNFData("proto tcp and flags S and not flags A and DST_ISP, \
    "srcip", minimum=TIME*60)
```

Abychom skutečně přistoupili k blokaci, musíme ještě prověřit, že z dané IP adresy neproudí i legitimní spojení. To budeme provádět tak, že z dané IP adresy hledáme toky, které neodpovídají dříve hledanému, tedy takové, které mají dokončen *třícestný handshake*. Ani toky jen se samotným RST příznakem nebudeme považovat za legitimní. Zároveň při dané situaci, kdy bylo provedeno průměrně více než jeden SYN požadavek za každou sekundu, nebudeme ani toky s méně než čtyřmi pakety považovat za legitimní.

```
getStatNFData("not ( (proto tcp and flags S and not flags A) or \
   (proto tcp and flags R and not flags UAPSF) or packets<4 ) and \
   %s and src ip DST_ISP"), "srcip")
```

Pokud bude detekován SYN flooding, ale zároveň bude z dané IP detekována i legitimní komunikace, nebude daná IP blokována a bude jen vypsáno upozornění pro prověření technikem. Může se stát, že útočník bude kombinovat SYN flooding s jiným útokem. Nebo naopak, že bude nějaký rozumný důvod pro to, aby vysílal takové množství SYN požadavků, v tom případě by se jednalo o legitimní komunikaci. V obou případech je možné detekci patřičně upravit. Technik má i možnost přidat útočníka na seznam k blokaci ručně do konkrétního data, ale to by bylo až krajní řešení.

UDP flooding / scan

Dále budeme detekovat záplavové útoky na UDP a skenování UDP portů. Možnosti detekce v případě UDP jsou úplně jiné než v případě TCP, neexistuje zde žádný *třícestný handshake* ani příznaky. Kvůli tomu nejsme schopni stejně jednoduše jako při TCP detekci rozlišit, která strana spojení iniciovala. Samozřejmě můžeme vycházet z času počátku spojení, ale to by znamenalo porovnávat a párovat vstupní a výstupní *NetFlow* data, což je náročné na výkon a prodlužuje to tak dobu zpracování.

Bez informace o tom, kdo spojení iniciuje se samozřejmě neobejdeme. Nevěděli bychom, jestli z dané IP v internetu potenciálně proudí útok na našeho zákazníka a nebo naopak, jestli od našeho zákazníka proudí potenciálně útok do internetu. Zjistíme to ale efektivnější metodou.

Budeme detekovat kolik datových toků nám z konkrétních IP adres do vnitřní sítě přichází a kolik datových toků putuje na tyto IP obráceným směrem. Zde využijeme dříve nadefinované konstanty `DST_SNAT`, `DST_NOUSE`, `SRC_SNAT` a `SRC_NOUSE`. Chování budeme totiž rozlišovat dle toho, zda komunikace probíhá s rozsahem, kde mohou být provozovány veřejně dostupné služby. Na IP rozsazích, kde je jen source *NAT*, není možné navazovat spojení z internetu, a tedy pokud zde najdeme komunikaci směřující do vnitřní sítě, ale ne obráceným směrem, je zřejmé, že se nejedná o legitimní komunikaci.

Pro úvodní zjištění, které IP adresy budeme dále prověřovat, použijeme následující kód, který získá IP adresy, ze kterých přicházely toky na protokolu UDP, kde se daný datový tok skládal jen z jednoho paketu. Zajímat nás budou jen takové IP, které takto vyslali průměrně alespoň jeden takový datový tok každou sekundu:

```
getStatNFData("proto udp and packets<2 and DST_ISP", \
    "srcip", minimum=TIME*60)
```

Takových IP adres bude mnoho a určitě mezi nimi budou mimo jiné figurovat i *DNS* servery, u kterých je naprosto logické, že jejich datový tok v jednom směru je běžně právě jeden paket. Proto budeme prověřovat i UDP komunikaci jdoucí opačným směrem, v tomto směru už nás budou zajímat i toky, které mají více než jeden paket:

```
getStatNFData("proto udp and dst ip %s", "dstip")
```

V případech, kdy směrem z vnitřní sítě máme reakci alespoň na 30 % toků putujících z internetu, považujeme tuto komunikaci za legitimní. Případně by bylo možné informovat techniky o potřebě prověřit tuto komunikaci, ale určitě ji nemůžeme automaticky blokovat, neboť zde s pravděpodobností blížící se jistotě najdeme i legitimní komunikaci.

Pro detailnější rozhodování dále vyčteme informace o procházející komunikaci s touto IP adresou dle rozdělení na rozsah *ISP*, kde mohou nebo nemohou být provozovány veřejně dostupné služby:

```
"proto udp and packets<2 and src ip %s and (DST_SNAT or DST_NOUSE)"
"proto udp and packets<2 and src ip %s and not (DST_SNAT or DST_NOUSE)"
"proto udp and dst ip %s and (SRC_SNAT or SRC_NOUSE)"
"proto udp and dst ip %s and not (SRC_SNAT or SRC_NOUSE)"
```

Vyčteme také informace o odpovědích *ICMP Destination Unreachable* z vnitřní sítě. Ty se vrací například, když port, se kterým se protistrana snaží spojit, je uzavřený. V případě protokolu IPv4 je *ICMP Destination Unreachable* typ 3 [2], v případě IPv6 je to typ 1 [3].

```
getStatNFData("dst ip %s and icmp-type 3", "dstip")
getStatNFData("dst ip %s and icmp-type 1", "dstip")
```

Také zjistíme počet protistran se kterými tato IP adresa komunikuje:

```
getStatNFData("proto udp and src ip %s", "dstip")
```

A opět musíme detekovat, jestli z dané IP nenacházíme nějaký legitimní datový provoz. Mezi ten nepočítáme *ICMP* komunikaci, TCP komunikaci s toky o jediném paketu nebo s nedokončeným *třícestným handshakem* a také zbývající UDP komunikaci.

```
getStatNFData("src ip %s and not proto udp and (not proto tcp or \
    (proto tcp and packets>1 and not (flags S and not flags A))) \
    and not proto icmp and not proto icmp6", "srcip")
```

Získané informace vyhodnotíme. Budeme blokovat jen takové IP adresy, se kterými nedocházelo k žádné legitimní komunikaci, ani nebyl nalezen žádný datový tok směrem z vnitřní sítě z rozsahu bez veřejných služeb. Další podmínky jsou, že z rozsahu pro veřejné služby přichází odpověď v maximálně 10 % případů a zároveň se vrací více než 10 % *ICMP Destination Unreachable* zpráv.

Další detekce

Součástí detekce na protokolu TCP je též *Null scan*, tedy vyhledávání dat dle filtru viz. řádek 1, *FIN scan* s filtrem viz. řádek 2 nebo *Xmas Tree scan* s filtrem viz. řádek 3. Sleduje se též neobvyklé množství provozu na protokolu *ICMP* viz řádek 4, například kvůli detekci *ping flood*. Nevynechávají se ani jiné protokoly, pro které nemáme specifičtější filtr, a sleduje se také neobvyklé množství provozu na těchto protokolech viz řádek 5. Detekce všech těchto typů je zatím především informačně, aby bylo v případě výskytu takových útoků možno zareagovat a dopsat patřičné detekce, které by už mohly zařídit přímo blokaci.

```
proto tcp and not flags ASRUPF
proto tcp and flags F and not flags ASRPU
proto tcp and flags UPF and not flags ASR
proto icmp or proto icmp6
not proto tcp and not proto udp and not proto icmp and not proto icmp6
```

Abychom byly schopni odhalit cíl útoku na zahlcení šířky pásma, ať už *DoS* nebo *DDoS*, sledujeme cíle na které proudí velké množství dat. V ukázce níže je filtr pro informování o překročení průměrné rychlosti 1 Gbit za daný čas. Stejným způsobem sledujeme i zdrojové IP ze kterých proudí velké množství dat. Například, pokud by cílem útoku nebyl jeden koncový zákazník, ale přímo *ISP*. Obě varianty detekcí útoku na zahlcení šířky pásma jsou bez automatické reakce na takovou situaci a řešení se ponechává na rozhodnutí technika. Daný útok by pak technik mohl blokovat pomocí filtrů na páteřních zařízeních, nebo pomocí *RTBH*, v závislosti na mohutnosti daného útoku.

```
getStatNFData("DST_ISP", "dstip", poradi='bytes', \
    minimum=1000*1000*1000/8*60*TIME)
```

4.4.3 Blokace útoků z internetu

Po detekci útoků z internetu je spouštěna blokace nalezených útočících IP adres. Blokace se provádí na několika desítkách hlavních routerech, které jsou hlavní pro jednotlivé oblasti. Blokaci zařizuje stavový firewall, který blokuje nová spojení z internetu do vnitřní sítě z IP adres uvedených na seznamu „blokace“:

```
/ip firewall filter
add action=drop chain=forward connection-state=new src-address-list=blokace
```

Z vnitřní sítě lze spojení s těmito blokovanými IP adresami navázat a odpovědi protistrany bez újmy firewallem projdou. Tato funkčnost je cíleně zachována, neboť i přes všechny kontroly může s detekovanou útočící IP adresou probíhat občasná regulérní komunikace, kterou chceme pro zákazníky zachovat. Je ale otázka, zda tímto nenecháváme možnost např. komunikovat napadenému zařízení zákazníků se servery botnetu a tak by mohlo být zajímavé více prověřit takto povolenou komunikaci.

Seznam „blokace“ je udržován aktuální skriptem `blokace_internet.py`, který se vždy po proběhlé detekci útoků z internetu připojí na každý hlavní router v jednotlivých oblastech a zkontroluje obsah seznamu „blokace“ vůči databázi. Chybějící záznamy přidá, již neaktuální záznamy smaže.

4.4.4 Detekce a blokace rozesílání nevyžádané pošty

Vytvořené řešení pro detekci útoků z vnitřní sítě nedostačuje na potřeby ochrany proti rozesílání spamu. Doba mezi počátkem rozesílání nevyžádané pošty a reakcí v řádu minut je příliš dlouhá, za tu dobu je možné odeslat i tisíce spamů. Přitom zařazení IP adresy na černou listinu může způsobit i jediný spam. Na efektivní ochranu je potřeba takové řešení, které zamezí právě i odeslání jediného takového e-mailu.

V této kapitole se nebudeme zabývat detekcí a blokací nevyžádané pošty na poštovním serveru. Řešení pro tento účel existují a běžně se používají. My se budeme zabývat detekcí a blokací na úrovni *ISP*, kdy chceme detekovat takový provoz nejen pro *SMTP* servery provozované *ISP*, ale obecně jakoukoliv *SMTP* komunikaci, u které *ISP* nechce, a díky šifrování většinou ani nemůže, sledovat obsah e-mailů.

V takové situaci samozřejmě vůbec není možné zjistit zdrojovou ani cílovou e-mailovou adresu nebo text daného e-mailu. Musíme si poradit jinak. Využijeme časté nenasytnosti malwaru rozesílajícího nevyžádanou poštu, jehož cíl je odeslat opravdu masívní množství takové pošty v co možná nejkratším čase, dokud jeho IP adresa ještě nefiguruje na černých listinách. Z toho důvodu se snaží otevřít velké množství spojení na mnoho poštovních serverů.

Na takovéto jednání budeme reagovat a to přímo na routerech, přes které daný provoz putuje. Princip spočívá ve sledování počtu zároveň otevřených spojení z jednotlivých zákaznických IP adres na poštovní služby (porty 25, 465 a 587). Pokud se z některé IP adresy otevře více než 30 spojení zároveň, bude takováto komunikace zablokována. Na první pohled by to mohlo vypadat, že je takto možné odeslat až 30 e-mailů, ve skutečnosti se však standardně neodešle ani jeden. Pokud se skutečně jedná o malware a funguje popsaným způsobem, začne otevírat velké množství spojení na poštovní servery a snaží se rozeslat spam. Započít 30 spojení však stihne dříve, než se mu vůbec podaří první e-mail odeslat. A proto ještě než vůbec první spojení dokončí a první e-mail odešle, je již zablokován.

V této práci využijeme platformu Mikrotik, ale obdobná pravidla lze vytvořit ve firewallu i na zařízeních jiných výrobců, nebo například pomocí *nftables* nebo *iptables* na systému *GNU/Linux*. Základní struktura pravidel ve firewallu pro tento účel vypadá v případě IPv4 následovně:

```
/ip firewall filter
add chain=forward out-interface=rozhrani_do_internetu protocol=tcp \
    dst-port=25,465,587 action=jump jump-target=smtp-omezeni
add chain=smtp-omezeni src-address-list=spam_blokace action=reject \
    reject-with=icmp-admin-prohibited
add chain=smtp-omezeni address-list=spam_blokace address-list-timeout=1d \
    connection-limit=30,32 action=add-src-to-address-list
```

Zásadní jsou dvě pravidla, pravidlo na 4. řádku kontroluje překročení počtu naráz otevřených spojení na poštovní služby. V případě překročení mezní hodnoty (30 spojení na konkrétní IP adresu – tedy s maskou /32) přidá zdrojovou IP adresu na jeden den na seznam `spam_blokace`. Pravidlo na 3. řádku pak blokuje komunikaci s poštovními službami z IP adres na seznamu. Pravidlo na 2. řádku je kvůli lepší přehlednosti a také optimalizaci. Díky němu máme všechna pravidla ve vlastním řetězu pravidel (chain) a zároveň v dalších pravidlech už znovu neprovádíme kontrolu protokolu a cílového portu.

Na všech zařízeních, která detekci a blokaci provádějí, jsou pravidelně kontrolovány IP adresy na seznamu `spam_blokace` a zákazníci, kterých se tato blokace týká, jsou o jejich blokaci a ukončení blokace informováni.

Před nasazením takových pravidel je vhodné informovat koncové zákazníky o aplikaci tohoto řešení a mít zmínku například ve všeobecných obchodních podmínkách o této praktice. Především pro firemní zákazníky, kteří provozují *SMTP* server je tato informace důležitá, aby patřičně k tomuto opatření mohli nastavit svůj *SMTP* server. Důležité je hlavně pro konfiguraci počtu vláken, ve kterých daný *SMTP* server e-maily odesílá.

Na první pohled by pro koncové zákazníky tato pravidla mohla vypadat jako omezující, v běžných případech však nejsou. Naopak, pokud *ISP* tento problém vyřešen nemá, může nastat situace, kdy je napadení zařízení jednoho zákazníka omezující pro využití poštovních služeb zákazníka jiného. V případě výskytu problému tato pravidla efektivně zasáhnou a zachrání zákazníka od přidání jím využívané veřejné IP adresy na internetové černé listiny, ze kterých již nemusí být jednoduché dosáhnout smazání, především při opakované blokaci. Někdy bývá odstranění z těchto internetových černých listin i zpoplatněno.

Tato ochrana pak lze kombinovat s dalšími detekcemi a na seznam `spam_blokace` mohou být IP adresy přidávány i z jiných zdrojů.

Pro koncové zákazníky, kteří sdílí veřejnou IPv4 adresu s dalšími zákazníky se ochrana jeví jako nutnost, neboť bez ní jsou zařazením takové IP na internetovou černou listinu ovlivněni všichni zákazníci, kteří tuto IPv4 adresu sdílí.

Pro zákazníky s vlastní veřejnou IPv4 adresou tato kontrola nezbytná není, především pokud mají svoji síť ochráněnu vlastním způsobem. Je možné ji ponechat jako záložní řešení jejich ochrany, nebo ji vypnout, pokud jim nevyhovuje. Pro tyto případy je možné přidat před výše vypsaná pravidla různé výjimky. Například takovéto:

```
add chain=smtp-omezeni src-address=198.51.100.7 action=accept
add chain=smtp-omezeni src-address=198.51.100.14 \
    dst-address=203.0.113.25 action=accept
```

První pravidlo ukazuje možnost jak vynechat z kontroly konkrétní zákaznickou IP adresu. Druhé pravidlo pak jen komunikaci s konkrétním poštovním serverem, v případě zablokování by pak odesílání e-mailů přes tento poštovní server ovlivněn nebyl. Pokud by bylo potřeba, je možné vytvořit i další výjimky, vlastně jakékoliv pomocí firewallu definovatelné.

5 Testování vyvinutých aplikací

5.1 Přetížení rádiových spojů

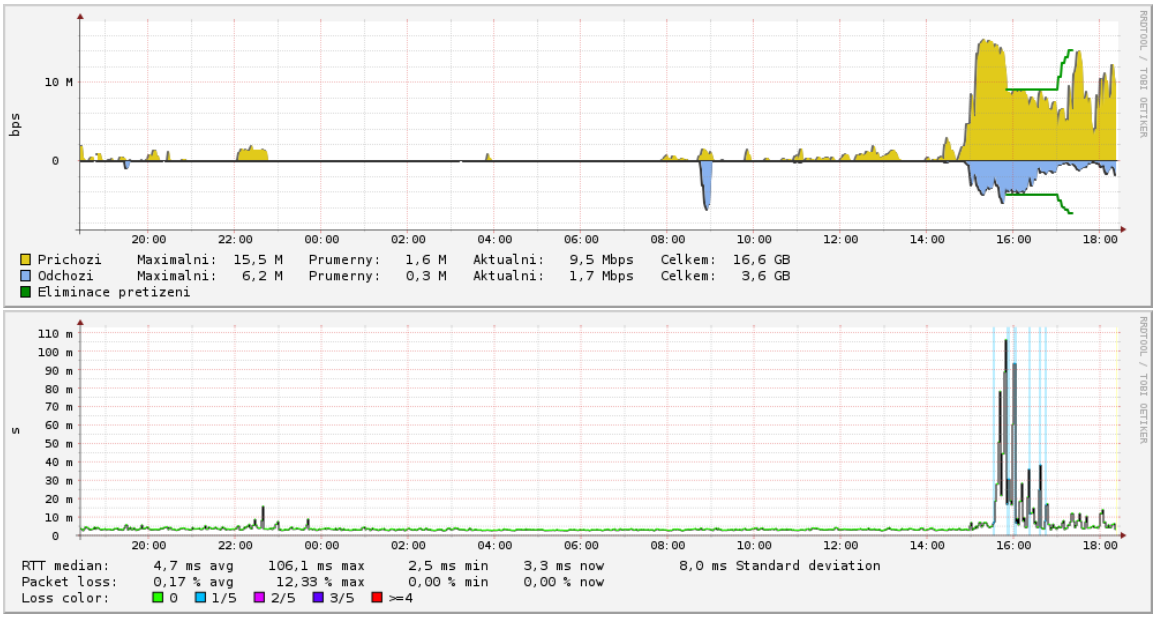
Jelikož aplikace popisovaného řízení shapingu je ovlivněna mnoha vstupními proměnnými měnícími se v čase a také reagujícími na samotné řízení shapingu, bylo schůdnější testovat aplikaci přímo v reálném prostředí přímo u *ISP*. Nejprve bylo testováno, zda aplikace správně nachází zákazníky, kteří trpí přetížením rádiového spojení. Když aplikace začala navrhovat vhodné rychlosti, zatím je nerealizovala, jen bylo dle dostupných statistik sledováno, jestli jsou návrhy odpovídající. V momentě, kdy byly výsledky dosaženy, bylo testováno nastavování navržených rychlosti ručně velmi malé skupině zákazníků, kde se detailně sledovalo chování po úpravách. Až po odladění bylo přistoupeno ke skutečnému řízení shapingu aplikací, ale s mnoha limity, které měly za cíl zajistit udržení shapingu v mezích i v případě chyby v aplikaci. Následně byla aplikace detailně sledována a kontrolována její činnost.

Při vývoji vznikla chyba, kdy nebylo ošetřeno, aby aplikace řídila shaping jen na bezdrátových spojích. Ve statistikách informace o nastavené službě zákazníka byla a tak byl takto nalezen jeden zákazník s optickým tarifem. Po prozkoumání bylo zjištěno, že na vině byl router zákazníka, který byl schopen rozumně přenášet data jen do rychlosti cca 12 Mbit, pak již začala strmě stoupat odezva. Tento stav aplikace detekovala jako přetížení a snažila se řízeným shapingem zařídit přijatelnou odezvu. Bylo otestováno, jak by se aplikace chovala, kdyby mohla rychlost snížit až na takto nízkou rychlost a v takovém případě zareagovala správně a držela rychlost právě kolem 12 Mbit. Zákazník byl již o tomto problému s jeho routerem informován.

Průběžné testování zabralo mnoho hodin, ale díky němu byla získávána důležitá zpětná vazba na prováděné úpravy a bylo tak možné v jednotlivých částech vývoje lépe korigovat další směr. Nyní již aplikace u *ISP* ke spokojenosti běží.

Aby bylo možné přehledně sledovat aplikací prováděné úpravy, byl vytvořen další *Python* skript, který do *Round Robin* databáze zaznamenává nastavené rychlosti v průběhu času. Upraveno bylo též generování grafu odesílání a přijímání dat zákazníka a byly do něj přidány tyto statistiky. Příklad vygenerovaných grafů najdete na obrázku 12. Z grafu lze vyčíst, že

zákazníkovi cca v 15:30 výrazně vzrostla odezva, na což zareagovala aplikace snížením rychlosti. Jelikož zhoršení odezvy bylo velmi výrazné, snížila aplikace rychlost až na minimální mez 60 % maximální rychlosti. Odezva se výrazně zlepšila, stále ale ještě nad přijatelnou mez, ale aplikace již dále rychlost snížit nemohla. Zhruba od 17 hodin pak došlo ke zlepšení odezvy a aplikace postupně navýšila rychlost opět na maximum a zákazník tak opustil řízený shaping. Okolo 17:30 pak zákazník využíval téměř 100 % své maximální rychlosti, ale v tu dobu již bez výrazného vlivu na odezvu, z čehož je zřejmé, jak rychle se změnila pro něj dostupná rychlost.



Obrázek 12: Ukázka řízeného shapingu v praxi. Nahoře denní graf datového toku, dole graf mediánu odezvy. [autor]

Přestože je aplikace již v provozu, bylo by možné provést další úpravy, kterými by se mohla dále vylepšovat. Bylo by například možné zkrátit čas nutný k prvotní reakci na zhoršenou odezvu. Hraniční hodnota pro spuštění řízení shapingu u zákazníka je nyní 2,5 krát zhoršení odezvy oproti dennímu průměru – pokud by se reagovalo již na nižší hodnotu, byla by reakce rychlejší, ale přibylo by tak k řízení další množství zákazníků, kteří by ale, dle provedených měření, systém řízení z větší části rychle opouštěli. Takové množství řízení shapingu však aktuálně není vhodné realizovat z důvodu časové náročnosti změn shapingu. Možnosti by se ale mohly výrazně vylepšit, pokud by se změny shapingu spouštěly ve více vláknech, dle příslušnosti zákazníků k jednotlivým zařízením zodpovědných za shaping.

5.2 Síťové útoky

Detekce útoků z vnitřní sítě byla testována v reálné síti *ISP*. Při této detekci nedochází k automatické blokaci, ale jen k upozornění správců sítě o případném problému, tedy zde nevzniká výraznější problém v případě chybné detekce. Pokud správci sítě vyhodnotí upozornění jako nevhodné, může být detekce dále upravena, aby lépe odpovídalo jejich představám.

Pokud by detekce problém neodhalila, tak na tento nedostatek bude *ISP* pravděpodobně upozorněn protistranou. V takovém případě je nutné prověřit, proč problém nebyl detekován a nedostatek vhodně doplnit. Nejhorší případ by nastal v situaci, kdy by útok nebyl ani detekován, ani by na něj *ISP* nebyl upozorněn a protistrana by přistoupila rovnou k blokaci. Tato situace by ale pravděpodobně dříve či později byla odhalena, neboť by způsobovala problémy uživatelům. Tedy po jejich nahlášení by opět bylo nutné zajistit nápravu, aby se podobná situace dále neopakovala.

Detekce útoků z internetu probíhala taktéž v reálném prostředí *ISP*. Aby nedocházelo k nechtěným blokacím, prováděly se různé kontroly, ale i přes ně by však především v počátku mohla nastat nějaká neočekávaná situace. S takovou možností se počítalo a tak samotná blokace byla prováděna jen na firemní internetové přípojce a dále na přípojkách zaměstnanců *ISP*. Jelikož o nasazení této ochrany věděli, bylo jednodušší informovat o případných problémech. V této situaci bylo přikročeno i k blokaci nejen spojení iniciované útočníky směrem z internetu ale kompletní komunikace s IP útočníků, tedy i iniciované z vnitřní sítě. Takto sice blokace není plánována, ale při daném testování lépe odhalí problém. Žádné problémy se ale neobjevily a tak bude brzy možné otestovat blokaci i v další části sítě a postupně ji takto nasazovat v celé síti.

Detekce a blokace rozesílání nevyžádané pošty byla testována také v reálném prostředí *ISP*. Po úvodních testech byla nasazena několik týdnů v celé síti *ISP* jen v módu detekce a informační e-maily pro zákazníky byly směrovány na k tomu vytvořenou e-mailovou schránku *ISP*. Sledovalo se chování a domlouvalo se řešení s firemními zákazníky, kterých by se blokace týkala z důvodu odesílání e-mailů z jejich poštovních serverů ve více než 30 vláknech. Nyní je tato detekce i s blokací již několik měsíců spuštěna v celé síti a slouží svému účelu.

6 Shrnutí výsledků

Výsledkem první části této práce je funkční řízení shapingu provozu, které vychází z detekce přetížení bezdrátových spojů. Trvale monitoruje situaci v síti, detekuje snížení kapacity bezdrátových spojů a pomocí řízeného shapingu upravuje dle potřeby přidělené rychlosti uživatelů v rámci parametrů jejich tarifu. Situaci dále sleduje, přidělené rychlosti zvyšuje nebo snižuje dle aktuální kapacity spoje. V momentě vyřešení situace je řízení rychlosti daného uživatele ukončeno.

Toto řešení je nasazeno u *ISP*, kde zdárně pomáhá přetížení detekovat a řešit. Oproti porovnávaným existujícím řešením nepotřebuje množství nákladných senzorů a ani ke svému chodu nepotřebuje vytvářet mapu bezdrátových sítí. Jedná se o čistě softwarovou aplikaci, která vychází z měřené odezvy a statistik datových toků. Tyto statistiky jsou navíc pro *ISP* hodnotné i při řešení jiných problémů a pro správné dimenzování sítě.

Vytvořená aplikace pro *ISP* nepředstavuje žádné náklady navíc, ke svému chodu potřebuje jen popisované statistiky, informace o přidělených rychlostech zákazníkům a napojení na shaping. Tato aplikace je vhodná především pro poskytovatele internetu, ale s určitými úpravami by mohla být použita třeba i pro jednotlivé bezdrátové vysílače s mnoha uživateli, jako jsou například přístupové body na koncertech nebo festivalech.

Pro detekci síťových útoků bylo vytvořeno řešení, které sleduje kompletní příchozí i odchozí síťovou komunikaci v rámci celého *ISP*. Příchozí komunikaci sleduje na vstupních rozhraních jednotlivých konektivit, odchozí komunikaci pak co nejblíže zákazníkům. Na těchto datech detekuje útoky.

V případě detekce útoků z internetu zkoumá, zda by u detekovaných útočících IP nedošlo jejich blokací k narušení legitimní komunikace. Pokud ne, blokuje daný útok. Pokud by tak k narušení legitimní komunikace došlo, informuje obsluhu.

V případě detekce útoků z vnitřní sítě nedochází k automatické blokaci uživatele, ale je informována obsluha o problematické komunikaci. Obsluha pak může vyhodnotit situaci, informovat zákazníka, nebo přistoupit k blokaci.

Dále je vytvořeno řešení pro detekci a blokaci rozesílání spamu, které reaguje na navazování velkého množství spojení na poštovní servery a není tak omezeno na využití jen na kon-

krétním poštovním serveru. V případě detekce takového stavu dojde k blokaci odesílání a k informování zákazníka.

Všechny části detekce síťových útoků jsou nasazeny v reálném prostředí *ISP*, kde přináší svůj užitek. Oproti existujícím řešení nepotřebuje vytvořená detekce nákladné senzory, ale vyčítá informace o síťových tocích na již běžících zařízeních. Nespouští se až v případě závažnějšího problému, ale běží trvale. Funguje nezávisle na použitých routerech u zákazníků. Útoky jsou detekovány nejen z internetu, ale i z vnitřní sítě, a toto řešení tak dokáže odhalit i napadená zařízení uvnitř sítě *ISP*. Před samotnou blokací útočníků se provádí kontroly, aby tato blokace neměla vliv na legitimní provoz.

7 Závěry a doporučení

V této práci jsme se zabývali využitím automatizace v síti *ISP* pro zlepšení kvality služeb dodávaných zákazníkům. V první části této práce jsme se zabývali detekcí a eliminací přetížení rádiových spojů, v druhé pak detekcí síťových útoků. Představili jsme dané problémy, aktuální možnosti jejich řešení a vlastní aplikaci pro řešení těchto problémů.

Vytvořené řešení bylo testováno a následně nasazeno v reálné síti *ISP*. Nyní úspěšně slouží svému účelu, řeší nedostatky existujících řešení a není náročné finančně ani výkonově. Obě části je vhodné ještě dále u *ISP* sledovat a vylepšovat, aby dokázaly plně využít svůj potenciál.

V případě detekce a eliminace přetížení rádiových spojů by mohla být dále zlepšena rychlost prvotní reakce na vzniklé přetížení. Docílit by toho bylo možné užším propojením s aplikací na shaping, tak aby se shaping upravoval ve více vláknech na jednotlivých zařízeních zároveň. Díky tomu by bylo možné obsloužit větší množství úprav shapingu a mohlo by se tak reagovat již na menší odchylky detekované na bezdrátových spojích.

Detekci útoků z internetu by bylo možné podpořit spuštěním *honeypot*, díky tomu by mohly být zachyceny další útoky. Na detekci útoků z vnitřní sítě sítě by se mohlo při překročení mezních hodnot reagovat i automaticky blokací daného útoku, tedy například blokací komunikace na konkrétním portu.

Literatura

[1] Cisco IOS Netflow Data Sheet. [Online], [rev. 2004-09-17], [cit. 2021-12-13].
URL https://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/ios-netflow/product_data_sheet0900aecd80173f71.html

[2] Internet Control Message Protocol (ICMP) Parameters. [Online], [rev. 2020-09-25], [cit. 2022-03-13].
URL
https://www.iana.org/assignments/icmp-parameters/icmp-parameters.xhtml

[3] Internet Control Message Protocol version 6 (ICMPv6) Parameters. [Online], [rev. 2022-03-04], [cit. 2022-03-13].
URL https:
//www.iana.org/assignments/icmpv6-parameters/icmpv6-parameters.xhtml

[4] The Internet Organised Crime Threat Assessment (IOCTA) 2014. [Online], [cit. 2022-01-09].
URL https://www.europol.europa.eu/content/internet-organised-crime-threat-assesment-iocta

[5] The Internet Registry System. [Online], [cit. 2022-02-04].
URL https://www.ripe.net/participate/internet-governance/internet-technical-community/the-rir-system

[6] nfdump. [Online], [cit. 2021-12-14].
URL https://github.com/phaag/nfdump

[7] RTBH filtrování. [Online], [rev. 2016-12-15], [cit. 2022-03-05].
URL https://www.cesnet.cz/sluzby/rtbh/

[8] Sentinel:view. [Online], [cit. 2022-03-05].
URL https://view.sentinel.turris.cz/

[9] Turris Documentation. [Online], [cit. 2022-03-05].
URL https://docs.turris.cz/

[10] Všeobecné oprávnění č. VO-S/1/08.2020-9. [Online], [rev. 2020-08-18], [cit. 2020-12-27].
URL https://www.ctu.cz/sites/default/files/obsah/stranky/36864/soubory/vos1final.pdf

[11] Inline DDoS Protection versus Scrubbing Center Solutions. 2018, [Online], [cit. 2022-03-06].
URL https://www.allot.com/resources/SB-DDoS-Protection-inline-vs-scrubbing.pdf

[12] AGARWAL Sharad; DAWSON Travis; TRYFONAS Christos; aj.: DDoS Mitigation via Regional Cleaning Centers. 2004: str. 11.
URL https://www.senki.org/wp-content/uploads/2017/05/DDoS-Mitigation-via-Regional-Cleaning-Centers-RR04-ATL-013177-Sprint.pdf

[13] IEEE Mediterranean Electrotechnical Conference; Institute of Electrical and Electronics Engineers; Morocco Section (editoři): *2018 19th IEEE Mediterranean Electrotechnical Conference: 2-7 May 2018, Marrakech, Morocco*. 2018, ISBN 978-1-5386-3738-8, oCLC: 1182820672.

[14] JIRAN Petr: Evoluce RTBH v NIX.CZ. Červen 2017, [Online], [cit. 2022-03-11].
URL https://www.nic.cz/files/nic/IT_17/Prezentace/Petr_Jiran.pdf

[15] KOLOUCH Jan: *CyberCrime*. číslo 14. publikace in Edice CZ.NIC, Praha: CZ.NIC, z.s.p.o, první vydání, 2016, ISBN 978-80-88168-15-7.

[16] KOLOUCH Jan; BAŠTA Pavel; KROPÁČOVÁ Andrea; aj.: *CyberSecurity*. CZ.NIC, z.s.p.o, první vydání, 2019, ISBN 978-80-88168-31-7, oCLC: 1089706693.

[17] KUROSE James F; ROSS Keith W: *Počítačové sítě*. Brno: Computer Press, 2014, ISBN 978-80-251-3825-0, oCLC: 890260002.

[18] NISHIMAKI Satoru; YAMAMOTO Hiroshi; YAMAZAKI Katsuyuki: WiFi concierge at home network focusing on streaming traffic. *IEICE Communications Express*, ročník 4, č. 2, 2015: s. 67–72, ISSN 2187-0136, doi:10.1587/comex.4.67.
URL https://www.jstage.jst.go.jp/article/comex/4/2/4_67/_article

[19] PETRE Ionut; BONCEA Radu; RADULESCU Constanta Zoie; aj.: A Time-Series Database Analysis Based on a Multi-attribute Maturity Model. *Studies in Informatics and Control*, ročník 28, č. 2, Červenec 2019, ISSN 12201766, 1841429X, doi:10.24846/v28i2y201906.
URL https://doi.org/10.24846/v28i2y201906

[20] PUŽMANOVÁ Rita: *TCP/IP v kostce*. České Budějovice: Kopp, 2009, ISBN 978-80-7232-388-3, oCLC: 505914090.

[21] RADEMACHER Michael; JONAS Karl: Interference of simulated IEEE 802.11 links with directional antennas. In *2017 Wireless Days*, Porto, Portugal: IEEE, Březen 2017, ISBN 978-1-5090-5856-3, s. 27–32, doi:10.1109/WD.2017.7918110.
URL http://ieeexplore.ieee.org/document/7918110/

[22] RESPETO Jonathan: New DDoS Vector Observed in the Wild: WSD attacks hitting 35/Gbps. Srpen 2021.
URL https://www.akamai.com/blog/security/new-ddos-vector-observed-in-the-wild-wsd-attacks-hitting-35gbps

[23] SAMMOUR Ibrahim; CHALHOUB Gerard: Evaluation of Rate Adaptation Algorithms in IEEE 802.11 Networks. *Electronics*, ročník 9, č. 9, Září 2020: str. 1436, ISSN 2079-9292, doi:10.3390/electronics9091436.
URL https://www.mdpi.com/2079-9292/9/9/1436

[24] SINGH Karanbir; DHINDSA Kanwalvir Singh; BHUSHAN Bharat: Collaborative Agent-based Model for Distributed Defense against DDoS Attacks in ISP Networks. *International Journal of Security and Its Applications*, ročník 11, č. 8, Srpen 2017: s. 1–12, ISSN 17389976, 17389976, doi:10.14257/ijsia.2017.11.8.01.
URL http://article.nadiapub.com/IJSIA/vol11_no8/1.pdf

[25] SOSINSKY Barrie: *Mistrovství - počítačové sítě: [vše, co potřebujete vědět o správě sítí]*. Brno: Computer Press, 2010, ISBN 978-80-251-3363-7, oCLC: 697268026.

[26] TRAGOS Elias Z.; FRAGKIADAKIS Alexandros; ASKOXYLAKIS Ioannis; aj.: The impact of interference on the performance of a multi-path metropolitan wireless mesh network. In *2011 IEEE Symposium on Computers and Communications (ISCC)*,

Corfu, Greece: IEEE, Červen 2011, ISBN 978-1-4577-0680-6, s. 199–204, doi:10.1109/ISCC.2011.5983840.
URL http://ieeexplore.ieee.org/document/5983840/

[27] TYTGAT Lieven; YARON Opher; POLLIN Sofie; aj.: Analysis and Experimental Verification of Frequency-Based Interference Avoidance Mechanisms in IEEE 802.15.4. *IEEE/ACM Transactions on Networking*, ročník 23, č. 2, Duben 2015: s. 369–382, ISSN 1063-6692, 1558-2566, doi:10.1109/TNET.2014.2300114.
URL http://ieeexplore.ieee.org/document/6729100/

[28] WEI Yi-Hung; LENG Quan; CHEN Wei-Ju; aj.: Schedule Adaptation for Ensuring Reliability in RT-WiFi-Based Networked Embedded Systems. *ACM Transactions on Embedded Computing Systems*, ročník 17, č. 5, Listopad 2018: s. 1–23, ISSN 1539-9087, 1558-3465, doi:10.1145/3236011.
URL https://dl.acm.org/doi/10.1145/3236011

Seznam zkratek

ARP *Address Resolution Protocol*

CSV *Comma Separated Values*

CVE *Common Vulnerabilities and Exposures*

DDoS *Distributed Denial of Service*

DHCP *Dynamic Host Configuration Protocol*

DNS *Domain Name System*

DoS *Denial of Service*

DRDoS *Distributed Reflected Denial of Service*

FTP *File Transfer Protocol*

HaaS *Honeypot as a Service*

HTTP *Hypertext Transfer Protocol*

IANA *Internet Assigned Numbers Authority*

ICMP *Internet Control Message Protocol*

IDS *Intrusion Detection System* (z angličtiny „*systém detekce průniku*“)

IoT *Internet of Things* (z angličtiny „*internet věcí*“)

IPS *Intrusion Prevention System* (z angličtiny „*systém prevence průniku*“)

ISP *Internet Service Provider*

LAN *Local Area Network* (z angličtiny „*lokální síť*“)

LIR *Local Internet Registry*

MITM *Man in the Middle* (z angličtiny „*člověk uprostřed*“)

NAS *Network Attached Storage*

NAT *Network Address Translation* (z angličtiny „*překlad síťových adres*“)

NTP *Network Time Protocol*

QoS *Quality of Service*

RIR *Regional Internet Registry*

RRDTool *Round Robin Database Tool*

RSSI *Received Signal Strength Indication*

RTBH *Remotely Triggered Black Hole*

RTT *Round-Trip Time*

SINR *Signal to Interference and Noise Ratio*

SMTP *Simple Mail Transfer Protocol*

SSH *Secure Shell*

WiFi *Wireless Fidelity*

WMN *Wireless Mesh Network*

Přílohy

A Obsah elektronické přílohy

- `/`
  - `src` — zdrojové kódy
    - `elpr` — detekce a eliminace přetížení rádiových spojů
      - `elpr.py` — detekce a eliminace pomocí řízeného shapingu provozu
      - `stat.py` — statistiky eliminace přetížení
      - `README.md` — informace o účelu a použití
    - `desu` — detekce síťových útoků
      - `detekce.py` — detekce útoků z vnitřní sítě do internetu
      - `detekce_internet.py` — detekce útoků z internetu do vnitřní sítě
      - `blokace_internet.py` — blokace detekovaných útočníků
      - `README.md` — informace o účelu a použití
    - `spam` — detekce rozesílání nevyžádané pošty
      - `kontrola_spam.py` — kontrola blokace rozesílání pošty
      - `README.md` — informace o účelu a použití
    - `thesis` — zdrojová forma práce ve formátu LaTeX
  - `thesis.pdf` — text práce ve formátu PDF

B Odkazy na GIT repozitáře

Aplikace, které jsou výsledkem této práce, jsou dostupné na následujících odkazech:

- **Detekce a eliminace přetížení rádiových spojů**
  https://github.com/vrbaji1/elpr
- **Detekce a eliminace síťových útoků**
  https://github.com/vrbaji1/desu
- **Detekce rozesílání nevyžádané pošty**
  https://github.com/vrbaji1/spam

C Příklad stížnosti na útok

Následuje příklad stížnosti na útok. IP adresy ze kterých útok pocházel, IP adresy cíle i emaily jsou upraveny, aby neukazovaly na konkrétní organizace. Ze stejných důvodů byl email také zkrácen.

Subject: brute-force from your network / domain (198.51.100.134)
Date: Mon, 30 Aug 2021 19:36:33 +0200
From: abuse+noreply@example.com
Reply-To: noc@example.com
To: abuse@firma.example

An attempt to brute-force account passwords over SSH/FTP by a machine in your domain or in your network has been detected. Attached are the host who attacks and time / date of activity. Please take the necessary action(s) to stop this activity immediately. If you have any questions please reply to this email.

Host of attacker: 198.51.100.134 => =>
Responsible email contacts: abuse@firma.example
Attacked hosts in our Network: 203.0.113.36, 203.0.113.11, 203.0.113.34, 203.0.113.153, 203.0.113.86

Logfile entries (time is CE(S)T):
Mon Aug 30 19:36:20 2021: user: root service: ssh target: 203.0.113.34 source: 198.51.100.134
Mon Aug 30 19:36:10 2021: user: root service: ssh target: 203.0.113.34 source: 198.51.100.134
Mon Aug 30 19:36:10 2021: user: root service: ssh target: 203.0.113.34 source: 198.51.100.134
Mon Aug 30 19:36:10 2021: user: root service: ssh target: 203.0.113.34 source: 198.51.100.134
Mon Aug 30 19:36:10 2021: user: root service: ssh target: 203.0.113.34 source: 198.51.100.134
Mon Aug 30 19:36:00 2021: user: ubnt service: ssh target: 203.0.113.34 source: 198.51.100.134
Mon Aug 30 19:36:00 2021: user: root service: ssh target: 203.0.113.34 source: 198.51.100.134
Mon Aug 30 19:36:00 2021: user: root service: ssh target: 203.0.113.34 source: 198.51.100.134
Mon Aug 30 19:33:34 2021: user: root service: ssh target: 203.0.113.11 source: 198.51.100.134

Mon Aug 30 19:33:34 2021: user: ubnt service: ssh target: 203.0.113.11 source: 198.51.100.134
Mon Aug 30 19:33:34 2021: user: root service: ssh target: 203.0.113.11 source: 198.51.100.134
Mon Aug 30 19:33:24 2021: user: root service: ssh target: 203.0.113.11 source: 198.51.100.134
Fri Aug 27 23:31:03 2021: user: root service: ssh target: 203.0.113.153 source: 198.51.100.134
Fri Aug 27 23:30:53 2021: user: root service: ssh target: 203.0.113.153 source: 198.51.100.134
Fri Aug 27 23:30:53 2021: user: root service: ssh target: 203.0.113.153 source: 198.51.100.134
Fri Aug 27 23:30:43 2021: user: ubnt service: ssh target: 203.0.113.153 source: 198.51.100.134
Fri Aug 27 23:30:43 2021: user: root service: ssh target: 203.0.113.153 source: 198.51.100.134
Fri Aug 27 23:30:43 2021: user: root service: ssh target: 203.0.113.153 source: 198.51.100.134
Thu Aug 26 04:37:04 2021: user: root service: ssh target: 203.0.113.86 source: 198.51.100.134
Thu Aug 26 04:36:54 2021: user: root service: ssh target: 203.0.113.86 source: 198.51.100.134
Thu Aug 26 04:36:54 2021: user: ubnt service: ssh target: 203.0.113.86 source: 198.51.100.134
Thu Aug 26 04:36:44 2021: user: root service: ssh target: 203.0.113.86 source: 198.51.100.134
Thu Aug 26 04:36:44 2021: user: root service: ssh target: 203.0.113.86 source: 198.51.100.134
Thu Aug 26 04:36:14 2021: user: root service: ssh target: 203.0.113.36 source: 198.51.100.134
Thu Aug 26 04:36:04 2021: user: root service: ssh target: 203.0.113.36 source: 198.51.100.134
Thu Aug 26 04:36:04 2021: user: root service: ssh target: 203.0.113.36 source: 198.51.100.134
Thu Aug 26 04:36:04 2021: user: ubnt service: ssh target: 203.0.113.36 source: 198.51.100.134
Thu Aug 26 04:35:54 2021: user: root service: ssh target: 203.0.113.36 source: 198.51.100.134
Thu Aug 26 04:35:54 2021: user: root service: ssh target: 203.0.113.36 source: 198.51.100.134
...

Regards,
Example Team

If you wish to change or report a non-working abuse contact address.
please contact the appropriate RIR responsible for managing the underlying data.

Univerzita Hradec Králové
Fakulta informatiky a managementu

Zadání diplomové práce

Autor: **Bc. Jindřich Vrba, DiS.**

Studium: I1900555

Studijní program: N1802 Aplikovaná informatika

Studijní obor: Aplikovaná informatika

Název diplomové práce: **Automatizace v síti poskytovatele internetu**

Název diplomové práce AJ: Automation in Internet Service Provider's network

Cíl, metody, literatura, předpoklady:

Cíl: Navrhnout a implementovat systém, který by zajistil automatizaci dílčích procesů v síti poskytovatele internetu, vedoucích k vyšší stabilitě sítě a lepší kvalitě služeb. Konkrétně půjde o detekci a eliminaci přetížení rádiových spojů pomocí řízeného shapingu provozu a o detekci a eliminaci nebezpečného provozu (síťových útoků).

Osnova:

1. Úvod
2. Cíl práce
3. Analýza problému
4. Návrh a implementace řešení
5. Zhodnocení výsledků a doporučení
6. Závěr

- Všeobecné oprávnění č. VO-S/1/08.2020-9. Aug. 2020. Available from: https://www.ctu.cz/sites/default/files/obsah/stranky/36864/soubory/vos1final.pdf
- Petre, I.; Boncea, R.; Radulescu, C. Z.; et al. A Time-Series Database Analysis Based on a Multi-attribute Maturity Model. Studies in Informatics and Control, volume 28, no. 2, July 2019, ISSN 12201766, 1841429X, doi:10.24846/v28i2y201906. Available from: https://doi.org/10.24846/v28i2y201906

Garantující pracoviště: Katedra informatiky a kvantitativních metod, Fakulta informatiky a managementu

Vedoucí práce: Ing. Pavel Kříž, Ph.D.

Datum zadání závěrečné práce: 15.10.2021