VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2019 Jakub Rajj

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

FILTRACE DISTRIBUOVANÝCH ÚTOKŮ NA ODEPŘENÍ SLUŽEB POMOCÍ SÍŤOVÝCH PRVKŮ MIKROTIK
DISTRIBUTED DENIAL OF SERVICE FILTERING BASED ON MIKROTIK NETWORK DEVICES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **Jakub Rajj**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. Tomáš Gerlich**
SUPERVISOR

BRNO 2019

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní obor **Teleinformatika**
Ústav telekomunikací

Student: Jakub Rajj ***ID:*** 186451
Ročník: 3 ***Akademický rok:*** 2018/19

NÁZEV TÉMATU:

Filtrace distribuovaných útoků na odepření služeb pomocí síťových prvků Mikrotik

POKYNY PRO VYPRACOVÁNÍ:

Bakalářská práce se zabývá filtrací útoků na odepření služeb s využitím síťových prvků Mikrotik. Seznámit se s experimentálním pracovištěm využívající IDPS Suricata a síťové prvky Mikrotik.
Teoretická část práce bude rozebírat útoků na odepření služeb, detekční a filtrační mechanizmy. Praktická část se zaměří na realizaci filtrace útoků (nejméně ICMP flood, RST flood) a zajištění komunikace mezi IDPS Suricata a přepínačem Mikrotik. Cílem práce bude odfiltrování útoku a zachování legitimního provozu a ověřit výkonnost filtrace. V semestrální práci nastudovat detekci DDoS útoků RST flood a ICMP flood pomocí nástroje Suricata. Realizovat komunikaci mezi IDPS Suricata a sítovým prvkem Mikrotik.

DOPORUČENÁ LITERATURA:

[1] MIRKOVIC, Jelena, Peter REIHER a Kotagiri RAMAMOHANARAO. A taxonomy of DDoS attack and DDoS defense mechanisms. ACM SIGCOMM Computer Communication Review. 2004, 34(2), 3-es. DOI: 10.1145/997150.997156. ISSN 01464833.

[2] PENG, Tao, Christopher LECKIE a Kotagiri RAMAMOHANARAO. Survey of network-based defense mechanisms countering the DoS and DDoS problems. ACM Computing Surveys. 2007, 39(1), 3-es. DOI: 10.1145/1216370.1216373. ISSN 03600300

Termín zadání: 1.2.2019 ***Termín odevzdání:*** 27.5.2019

Vedoucí práce: Ing. Tomáš Gerlich
Konzultant:

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

ABSTRAKT

Bakalářská práce je zaměřena na problematiku útoků cílených na odepření služeb. Tyto typy útoku jsou stále aktuální, ale v současné době se využívají spíše ve své distribuované podobě. V teoretické části práce jsou popsány základní mechanizmy útoků, možné dělení útoků a jsou zde uvedeny a popsány nejznámější typy útoků. Teoretická část rovněž obsahuje popis způsobů reakce na tyto útoky. Zde jsou uvedeny čtyři základní metody pro boj proti útokům (prevence, detekce, identifikace a reakce) a dále jsou zde popsány systémy na detekci a prevenci průniku (IDPS). Praktická část se zabývá propojením systému detekce průniku se směrovačem, za účelem detekce a následné filtrace na zmíněném směrovači.

KLÍČOVÁ SLOVA

DDoS, DoS, IDS, MikroTik, Suricata

ABSTRACT

The bachelor thesis is focused on the issue of denial of service attacks. These types of attacks are still up to date, but are currently being used in their distributed form. The theoretical part of the thesis describes the basic mechanisms of attacks and the division of attacks. Also there are described the most famous type of attacks. The theoretical part also includes a description of how to respond to these attacks. Four basick methods are described (prevention, detection, identification and response), as well as intrusion detection and prevention systems (IDPS). The practical part deals with connection of intrusion detection system and router, for detection and filtration on said router.

KEYWORDS

DDoS, DoS, IDS, MikroTik, Suricata

Bibliografická citace:

RAJJ, J. Filtrace distribuovaných útoků na odepření služeb pomocí síťových prvků Mikrotik. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2019. 59 s. Vedoucí bakalářské práce Ing. Tomáš Gerlich.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Filtrace DDoS pomocí síťových prvků Mikrotik“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

V Brně dne ....

(podpis autora)

PODĚKOVÁNÍ

Chtěl bych poděkovat vedoucímu práce panu Ing. Tomáši Gerlichovi za odborné vedení, trpělivost, cenné rady a ochotu, kterou mi při realizaci bakalářské práce věnoval.

V Brně dne ....

(podpis autora)

Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic

http://www.six.feec.vutbr.cz

PODĚKOVÁNÍ

Výzkum popsaný v této bakalářské práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno ....

podpis autora

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

OP Výzkum a vývoj
pro inovace

OBSAH

Úvod **13**

1 Teoretický rozbor **14**

1.1 IP spoofing ... 18

1.2 Typy útoku ... 20

1.2.1 Fyzická vrstva ... 21

1.2.2 Linková vrstva ... 21

1.2.3 Síťová vrstva ... 22

1.2.4 Transportní vrstva ... 22

1.2.5 Relační vrstva ... 22

1.2.6 Prezentační vrstva ... 22

1.2.7 Aplikační vrstva ... 22

1.3 Útoky využívající jiných zařízení ... 23

1.3.1 Zesilující útok ... 23

1.3.2 Odrazený útok ... 23

1.4 Často se vyskytující nebo známé typy útoků ... 25

2 Obrana proti DOS a DDOS **30**

2.1 Prevence ... 30

2.1.1 Filtrování na vstupu a výstupu ... 30

2.1.2 Filtrování paketů na mezilehlých směrovačích ... 31

2.1.3 Filtrování s potvrzováním zdrojové adresy ... 31

2.1.4 Autentizované adresy ... 31

2.2 Detekce ... 32

2.2.1 Intrusion Detection Systems ... 33

2.2.2 Intrusion Prevention System ... 33

2.3 Identifikace zdroje útoku ... 33

2.4 Reakce ... 34

3 Mikrotik **35**

3.1 Komunikace se směrovačem ... 35

3.2 Konzole ... 35

3.3 Winbox ... 36

3.4 WebFig 37

4 Vlastní experimentální pracoviště a suricata 39

4.1 NetScan Tools 39

4.2 Trafgen 42

4.3 Suricata 45

5 Realizace na experimentálním pracovišti VUT 46

6 Závěr 51

Seznam symbolů, veličin a zkratek 56

Seznam příloh 58

SEZNAM OBRÁZKŮ

Obr. 1.1 Ukázka DoS 15
Obr. 1.2 Ukázka DDoS a botnetu 16
Obr. 1.3 Centralizovaná komunikace v botnetu 17
Obr. 1.4 Decentralizovaná komunikace v botnetu 18
Obr. 1.5 IP spoofing 19
Obr. 1.6 Model OSI a na typy útoků 21
Obr. 1.7 Zesilující DDoS 23
Obr. 1.8 Reflektivní útok 24
Obr. 1.9 DRDoS 25
Obr. 2.1 Vstupní, výstupní filtr 31
Obr. 3.1 Konzole 36
Obr. 3.2 Winbox 37
Obr. 3.3 WebFig 38
Obr. 4.1 LAN schéma 39
Obr. 4.2 Rozhraní NetScan Tool 40
Obr. 4.3 Vygenerovaný datový tok programem NetScan Tools 41
Obr. 4.4 Ping u prvního útoku 41
Obr. 4.5 Wireshark při prvním útoku 42
Obr. 4.6 UDP paket 43
Obr. 4.7 Trafgen 43
Obr. 4.8 Vygenerovaný datový tok nástrojem Trafgen 44
Obr. 4.9 Ping při druhém útoku 44
Obr. 4.10 Wireshark při druhém útoku 45
Obr. 4.11 Statistika zachycených paketů ze souboru stats.log 45
Obr. 5.1 Zjednodušené schéma pracoviště 46
Obr. 5.2 Okno softwaru 47
Obr. 5.3 Okno pro nastavení parametrů obsahu 47
Obr. 5.4 Výňatek z logu 48
Obr. 5.5 Expect script 48
Obr. 5.6 ICMP flood 49
Obr. 5.7 RST flood s větším datovým tokem 49

Obr. 5.8 Script 49
Obr. 5.9 Seznam adres 50

SEZNAM TABULEK

Tabulka 4.1 Vytížení CPU při přihlášení……39

ÚVOD

V informatice se již od počátků můžeme setkat se škodlivým kódem. Může se jednat o počítačové viry jako například červy, trojské koně a jiné. Také se můžeme setkat s internetovými útoky. Příkladem jsou právě útoky na odepření služeb a distribuované útoky na odepření služeb.

S rozvojem počítačových sítí se tyto útoky vyskytují čím dál tím častěji. Je jich celá řada a využívají nejrůznějších systémových slabin, bezpečnostních trhlin a neopatrnosti uživatelů systémů. Cílem útoků k odepření služeb není obohacení útočníka, nebo krádež osobních informací oběti. Je jím omezení dostupnosti služby legitimním uživatelům, a to buď vyčerpáním výpočetní kapacity, operační paměti či síťového pásma. Během útoku jsou služby, na které je útok cílen, nedostupné a oběť tak může přijít i o značný finanční zisk, který by získala za normálního fungování služeb. Motivem k tomuto typu útoku bývají osobní důvody, politické důvody, konkurenční boje, nebo získání prestiže v hackerské komunitě [1].

Ochrana před útoky na odepření služeb a distribuovanými útoky na odepření služeb není stoprocentní. Nejdůležitějším bodem ochrany je prevence, která modifikuje stávající protokoly a systém, aby zabránila útoku před tím, než způsobí závažnější škody. Jestliže již útok probíhá, je potřeba co nejdříve detekovat zdroj útoku, určit, o jaký útok se jedná a podniknout kroky k zrušení útoku nebo aspoň umírnění následků.

První kapitola se zabývá útoky na odepření služeb. Jsou zde popsány základní pojmy, cíle útoků, podvrhování zdrojových IP adres, typy jednotlivý útoků. V závěru kapitoly jsou vybrány a popsány známé, nebo často se vyskytující útoky.

Druhá kapitola se zabývá možnými návrhy ochrany proti útokům na odepření služeb. Důležitá je především prevence. Dále je důležitá detekce probíhajícího útoku, aby se co nejdříve zjistilo, že probíhá útok a na jaké zařízení útočí. Po detekci následuje identifikace typu útoku a od ní odvozená reakce na daný útok.

Ve třetí kapitole je stručně popsána společnost MikroTik, její zařízení a operační systém používaný v těchto zařízeních. Dále je zde popis způsobu komunikace se zařízeními od MikroTiku a porovnání spotřebovávaného výkonu při přihlašování pomocí zmíněných metod komunikace.

Ve čtvrté kapitole jsou popsány experimenty, prováděné s různými generátory na zapůjčeném zařízení, experimenty prováděné na laboratorním pracovišti a vytvořený script, pro filtraci útoků.

Závěrečná pátá kapitola popisuje experimentální pracoviště VUT a dále skript, který byl vytvořen pro filtraci útoku.

1 TEORETICKÝ ROZBOR

Zkratka DoS (Denial of Services) v překladu znamená odepření služeb. Jedná se o síťový útok, sloužící k znepřístupnění služby pro ostatní uživatele. To je dosaženo generováním nadměrného množství požadavků, které se odesílají například na server. Server požadavky přijímá a podle protokolů se snaží jednotlivé požadavky zpracovávat. Množství požadavků, které generuje útočník (amatérský hacker, profesionál, nebo organizovaná skupina hackerů), je obrovské a na straně serveru dojde dříve či později k vyčerpání zdrojů. Těmito zdroji jsou: výpočetní výkon, operační paměť a šířka pásma. Jakmile je některý z těchto zdrojů úplně vytížen, server už nemůže zpracovávat další požadavky a začne je zahazovat. Aby byl útok úspěšný a došlo k vyřazení služby z provozu, musí být útočník schopen vyčerpat zdroje za použití svých prostředků, nebo prostředků jiných zařízení. Cílů útoku [1] může být hned několik. Asi nejčastějším cílem je aplikace pracující na serveru (např. emailová služba, autentizační služba). Po zahlcení aplikace zůstávají ostatní aplikace dostupné. Další cílem bývá konečná stanice, kterou se útočník snaží odříznout od síťové komunikace. Možným cílem jsou i jednotlivá zařízeni síťové infrastruktury, poskytující služby ostatním stanicím. V neposlední řadě bývá cílem i samotná infrastruktura. Útočník si za cíl vybere některé důležité zařízení, sloužící ke směrování paketů, a provede útok, který může ochromit část sítě.

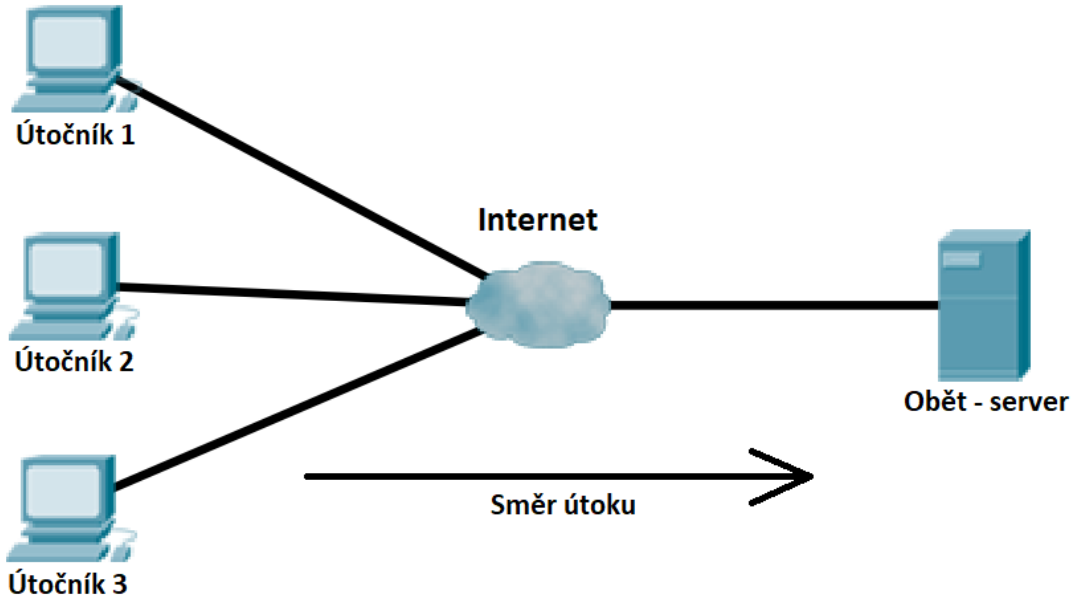
V počátcích útoků na odepření služeb byly útoky prováděny pouze z jednoho počítače, nebo z více počítačů v případě organizovaného útoku, kdy se domluvilo několik útočníků na datu a čase, kdy spustí útok současně. Toto bylo klasické pojetí DoS. Podtypem DoS je DDoS (Distributed Denial of Services) [2], co znamená distribuované odepření služeb. Zde se liší počet zdrojů, které vedou útok. Útočníci využívají tzv. botnety [3], tvořené velkým množstvím počítačů. Počítače v botnetu se nazývají zombie nebo bot.

Za to, že jsou útoky DoS a DDoS pořád v oblibě, mohou vlastnosti internetových sítí [2]. Důležitou vlastností sítí je pokus o co nejrychlejší přenos paketů na distribuční a páteřní vrstvě. Zařízení v páteřní a distribuční vrstvě poskytují všechny své zdroje na směrování paketů, ale úplně postrádají nástroje k zabezpečení sítě (např. autentizace uživatelů) To má za následek, že zabezpečení proti (D)DoS musí implementovat koncové sítě. Míra zabezpečení se v různých sítích může lišit, což může vést k potencionálním útokům. Charakteristickou vlastností internetu je i sdílení prostředků. To umožnuje rušení služeb (úmyslné ale i neúmyslné) ostatními uživateli (vyčerpání šířky pásma). Další vlastností sítí je přenos paketu po různých cestách, kvůli čemuž není vždy snadné vysledovat zdroj útoku. Útokům rovněž napomáhá využívaní techniky IP spoofingu k zamaskování zdroje útoku. Také využívání různých síťových zařízení k zesíleni nebo odrazu útočníkova provozu přispívá k hojnému využívání (D)DoS útoků.

Služba, na kterou je útočeno, může být ovlivněna dvěma způsoby [1]. Prvním z nich je úplné rušní služby, kdy je útokem úplně zabráněno využívat službu ostatními účastníky. Z tohoto stavu se služba může vzpamatovat, pokud je to možné. Pokud ne, musí se restartovat ručně uživatelem. Důležité je, aby byla funkce obnovena až po skončení útoku, nebo po provedení opatření k zabránění dalšího vyčerpání zdrojů. Jestliže došlo k fyzickému poškození HW, jedinou možností je oprava, nebo nákup nového HW. Druhým vlivem je blokování zdrojů služby. Útočník využívá velkého množství zdrojů

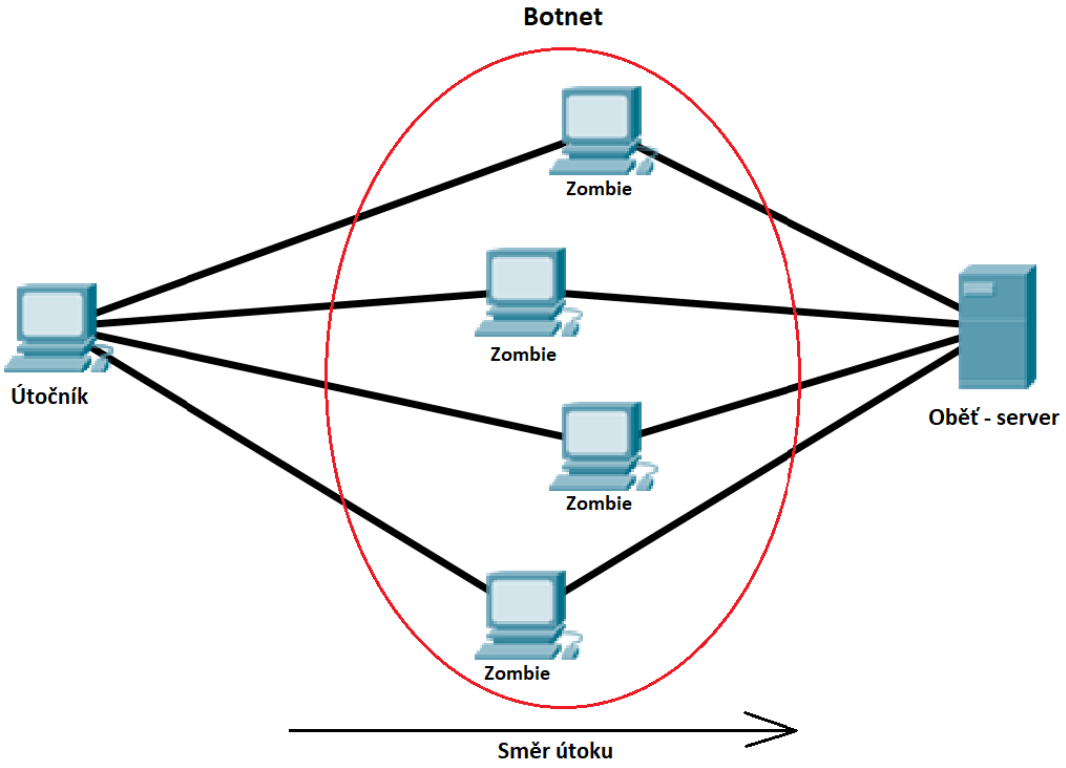
služby, a tak zhoršuje její rychlost a celkovou funkčnost. Druhá možnost je daleko hůře detekovatelná.

Datový tok útoku může mít různé charakteristiky [4]. Konstantní datový tok je první možností. Datový tok nekolísá a je víceméně konstantní. Využívá se jen nejmenší potřebná velikost datového toku, která stačí na odstavení požadované služby. Tyto datové toky jsou přesto objemné a snadno detekovatelné. Také se používají útoky s proměnlivým datovým tokem. Může se jednat o postupně narůstající datový tok, který dlouhodobě snižuje kvalitu služby. Nebo se používá kolísající datový tok, snažící se navodit pocit nahodilosti požadavků od legitimních uživatelů.



Obr. 1.1 Ukázka DoS

Na obrázku 1.1 je vidět útok DoS. Nejedná se o DDoS, protože jsou zde tři různí útočníci, kteří spolupracují, aby rychleji vyčerpali zdroje oběti útoku.

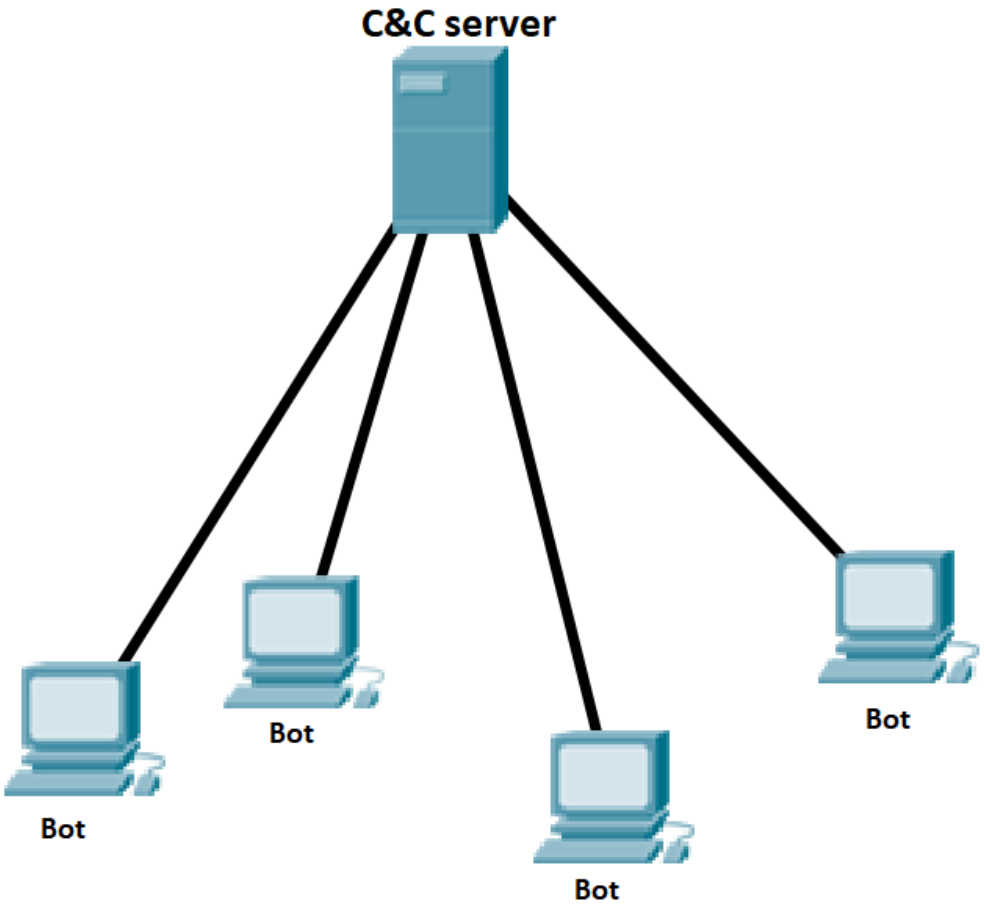


Obr. 1.2 Ukázka DDoS a botnetu

Na obrázku 1.2 můžeme vidět příklad DDoS útoku. Útočník vytvořil botnet ze čtyř počítačů (v praxi obsahují botnety daleko větší množství počítačů v řádu desítek až stovek tisíc) a ten využívá k odesílaní požadavků oběti.

Útočník může využívat různých metod [4] k rozšiřování botnetu o další boty. Jednou z nich jsou tzv. exploity [5]. Exploity využívají zranitelností operačního systému, používaného SW, nebo systémových knihoven k ovládnutí PC. Ochranou proti těmto zranitelnostem jsou záplaty (patche). Exploity nemusí být vždy použity se zlým úmyslem. Administrátoři si pomocí nich mohou testovat zranitelnost vlastních sítí, nebo stanic. Hlavním rozdílem exploitů je to, jestli hledají zranitelné počítače manuálně nebo automaticky. V případě manuálního hledaní zkoumá útočník potenciální slabiny náhodných, nebo vytipovaných PC, k čemuž využívá právě exploitů a svých znalostí v dané problematice. Druhou možností je automatické hledáních možných zranitelných zařízení. V této možnosti útočníci využívají skripty, které jsou napsány tak, aby prohledali určitou skupinu stanic. Útočník může využít několik technik k hledaní zranitelných zařízení. Nejjednodušší technikou je náhodné prohledávání IP adres. Dále může hledat zranitelné zařízení podle předem sestaveného listu potencionálních nových botů, nebo je může hledat podle určitých společných parametrů, jako je geografická lokace nebo logické umístění v topologii. Toto byly přímé metody. Využívají se rovněž nepřímé metody, kdy se očekává, že si oběť stáhne škodlivý kód sama kliknutím na pochybnou stránku nebo stažením přílohy z emailu. Po nalezení bezpečnostní slabiny přenese na cílový počítač škodlivý kód a získá nad ním kontrolu. Cílem tohoto kódu není vyřadit počítač z provozu, jak to dělají jiné viry, nýbrž zůstává co nejdéle skrytý, dokud útočník nedosáhne požadované velikosti botnetu a nespustí útok.

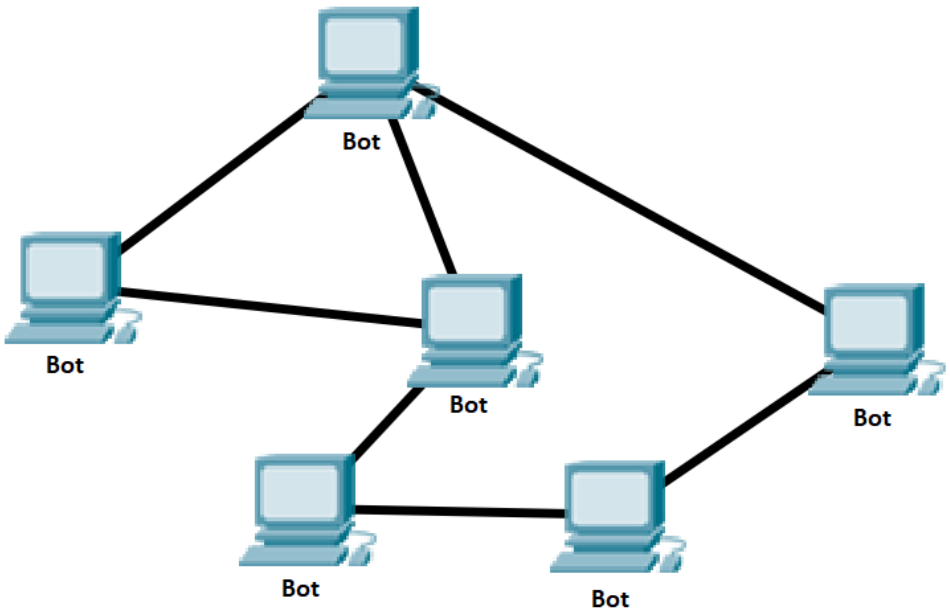
Po získání kontroly nad stanicí se tato stanice (již bot) snaží propojit na C&C (command-and-control) server. Z tohoto serveru očekává příkazy přicházející po C&C kanálu, a tak je realizována komunikace. Komunikace mezi C&C serverem a boty může být realizována několika způsoby [6][2]. Jednou z možností komunikace botů se serverem je pomocí telnetu. Dalším dříve velmi využívanou možností komunikace botů s C&C serverem je přes kanály IRC (Internet Realy Chat). K těmto kanálům se stanice připojí, jakmile jsou ovládnuty a naslouchají dalším příkazům od C&C serveru. Rovněž mohou být přes IRC kanál spravovány. Nevýhodou této metody je, že IRC provoz může být snadno identifikován, nalezen zdroj IRC provozu (IRC server) a ten může být vypnut. Další možností je využití domén pro vytvoření větších botnetů. Příkladem může být doména webové stránky, které obsahují příkazy pro kontrolu botů. Výhodou je, že komunikace se serverem se tváří jako legitimní HTTP provoz. Nevýhodou je potřeba velké šířky pásma pro komunikaci v rozsáhlých botnetech. Na obrázku 1.3 můžeme vidět komunikaci v botnetu s centrálním prvkem (C&C serverem).



Obr. 1.3 Centralizovaná komunikace v botnetu

Poslední možností je decentralizovaný přístup komunikace v botnetu: peer-to-peer (klient-klient). Boti komunikují mezi sebou, aby se odstranil slabý článek v centralizovaném modelu, kterým je právě C&C server. V tomto typu komunikace se příkazy od jiných stanic velice často šifrují, aby mohl příkazy posílat jen vlastník

šifrovacího klíče (privátní klíč). Na obrázku 1.4 můžeme vidět decentralizovanou topologii botnetu. Chybí zde server a místo toho si jednotliví boti rozesílají příkazy mezi sebou.



Obr. 1.4 Decentralizovaná komunikace v botnetu

V botnetu se mohou vyskytovat i jiná zařízení než PC, která jsou využívána k útokům. Příkladem je botnet Mirai [7], který umí ovládnou zařízeni s unixovým prostředím: IP kamery, a domácí routery. K získávání kontroly nad zařízeními využívá testování defaultních hesel, které administrátor nezměnil.

Dalším vhodným zařízením, nad kterým lze získat kontrolu a které lze využít k útoku je mobilní telefon. Oproti PC jsou mobilní telefony téměř neustále spuštěny a většinu času připojeny do sítě. Mobilních telefonů využíval botnet WireX [9], který získával kontrolu nad mobily při nainstalování podvodné aplikace.

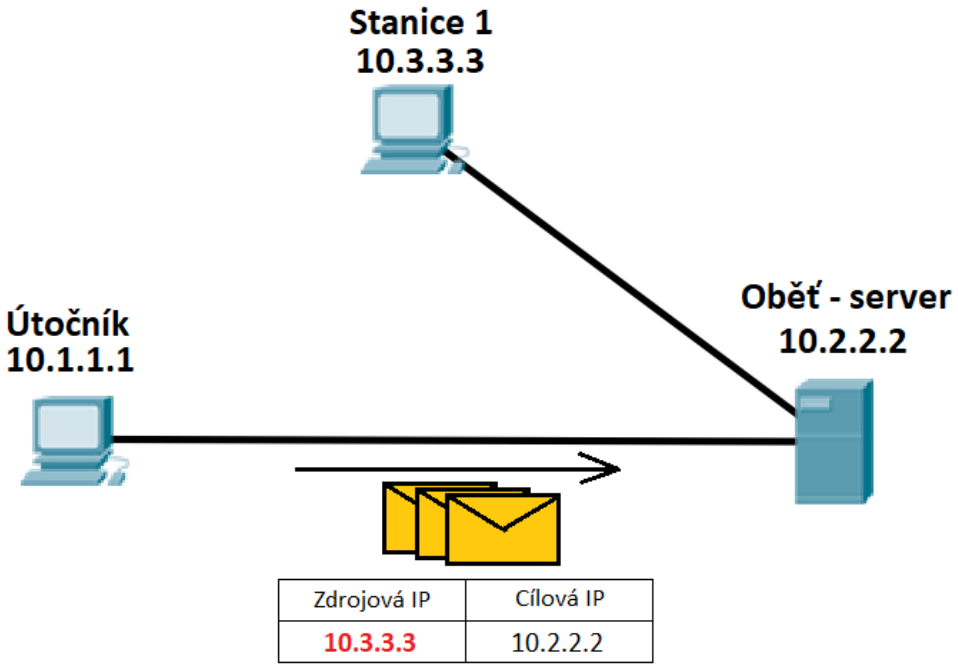
Pokud dojde k výpadku služby, nemusí se hned jednat o cílený útok. Může se jednat o úkaz nazývající se „Flash Crowds“ [8]. Jedná se o nadměrný provoz generovaný velkým množstvím přístupů. Vzniká například při významných sportovních událostech, kdy jsou některé stránky naprosto zahlceny. Od DDoS lze odlišit velkým, ale kolísavým datovým tokem.

1.1 IP spoofing

IP spoofing [1] je technika, kterou využívají útočníci k zamaskování zdroje útoku. V podstatě se jedná o záměnu zdrojové IP adresy paketů za jinou. Kvůli vlastnostem internetu není sledována cesta mezi zdrojem a cílem paketů, ani není stanovena jediná cesta na celou dobu komunikace, jako tomu je u komutace okruhů. V DoS se jednalo

o základní techniku pro ztížení zpětného dohledání pachatele. V DDoS útocích je IP spoofing rovněž důležitý. Pro zaměňování zdrojových adres se využívá hned několik technik.

První z nich je náhodné zaměňování. Útočník prostě vygeneruje náhodnou 32bitovou IP adresu a tu pak využívá jakou zdrojovou adresu v paketech. Nevýhodou této techniky je, že některé obranné mechanizmy, jako jsou filtry, umí detekovat pakety se zdrojovou IP adresou, která by se do sítě neměla vůbec dostat, protože není v očekávaném rozsahu IP, které do sítě obvykle přicházejí. K obejití této nevýhody a překonání filtrů lze využít jiných technik záměny IP. Jednou z nich je záměna podle používaných rozsahů v jednotlivých sítích. Provoz od jednotlivých botů se poté tváří, jako by skutečně pocházel z dané sítě, filtry nedetekují útok a zároveň nedojde k odhalení bota díky zaměněné IP. Také je možno přidělovat IP z nějakého seznamu. Poslední technika [2] vlastně vůbec nevyužívá záměnu IP. Botům jsou ponechány jejich IP a odesílají klasicky požadavky. Takto není možno rozeznat, jestli provoz pochází od uživatele kompromitovaného PC nebo jej vygeneroval útočník. V DDoS s botnetem obsahujícím desítky nebo stovky tisíc botů již útočník nepotřebuje skrývat jejich přítomnost a malý tok, spojený dohromady může vyřadit službu.



Obr. 1.5 IP spoofing

Na obrázku 1.5 můžeme vidět jednoduchou topologii obsahující pouze tři prvky. Útočník s IP 10.1.1.1, server s IP 10.2.2.2 a náhodná stanice 1 s IP 10.3.3.3, která se na útoku vůbec nepodílí. Její IP adresou je nahrazena skutečná adresa útočníka. Při provádění útoku si server myslí, že zdrojem datového provozu je právě stanice 1. Takto je možné vytvořit nekonečně zacyklené dotazy s minimálním vynaložením vlastních zdrojů.

1.2 Typy útoku

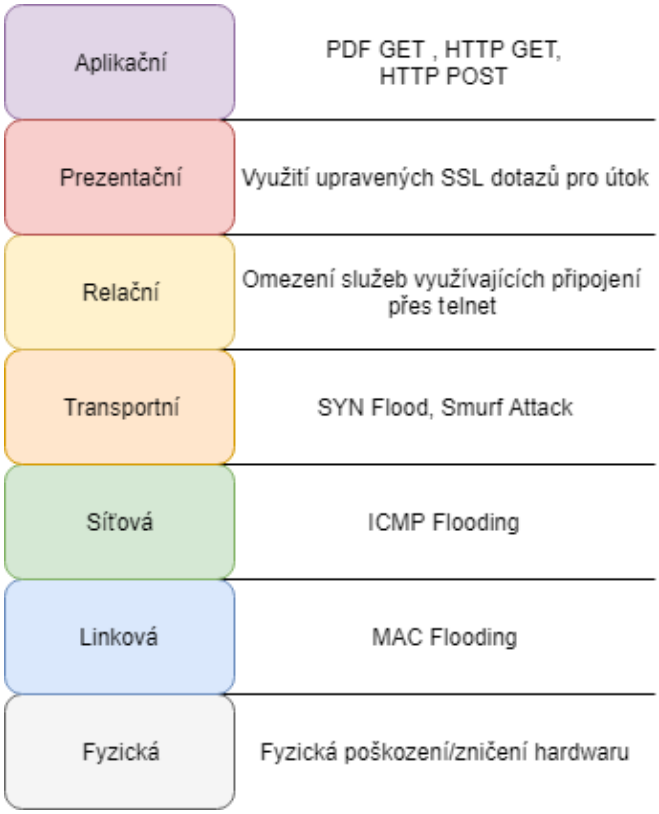
Útoky je možno rozlišovat [2] podle různých parametrů: podle účelu útoku (vyčerpání zdrojů, nebo šířky pásma), podle způsobu útoku (záplavově nebo využíváním slabin), podle toho, na jakou vrstvu referenčního modelu je útok cílen.

Podle účelu útoku lze rozlišit, jestli slouží k vyčerpání zdrojů oběti (RAM, CPU), nebo k vyčerpání šířky pásma. Při vyčerpání zdrojů oběti dochází k zahazování požadavků, které nemůže oběť zpracovat. Zároveň se snaží upozornit ostatní, aby redukovali množství dotazů, které oběti posílají. Normální uživatelé tuto žádost vyslyší, ale útočník začne posílat ještě více žádostí. U útoku cíleného na vyčerpání šířky pásma většinou převažuje množství požadavků útočníka nad ostatním provozem. Požadavky zahltí celou linku a opět dojde k zahazování. Pokud je některý z těchto útoků cílený na server poskytující služby ostatním stanicím nebo serverům, a tento útok se zdaří, poškozený bude nejen samotný server ale i ostatní účastníci, spoléhající na služby napadeného serveru.

Při dělení útoků podle způsobu útoku [10] rozlišujeme útoky záplavové (také nazýván útok hrubou silou) a útoky zaměřené na využívání slabin (sémantické) a útoky zaměřené na zneužití zařízení třetích stran. Záplavový útok zaplavuje oběť velkým množstvím žádostí, a tak vyčerpá její zdroje. V současné době se převážně používají záplavové. Útoky využívající slabiny se zaměřují na chyby v programu. Při útoku si připraví žádosti, které jsou cíleny na tyto slabiny tak, že je oběť buď nebude schopna zpracovat, což zapříčiní výpadek, nebo na jejich zpracování vynaloží velké množství zdrojů.

Také se rozlišuje mezi spojově orientovanými útoky (útok proběhne až po navázání TCP spojení s obětí) a nespojově orientovanými útoky (k provedení útoku není potřeba navázat spojení).

Dále jsou útoky rozlišovány podle toho, na kterou vrstvu referenčního modelu OSI/ISO je útok cílen. Bezpečností mechanizmy by se měly zaměřovat na zabezpečení všech vrstev. Na obrázku 1.6 je vidět referenční model OSI/ISO [11] a přehled některých útoků, které jsou mířené na danou vrstvu [12][13]. Typů útoků je daleko víc a pořád vznikají nové způsoby zaměřené na nové chyby. Rozdělení v tabulce je pouze informativní a slouží k pochopení toho, co je cílem útoku na dané vrstvě. Některé útoky mohou zasahovat do více vrstev najednou.



Obr. 1.6 Model OSI a na typy útoků

1.2.1 Fyzická vrstva

Fyzická vrstva [11] slouží k fyzické komunikaci mezi zařízeními. Popisuje parametry přenosového kanálu. Jejím úkolem je aktivace, udržování a uzavírání fyzických spojení.

Do kategorie DoS útoků může patřit i útok na fyzickou vrstvu, tedy na samotný hardware (kabely, konektory, opakovače). Může se jednat o cílený útok s úmyslem poškodit nějakou část HW, nebo o náhodu, jako jsou různé přírodní katastrofy [12][13].

1.2.2 Linková vrstva

Linková vrstva [11] má za úkol přenos datových bloků (rámců) a kontrolu chyb při přenosu. Vytváří, udržuje a ruší spojení mezi dvěma komunikujícími, protilehlými vrstvami. Rovněž do rámců přidává hlavičku obsahující MAC adresy. Na této vrstvě pracují přepínače.

Příkladem útoku na tuto vrstvu je MAC flooding. Útočník posílá nadměrné množství paketů s různými MAC adresami. Toto velké množství MAC adres může zaplnit tabulku na směrovači a dojde k tomu, že se přepínač začne chovat jako rozbočovač. To znamená že začne rozesílat rámce na všechny své porty [12][13].

1.2.3 Síťová vrstva

Síťová vrstva [11] je třetí vrstvou referenčního modelu a slouží k zajištění komunikace mezi nesousedícími uzly. K tomu se využívá mezilehlých uzlů (směrovačů). Také zajišťuje směrování mezi uzly. S ostatními zařízeními dokáže komunikovat i přes nesouvisející vlastnosti sítových technologií. Formátuje data vyšších vrstev do rámců, které obstarává hlavičkou obsahující IP adresy.

Útoky [12][13] zaměřené proti této vrstvě slouží k zahlcení šířky pásma sítě. K tomu využívá různých obměn ICMP paketů. Příkladem je Ping of death, Ping flood nebo Smurf attack.

1.2.4 Transportní vrstva

Na transportní vrstvě [11] se datové jednotky z vyšších vrstev dělí do segmentů, nebo datagramů, podle využívaného protokolu. Vrstva má za úkol dělení, identifikaci a zpětné skládání datových toků. Dále dělí datové toky různých aplikací a rozlišuje je pomocí portu.

Obdobně jako u síťové vrstvy jsou i na transportní vrstvě útoky [12][13] cíleny na spotřebování šířky pásma sítě. Útoky se snaží vyčerpat maximální počet možných vytvořitelných spojení. Vytvářejí se žádosti o spojení, ale samotné spojení se nikdy nevytvoří. Příkladem je Syn flood.

1.2.5 Relační vrstva

Relační vrstva [11] slouží k navázání, udržování, rušení relací a v případě potřeby řídí komunikaci. Také synchronizuje a organizuje dialog mezi relačními vrstvami komunikujících systémů.

Útoky na této vrstvě [12][13] jsou cíleny na aplikace a uživatele využívající program telnet ke komunikaci. Může se jednat o odposlouchávání komunikace (telnet postrádá šifrování), útok hrubou silou (zkoušení hesel), nebo o DDoS útok (chyba v SW – odesílání velkých množství rámců vyřadí službu telnet z provozu).

1.2.6 Prezentační vrstva

Hlavní funkcí prezentační vrstvy [11] je transformování dat z aplikační vrstvy do podoby, aby byla srozumitelná nižším vrstvám (shodné kódování, datové formáty, syntaxe). Má na starost také šifrování. Zajímá ji pouze forma zprávy, obsah zprávy má na starost aplikační vrstva.

K útokům na tuto vrstvu [12][13] se využívají pozměněné SSL dotazy. SSL se vyžívá při šifrované komunikaci. Využívá se u online transakcí nebo při zabezpečeném přístupu k webovým stránkám. Pří útoku se generuje nadměrné množství chybných SSL požadavků. Kontrola žádostí serveru zabere značné množství času a může dojít i k omezení služby.

1.2.7 Aplikační vrstva

Aplikační vrstva [11] je sedmá, nejvýše umístěná vrstva v referenčním modelu OSI/ISO.

Za úkol má tvorbu dat pro přenos a zpracovávání přijatých zpráv (určuje formát dat). Na aplikační vrstvě se nacházejí programy a služby jako jsou: souborové servery, emailové servery (protokol POP3, SMTP), webové prohlížeče (protokol HTTP) nebo služby pro přenos souborů (protokol FTP). Často je k obsluze těchto programů zapotřebí uživatel.

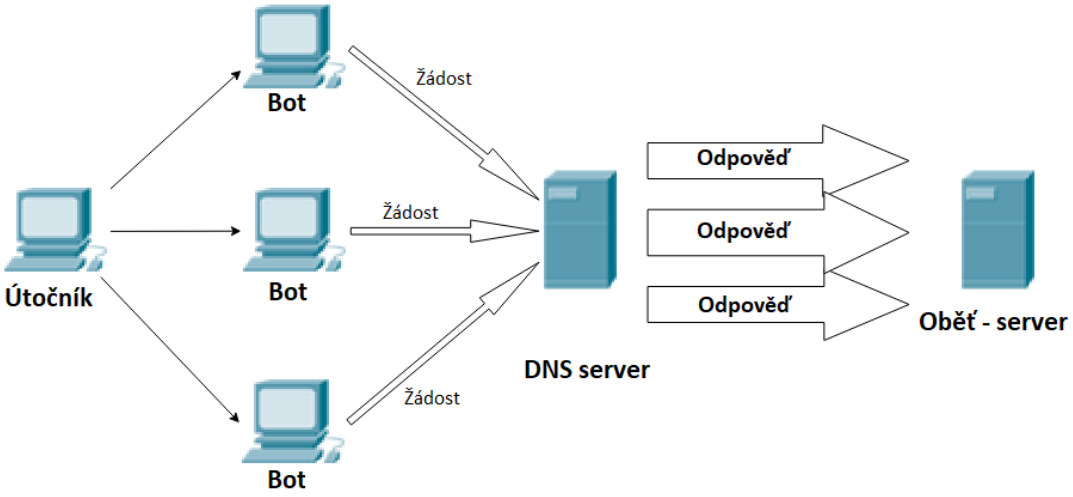
Útoky na tuto vrstvu často využívají protokolu HTTP. Na této vrstvě je obtíže rozlišit datový provoz útočníka a legitimní datový provoz. Příkladem je HTTP GET, HTTP POST, PDF GET request [12][13].

1.3 Útoky využívající jiných zařízení

V úvodu podkapitoly 1.2 bylo uvedeno rozdělení útoků podle způsobu útoku (záplavové, využívající slabin programu, využívající zařízení třetích stran). Útoky využívající jiná zařízení lze dále rozdělit na zesilující útoky a odražené útoky.

1.3.1 Zesilující útok

Zesilující útoky [14] slouží k zesílení datového toku. K tomu využívají prostředníky – veřejně dostupné servery k obsluze požadavků (např. DNS server) Zesílení je realizováno tak, že útočník pošle dotaz na server a v dotazu zamění zdrojovou IP adresu viz 1.1. K vygenerování takového dotazu spotřebuje málo svých zdrojů, ale odpověď od serveru je mnohem datově objemnější. Tyto odpovědi jsou kvůli zaměněné zdrojové adrese odesílány oběti útočníka.



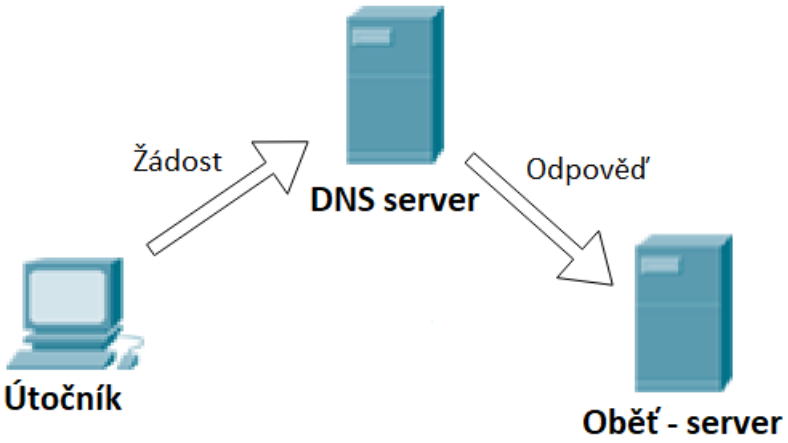
Obr. 1.7 Zesilující DDoS

Na obrázku 1.7 můžeme vidět DDoS zesílení útoku. Botnet obsahuje tři boty, kontrolované útočníkem. Každý bot generuje dotazy s podvrženou zdrojovou IP na DNS server. Daleko objemnější odpovědi jsou odesílány oběti útoku.

1.3.2 Odrazený útok

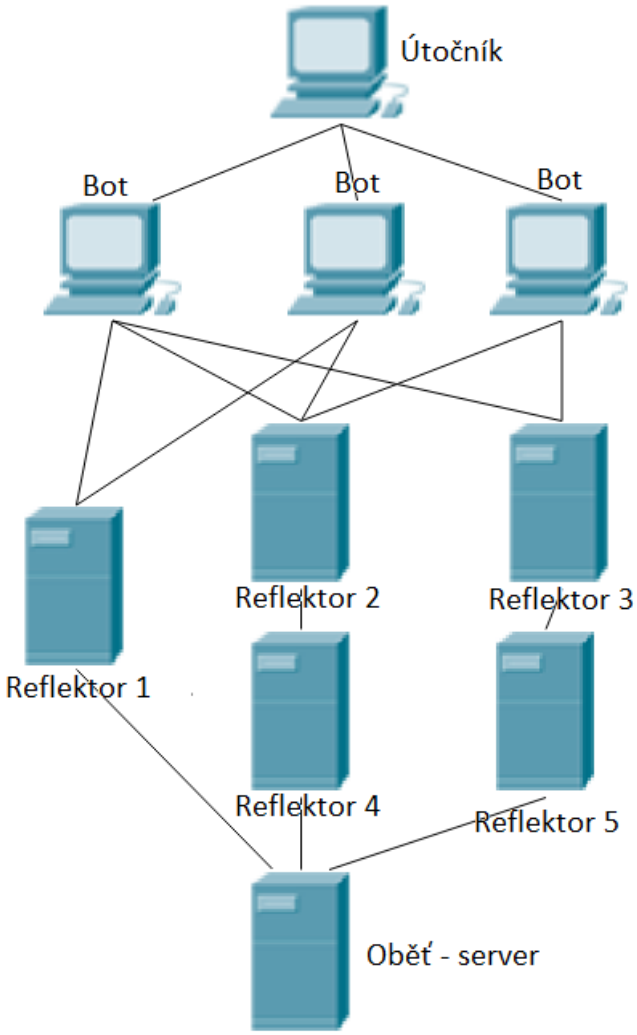
Odrazené otoky [14] využívají rovněž zařízení třetích stran (reflektorů), ale trochu jinak než u zesilujících útoků. Při provádění útoku se pozmění zdrojová IP adresa za adresu

oběti. Poté co server obdrží žádost, odesílá odpovědi oběti útoku, od které si myslí, že přijal žádosti. Cílová adresa udává adresu reflektoru. Toto ještě více ztěžuje dohledání pachatele.



Obr. 1.8 Reflektivní útok

Na obrázku 1.8 můžeme vidět odrazený útok v klasickém pojetí (DoS). Jeden útočník využívá odrazení od serveru k útoku na svůj cíl.



Obr. 1.9 DRDoS

Pokud DDoS využívá odražení, tak se tento útok nazývá Distributed Reflektor Denial of Service (DRDoS) [2]. Postup útoku je prakticky stejný jako u klasického DDoS, ale boti následně využívají prostředníků k útoku na cíl. Na obrázku 1.9 je vidět ukázka distribuovaného reflektivního útoku.

Existují útoky využívající zesílení i odražení. Tyto útoky jsou ještě nebezpečnější, protože poskytují zamaskování pachatele a zároveň zvětšení datového provozu.

1.4 Často se vyskytující nebo známé typy útoků

Tato podkapitola popisuje základní funkce nejčastěji se vyskytujících nebo významných DoS a DDoS útoků.

ICMP flood

ICMP flood [16] je záplavový útok cílený na síťovou vrstvu. ICMP flood vznikl jako klasický DoS útok, ale lze ho provádět i distribuovaně. K útoku využívají ICMP pakety [15]. Ty mají hned několik různých funkcí jako například oznámení

o nedostupnosti cíle paketu nebo k informování odesílatele o změně cesty k cíli. Další funkcí ICMP je ping. Ping normálně slouží k zjištění dostupnosti uzlu nebo koncové stanice. Na požadované zařízení (její IP adresu) se odešle paket ICMP Echo. Pokud je dotazovaná stanice nebo uzel dostupný, tak odpoví paketem ICMP Echo Reply.

Útok probíhá tak, že se odesílá velké množství paketů ICMP Echo (jeden paket má povolenou maximální velikost 65 kB) na stanici a nečeká se přitom na odpověď. Pokud stanice odpovídá, tak se šířka pásma vyčerpá dvakrát rychleji, kvůli příchozím dotazům a odchozím odpovědím.

UDP flood

UDP je bezstavový protokol na transportní vrstvě. Oproti TCP, který je stavový, není potřeba vytvořit nejprve spojení mezi dvěma, nebo více komunikujícími uzly. Výhodou tohoto protokolu je rychlejší komunikace z důvodu menších hlaviček. Nevýhodou je celkem nebezpečná komunikace. Data se mohou na přenosové cestě ztratit, duplikovat nebo zpřeházet.

Jak již z názvu vyplývá, tak se jedná o záplavový útok. Cílem je zahltit celou datovou linku oběti vygenerovanými UDP pakety. Pokud tento útok využívá služby chargen a echo, stává se ještě efektivnější. Služba chargen slouží k odeslání náhodných dat poté, co přijme na svém portu nějaká data. Naopak služba echo negeneruje náhodná data, ale posílá data, která přijala zpět na stejný port. Dalo by se říct, že tento typ útoku je reflektivní. Útok probíhá tak, že si útočník nejprve najde svou oběť a nějaký server, nebo stanici, která má povolenou funkci echo, nebo chargen. Poté vygeneruje paket, obsahující cílovou adresu oběti a podvrženou zdrojovou adresu jiného zařízení. Rovněž nastaví porty na některou ze zmíněných služeb. Jakmile paket odešle, tak si začnou v nekonečné smyčce vyměňovat pakety mezi obětí a jiným zvoleným zařízením. Tento úrok je efektivní, ale služby echo a chargen bývají ve většině případů vypnuté [16][17].

Ještě se používá jiný typ záplavového UDP útoku [18]. Tento typ rozesílá pakety s různými čísly portů. Snaží se docílit toho, aby stanice vynaložila úsilí a zdroje k tomu, aby zjistila, jestli na daném portu naslouchá nějaká aplikace. Pokud tomu tak není, tak se navíc odesílá ICMP paket informující odesílatele o nedostupnosti. Tento typ útoku lze provést z jedné stanice (DoS), nebo je ho možné provádět distribuovaně (DDoS). Útočník samozřejmě používá podvržené IP adresy, aby nezaplavil sebe, nebo boty s velkým množstvím ICMP paketů a následně by nemohl pokračovat v útoku.

TCP flood

Protokol TCP [11] je stavový protokol transportní vrstvy. Před uskutečněním komunikace je potřeba navázat spojení. To se provádí pomocí „potřesení rukou“ (three-way handshake). Vytvoření spojení je zahájeno odesláním paketu SYN a odesílá jej strana, která požaduje službu (většinou klient požaduje službu od serveru – klient odesílá paket SYN). Server odpovídá paketem SYN-ACK. Pro úspěšné vytvoření spojení ještě klient odpovídá paketem ACK. Při vytváření spojení se samozřejmě využívá sekvenčních čísel. Útoků TCP flood je hned několik typů. Asi nejznámější je Syn flood. Útok Syn flood je záplavový útok a je ho možno provádět jako DoS i DDoS. TCP flood cílí na to, aby vyčerpal veškerá možná spojení, které server může vytvořit. Vlastností navazování spojení je i čekání na potvrzení od toho, kdo zahájil three-way handshake. Důvodem čekání je například možné zpoždění při přenášení paketů sítí. Této vlastnosti

útočník využívá tak, že odešle co nejvíce paketů SYN. Server správně odpovídá paketem SYN-ACK a čeká na příchod ACK. Po dobu čekaní jsou alokovány prostředky pro budoucí komunikaci. Velkým množstvím otevřených spojení dochází k vyčerpaní zdrojů. Syn flood v podobě DDoS útoku nutí boty, aby spojení vytvořili úplně. Toto rovněž vyčerpá množství vytvořitelných spojení a zároveň to obejde ochrany proti Syn flood [19].

Smurf

V podkapitole 1.3.1 a 1.3.2 byly popsány základní principy odražených a zesilujících útoků. Prvním zástupcem těchto útoků je Smurf [20]. Tento útok je zesilující a také odražený útok. Útok funguje tak, že rozešle záplavu ICMP paketů do různých sítí, z těchto sítí a všech zařízení v síti dostane odpověď v podobě ICMP Echo Reply. Tímto bylo dosaženo zesílení a odražení. Aby měl útok nějaký význam a aby si útočník nezahltil vlastní spojení, tak musí zaměnit svou zdrojovou adresu za adresu oběti. To vede k tomu, že ping odeslaný do různých sítí vygeneruje objemnější odpovědi a zaplaví jimi oběť útoku.

DNS Amplification Attack

Dalším zástupcem odražených a zesilujících útoků je DNS Amplification Attack [20]. Pro odražení a zesílení datového toku využívá tento útok veřejných DNS serverů. Útočník posílá DNS dotazy serveru. Zdrojová adresa dotazů je pozměněna na adresu oběti. Chytrým kladením dotazů je možno získat velkého zesilujícího efektu. Podobně fungují zesilující útoky i na jiné servery (SNTP, NTP) rozdíl je v podobě dotazu a v tom, jak velké zesílení jsou schopny různé servery/služby vyvinout.

Distributed Reflection Denial of Service

DRDoS (Distributed Reflection Denial of Service) [21] je distribuovaný odražený útok, využívající SYN flood. Útočník, respektive jeho boti rozesílají žádosti pro vytvoření spojení (pakety SYN) různým serverům. V paketech je podvržena zdrojová IP za IP oběti. Servery odpovídají paketem SYN-ACK a odesílají je oběti. Ta je zaplavována velkým množstvím těchto paketů a dochází k zahlcení. Pokud není nastaveno jinak, tak servery čekají (na „ztracený“ paket ACK) a pokud nepřichází odpověď od oběti, tak odesílají ještě několik SYN-ACK.

DDoS-as-a-Service

V poslední době se začaly rozšiřovat skupiny poskytující DDoS-as-a-Service (DDoSaaS) [22]. Jedná se o způsob, jakým se provádí různé útoky DDOS. Ty provádějí útoky na zákazníkem zvolené cíle, jejichž délka se odvíjí od finančního ohodnocení. Skupiny nabízejí různé druhy DDoS útoků. Tyto útoky mohou sloužit k testování vlastních sítí, nebo k obyčejným útokům a provádět je může i méně zkušený uživatel s přístupem k internetu.

Ping of death

Asi nejznámější útok využívající chyb v programech nebo protokolech je Ping of death [23]. Chyba v programu/protokolu nemusí vždy znamenat, že programátor udělal chybu záměrně nebo nevědomky, ale může to znamenat i to, že se nepočítalo, že by se daná možnost mohla vyskytnout. Jak již z názvu Ping of death vyplývá, tak se jedná

o útok využívající ping (ICMP Echo). Velikost celého ICMP paketu je omezena na 65.535 bytů. Útočník odesílal větší pakety, než je povolená délka a k tomu využíval defragmentování paketu. Tyto defragmentované pakety se u příjemce zpětně složí a může dojít k přetečení zásobníku pro jeden paket a pádu systému.

Útoky na SSL

SSL (Secure Sockets Layer) [24] je protokol k šifrování dat a autentizaci účastníků. Využívá se při převodech peněz, zabezpečeném přístupu k webovým stránkám, nebo k přístupu k poštovnímu serveru. K vyřážení této služby existuje několik přístupů. Podobně jako u Three-way handshake (který musí být před samotným šifrováním proveden), tak i SSL požaduje „handshake". Jedná se o dohodu mezi komunikujícími stranami na šifrovaní. Toto zabere velké množství výpočetního výkonu, čehož útočníci zneužívají. Generování nadměrných dat anebo neustále vyžadování znovu ustanovení SSL handshaku spotřebovává velké množství výpočetního výkonu a může dojít k pádu SSL serveru.

Pomalé útoky

Dalším typem útoků jsou slow (pomalé) útoky [25]. Tyto útoky se zaměřují na to, aby odesílaly jen malé množství dat a nebyly detekovány. Nejprve je potřeba navázat spojení. Poté útočník odesílá různé pakety (dle zvoleného útoku), ale jen v takové míře, aby se udrželo spojení otevřené a pro toto spojení zůstaly přiděleny zdroje. Takové malé množství paketu se velmi obtížně detekuje. Do této kategorie patří útok Slow GET. Tento útok využívá požadavek GET. GET slouží k získání webové stránky, obrázku, nebo jiného obsahu ze serveru. Útočník odešle nedokončený požadavek GET a server následně čeká na jeho dokončení. Čekání je omezeno na určitou dobou, kterou odpočítává server a po jejím konci by se spojeni označilo za vypršelé a uvolnilo by se. Útočník ale opět posílá další část požadavku a tím se obnoví doba čekání. Velkým množstvím navázaných spojení je možno vyčerpat zdroje serveru. Dalším typem útoků slow je Slow POST. POST se používá k odesílání dat na server. Útočník nastaví v prvním paketu do pole hlavičky délku dat jako nějakou obrovskou hodnotu. Ve skutečnosti soubor o této délce nemá a odesílá náhodná data pouze tak, aby obnovil časovače a spojení zůstalo nadále otevřené. Server poté od prvního příchodu POST očekává přicházející data, což vede k plýtvání zdrojů. Slow Read je dalším typem pomalých útoků. Tento útok je speciální v tom, že využívá velikost okna, které se používá při spojově orientované komunikaci (TCP). Klouzavé okno udává, jaké množství dat je možno odeslat bez potvrzení. Pokud je toto okno velké, tak může odesílatel odeslat více dat najednou. V případě menšího okna se zase čeká na potvrzení po odeslání menších částí. Změnou velikosti okna lze korigovat rychlost odesílaní dat. Velikost okna se může v průběhu přenosu měnit v závislosti na různých parametrech přenosové cesty. Tohoto zneužije útočník. Vytvoří požadavek GET na nějakou webovou stránku, nebo soubor a následně upraví velikost okna na co možná nejmenší. Data se budou číst velmi pomalu a pro přenos budou alokovány zdroje. Čím víc takových požadavků vznese, tím víc zdrojů bude vymezeno pro přenos.

Memcash útoky

Jedním z novějších útoků jsou útoky na memcash servery [26]. Memcash servery slouží obecně k urychlení operací, které požadují nejvíce času. Jedná se o obyčejnou paměť cashe s rychlým přístupem do paměti realizovanou deamonem. Příkladem využití

memcash je rychlejší přistupování k webovým stránkám. Útoky na memcash servery využívají slabiny velkého množství těchto serverů, čímž je neimplementovaná autentizace. I málo objemné dotazy (několik bitů) s podvrženou IP adresou za adresu oběti dokáží po odeslání na server vygenerovat mnohem větší odpověď (v řádu stovek kilobitů).

Záplavy HTTP dotazů

Jedná se o distribuované záplavové útoky [27], které cílí na zaplavení serveru nadměrnými HTTP žádostmi. Do této kategorie může patřit HTTP GET a HTTP POST. Rozdíl mezi záplavovými a pomalými HTTP útoky je, že pomalé útoky posílají minimální množství dat, aby pouze udržely spojení a vyčerpávaly zdroje přidělené těmto spojením, zatímco záplavové útoky odesílají maximální množství požadavků a server je nestíhá zpracovávat. V případě HTTP GET se odesílá tak velké množství požadavků GET, že úplně vytěžují danou službu a ostatní legitimní uživatelé mohou tuto službu využívat pouze s omezenou rychlostí nebo bude daná služba úplně nedostupná. HTTP POST se využívá například při vyplňování formulářů. Útočník opět zahltí server těmito požadavky a tím může znefunkčnit službu.

2 OBRANA PROTI DOS A DDOS

Při ochraně proti DoS a DDoS útokům existuje hned několik přístupů [1]. V základu je lze rozdělit na prevenci, detekci, identifikaci a reakci. Kvůli pořád nově se vyskytujícím útokům není žádná ochrana stoprocentně účinná, ale správnou implementací všech čtyř přístupů lze zmírnit dopady útoku na ostatní uživatele.

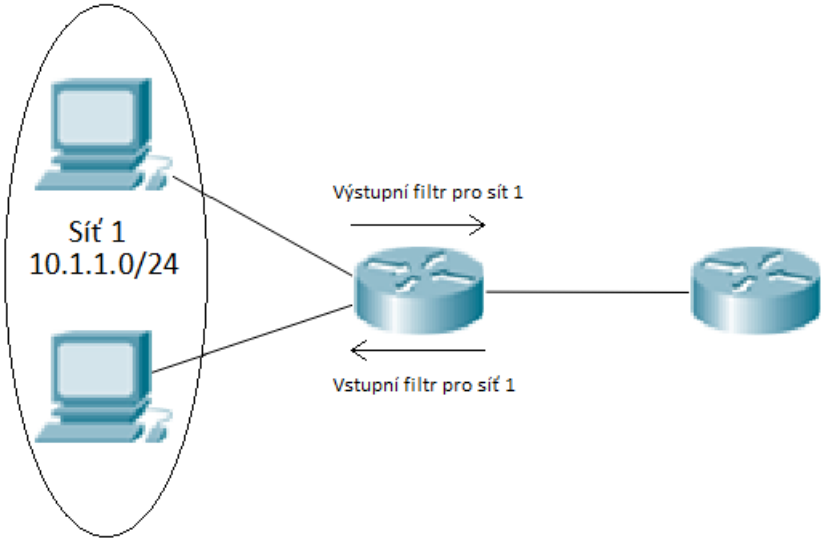
2.1 Prevence

Prevence [4] je jednou s nejúčinnějších ochran. Vychází z toho, že o útocích, které se jíž někde vyskytly, existují nějaké záznamy (logy, záznamy provozu při útoku). Příkladem mohou být útoky využívající slabin systému. Prevencí před těmito útoky je nainstalování nejnovější ochranné záplaty, které dalšímu útoku stejného typu zabrání. Z tohoto vyplývá, že nejsou všechny prvky sítí stejně zabezpečeny. Někteří administrátoři nedostatečně zabezpečí síť (neaktuální SW), a mohou být častěji vystaveni útokům.

Obětí útoku může být koncová síť nebo mezilehlá síť (přenosová). Stejným způsobem lze přistupovat i k implementaci ochrany těchto sítí. Uzly mezilehlých sítí využívají monitorovací a filtrovací mechanizmy. U cílových sítí se klade důraz na zabezpečení jednotlivých zařízeni sítě (bezpečnostní aktualizace – odstraňování chyb). Také vypnutí nepoužívaných služeb a zavření nepoužívaných portů napomáhá k ochraně proti útokům, které by mohly cílit na tyto služby, nebo otevřené porty. Dalšími prvky ochrany mohou být vyvažovače zátěže. Rovněž zvýšení počtu zdrojů pomáhá k zmírnění DDoS útoků (větší množství serverů a alternativních cest). Toto pouze zajišťuje, že útočník bude muset vynaložit větší úsilí k provedení útoku a zrušení služby. Další technikou může být přidělování zdrojů pouze ověřeným uživatelům. Jednotlivé stanice v sítí by měly být chráněny před škodlivým kódem. DDoS útoky jsou možné, protože je spousta uživatelských stanic nedostatečně chráněna a stávají se často účastníky botnetu. Ochranou pro stanice je antivirový program a firewall [28].

2.1.1 Filtrování na vstupu a výstupu

Filtrování slouží k tomu, aby do nebo ze sítě propustilo jen pakety, které jsou o očekávaných rozsazích IP adres [29]. Filtrování vstupu filtruje všechny pakety vstupující do lokální sítě a filtrování výstupu filtruje pakety odesílané z lokální sítě. Toto filtrování dokáže částečně zabránit IP spoofingu. Útočník může filtrování obejít, když použije při spoofingu IP adresu z očekávaného rozsahu, nebo v případě DDoSu ponechá botům v sítích jejich IP adresy. Na obrázku 2.1 můžeme vidět ukázkovou topologii. Síť 1 přiděluje stanicím v síti IP adresy z rozsahu 10.1.1.1 – 10.1.1.254. Výstupní filtr proto očekává, že v paketech odesílaných z této sítě bude zdrojová adresa vždy z tohoto rozsahu. Stejně to bude i u vstupního filtru, kde je potřeba správně zvolit rozsah očekávaných IP adres.[2] Obdobou je filtrování na základě historie IP adres. Tento mechanizmus si namísto hlídaní, zdali je IP z paketů v očekávaném rozsahu, vytváří vlastní databázi IP adres. Databázi sestavuje z IP adres příchozích paketů a portů, na které tyto pakety přišly.



Obr. 2.1 Vstupní, výstupní filtr

2.1.2 Filtrování paketů na mezilehlých směrovačích

Podobně jako jsou využívány filtry před vstupem, nebo výstupem z koncové sítě, tak lze používat filtry i na směrovačích v mezilehlých sítích [2]. Obdobně jako v předchozím případě se předpokládá, že na směrovač se dostanou jen určité IP adresy. Na internetu se vyskytují autonomní systémy (AS). Jeden AS může být složen z jedné nebo více sítí. Pro přenos dat v mezilehlých sítích se využívá Border Gateway Protocol (BGP). Je to dynamický směrovací protokol, který umí reagovat na změny v topologii. Tento protokol následně přenáší pakety mezi jednotlivými AS. Při filtrování se využívá směrovacích informací protokolu BGP a podle nich se vyhodnocují IP adresy, které se neočekávaně vyskytly na jiném portu a jsou tedy podvržené.

2.1.3 Filtrování s potvrzováním zdrojové adresy

U předchozích filtrů se očekávalo, že rozsahy IP adres jsou neměnné. Proto byl vymyšlen nový protokol s názvem Source Address Validity Enforcement (SAVE) [2]. SAVE se přizpůsobuje změnám a konstantně posílá ostatním směrovačům, využívajícím SAVE, zprávy, které obsahují informace o očekávaných IP adresách na každém portu. Podobně jako u filtrování vstupu je očekáváno že lokální síť připojení ke směrovači má neměnný rozsah. Útočníci jsou nuceni spoofovat IP adresy pouze za adresy z podsítě, jinak budou pakety filtrovány.

2.1.4 Autentizované adresy

Jedním z přístupů zabránění útokům je přidělování zdrojů jen ověřeným uživatelům. Proto byly navrženy různé mechanizmy pro jejich ověřování. Prvním z nich je Host Identity Protocol [10]. Tento protokol je navržen speciálně pro ověřování stanic a identifikuje je podle šifrovacího klíče, který je uložen v poli Host Identifier. V architektuře tohoto protokolu slouží IP adresy k identifikaci lokality stanice a Host Identifier slouží k identifikaci samotné stanice. Tyto identifikátory jsou dále změněny pomocí hashovací funkce, ke které má klíč pouze skutečný vlastník identifikátoru, čímž

je zajištěna autentizace účastníka. Dalším protokolem pro autentizaci je Accountable Internet Protocol (AIP) [30]. Tento protokol je založen na předpokladu, že v celém internetu jsou AS, nebo určité menší domény (accountable domains – AD), které jsou jednoznačně identifikovatelné a v těchto AS se vyskytují zařízení, které mají rovněž svůj identifikátor (Endpoint Idenfiticator – EID). Protokol využívá k jednoznačné identifikaci stanic předpis adresy: AD:EIDif. Tento předpis znamená, že jednoznačná identifikace stanice je složena z identifikátoru domény AD, identifikátoru stanice EID a portu (interface – if). Ověření správnosti této identifikace je opět prováděno pomocí veřejného klíče a hashovací funkce. Dalším mechanizmem je Secure Overlay Service (SOS) [10]. Ten je založen na tom, že komunikace mezi chráněnými sítěmi probíhá skrze zabezpečený tunel. Chráněné sítě se skládají ze směrovačů, které odesílají data ven ze sítě. V zabezpečené síti se také zvolí některé zařízení, které se bude nazývat beacon a bude sloužit k autentizaci ostatních komunikujících zařízení. Všechna zařízení v chráněné síti se musí nejprve připojit k beaconu, aby se autentizovala a následně mohou komunikovat. Efektivita všech výše popsaných mechanizmů závisní na rozsahu nasazení.

2.2 Detekce

Pokud se útočníkovi, i přes snahu útoku zabránit (prevencí), povede útok provést, je potřeba použít vhodné detekční mechanizmy [2]. Existuje mnoho typů útoků. Při podezření z útoku je vhodné sledovat několik příznaků. Základním příznakem je nedostupnost služby (např. webových stránek), nebo velmi pomalá načítací doba. Útoky jsou prováděny především záplavově, protože útoky zaměřující se na systémové chyby nemají dlouhou životnost. Záplavový charakter útoků napomáhá jejich detekci. Na druhou stranu se datový tok těchto útoků může zdát podobný datovému toku normálních uživatelů, což detekci ztěžuje. Včasnou detekcí můžeme zabránit ztrátám (zisků, úbytku uživatelů) a dále může sloužit k identifikaci útočícího zařízení, díky čemuž je možné na útok reagovat. Je proto vhodné využívat monitorovací nástroje.

Při detekci útoků se můžeme zaměřit na detekci určitých vzorů chování datového provozu a na detekci anomálií datového provozu. Detekce vzorů chování se zaměřuje na detekci již známých útoků. Naopak nové útoky nejsou vůbec detekovány. Detekce anomálií se zaměřuje na vytvoření modelu normálního provozu sítě a ten následně porovnává se skutečným datovým provozem, aby bylo detekováno nestandartní chování a případný útok. Zde musí být zvolena správná úroveň detekce. Pokud bude úroveň příliš nízká, útoky nemusí být vůbec detekovány. Naopak pokud bude hladina příliš vysoká, detekční mechanizmus bude často hlásit útok, i když to bude jen planý poplach. Vytvoření korektního modelu normálního chování je tedy zásadní. Aby nemusel být zaznamenáván kompletní datový provoz, což by vyžadovalo velký výpočetní výkon k porovnávání s normálním provozem, jsou zvoleny jen určité parametry, ze kterých je sestaven model.

MULTOPS [2] je jedno z detekčních schémat. Předpokládá, že datový tok v uplinku bude stejně velký jako datový tok v downlinku. Pokud tomu tak není, tak je detekován útok. V praxi se často stane, že downlink je větší než uplink. Příkladem je sledování videí online, nebo stahování souborů. Dalším návrhem detekčního systému je SYN detection [2]. Ten se zaměřuje na sledování množství přijatých paketů SYN ku FIN nebo RST. Tento návrh nesleduje celé relace vytvoření a ukončení spojení, ale pouze množství SYN a FIN paketů, čehož může útočník zneužít a při útoku SYN flood odesílat

mnoho paketů FIN. Účinnějším návrhem detekce je spektrální analýza datového toku [2]. Ta je založena na tom, že datový tok útoku má jiné vlastnosti (počet paketů, výkonová spektrální hustota) než normální datový tok. V případě že je detekován útok, musí být podstoupeny další kroky k jeho zamezení.

2.2.1 Intrusion Detection Systems

Výše popsaná schémata se mohou využívat v Intrusion Detection Systems (IDS) [31]. Jsou to systémy, který se zaměřuje pouze na detekci. Slouží pouze k sledování datového provozu sítě nebo aktivit systému. Při výskytu události, která neodpovídá pravidlům sítě, nebo která přímo útočí na síť, nebo její zařízení, zaznamená systém tyto události do logu a varuje administrátora. Systémy se rozlišují podle rozsahu jejich činnosti na Network Intrusion Detection Systems (NIDS) čili síťový IDS a Host-based Intrusion Detection Systems (HIDS), tedy IDS jednoho zařízení. Systémy využívají detekce anomálií i detekce vzorů chování datového toku.

2.2.2 Intrusion Prevention System

Intrusion Prevention System (IPS) [32] slouží také k monitorování síťového provozu. Oproti IDS má víc možností, které může podniknout v případě útoku. IPS může zahazovat pakety, vyvolat poplach, blokovat datový tok z IP adresy a nahlašovat tyto činnosti správci. Tento obraný mechanizmus bývá umístěn na linkách směrujících ven do internetu, a tak může aktivně sledovat a reagovat na útoky.

2.3 Identifikace zdroje útoku

Důležitou součástí ochrany proti útokům je identifikace [2] zdroje útoku. Toto není kvůli IP spoofingu vůbec jednoduché. Identifikaci také ztěžuje to, že směrovače znají pouze adresu dalšího směrovače, kterému musí pakety odeslat na cestě k cíli. Nepamatuje si tudíž celou cestu. Z těchto důvodů byla navržena schémata, která zlepšují vysledovatelnost IP.

První z metod k vysledování zdrojové IP je testování linek. Sledování probíhá tak, že na každém směrovači se sleduje, odkud pakety přicházejí. Takto se vysleduje celá trasa od oběti útoku až po zdroj. Z toho vyplývá, že útok musí neustále probíhat, protože směrovače si neuchovávají v paměti informace o odeslaných paketech.

Dále se využívá značkování paketů. Značka se ukládá do zřídka využívaného pole IP hlavičky paketu. Ze základní myšlenky se poté rozvinuly dva využívané postupy. Prvním z nich je pravděpodobnostní a druhý je deterministický. Pravděpodobnostní značkování označuje jen některé pakety. Z názvu vyplývá, že množství paketů, které projdou směrovačem, než dojde k označení, závisí na pravděpodobnosti. Tu lze nastavit různými způsoby (například 1/1000 = každý tisící paket bude označen). Tímto způsobem nedochází k postřehnutelnému zvětšení datového toku. Do 16bitového pole se ukládá informace o počtu přeskoků a informace o cestě. Pokud by bylo toto pole úplně vyčerpáno, lze pakety fragmentovat. Nevýhodou tohoto přístupu je, že pokud se nejedná o záplavový útok, ale pouze o útok využívající několik málo paketů, tak se může stát, že právě potřebné pakety nebudou označeny. Tomuto se snaží zabránit deterministické značkování. Značka se přidává do každého paketu a obsahuje pouze IP adresu hraničního

směrovače, na který paket přišel před vstupem do sítě. Tato metoda využívá hashovaní značky, aby ji nemohl útočník pozměnit a tím zabránit sledování. Nedokáže si poradit s podvrženou IP.

Další metoda sledování zdroje využívá ICMP. Opět se využívá pravděpodobnosti, ale místo značky v paketu se využívají ICMP zprávy. Ty se odesílají do počátečního a koncového směrovače na cestě. Jedná se o speciální ICMP, které obsahují informace o sousedících uzlech napříč celou cestou od zdroje k cíli.

Ideálně by se mělo využívat kombinace logování a značkování paketů, aby se předešlo nevýhodám jednotlivých metod.

2.4 Reakce

Včasnou reakcí [2] na útok lze zmírnit jeho následky. Jelikož jsou zdroje v síti sdíleny (spoje, směrovače atd.), musí se útoku zabránit co nejblíže u jeho zdroje. Pokud by se toto nepraktikovalo a útoku by se zabraňovalo jen u jeho cíle, tak by neustále docházelo k plýtvání zdrojů sítě. Reakci lze rozdělit podle jejího nasazení na správu zdrojů, reakci v mezilehlé síti a reakci u zdroje útoku.

Při záplavovém útoku vznikají v určitých místech topologie úzká hrdla. Vhodnou správou zdrojů lze zabránit jejich vzniku. Příkladem je změna front. Využitím Class-Based Queuing (CBQ) lze klasifikovat datový tok do tříd. Přidělením útočícího datového toku do třídy s nejnižší prioritou lze poskytnout datovým tokům ostatních uživatelů přednost před útokem.

Reakcí na útok u cílového zařízení je odstranění slabin. Příkladem je SYNkill, který se zaměřuje na rušení napůl otevřených spojení. Toto se dost mísí se samotnou prevencí. Jestliže jsou slabiny systému odstraněny před výskytem útoku, jedná se o prevenci. Pokud jsou slabiny odstraněny až poté, co je na ně upozorněno (útokem), jedná se o reakci. V mezilehlých sítích se využívá filtrování na směrovačích. Útoky je obtížné detekovat, a proto je vhodné udržovat určitým způsobem komunikaci mezi obětí útoku a směrovačem, který odfiltruje útok. Vzniklo několik schémat k filtraci útoků. Prvním z nich je Selektive Pushback. Směrovač dokáže rozeznat útok od normálního provozu. Útok filtruje sám, ale zároveň předává informace sousedním směrovačům na cestě a směrovač blíže ke zdroji přebírá úlohu filtrování. Takto se filtrování posouvá co nejblíže ke zdroji. Další schéma navrhlo architekturu agent-controller. Jako agent se označuje hraniční směrovač a controller je důvěryhodné zařízení. Pokud je nějaké zařízení v síti pod útokem, informuje o tom controller. Controller přikáže agentům, aby značkovali útočící datový tok. Jakmile se zjistí, ze kterého agenta vstupuje tento datový tok do sítě, tak se v tom místě začne filtrovat.

Ideálním přístupem by bylo odfiltrovat útok co nejblíže od útočníka. Schéma D-WARD bylo navrženo z tohoto důvodu. Toto schéma neustále monitoruje datový tok mezi zdrojem (umístěním monitorovacího nástroje) a zbytkem internetu. Má zaznamenané normální chování a sleduje určité statistiky aktuálního chovaní. Při detekci abnormalit začne datový tok filtrovat.

K zabránění DDoS útoků se v praxi často využívají výzvy. Jedná se o výzvu pro uživatele PC, aby provedl nějakou činnost, která je obtížná pro počítač, ale jednoduchá pro člověka. Příkladem může být přečtení textu ze zdeformovaného obrázku, nalezení

objektu v obrázku, nebo složení částí obrázku k sobě jako u puzzle. Podobného principu využívají i automatizované funkce. Například při dotazování na DNS, odešle DNS server výzvu, která je jednoduchá k vygenerování a vyhodnocení, ale obtížná pro řešení. Navíc jsou blokovány další požadavky od jedné stanice, dokud nebyla vyřešena zadaná výzva.

V případě že jeden ISP není schopen odfiltrovat útok, a tak i zamezit vyčerpání zdrojů, může požádat ISP, od kterého přichází útočící datový tok, aby se zapojil do filtrování.

3 MIKROTIK

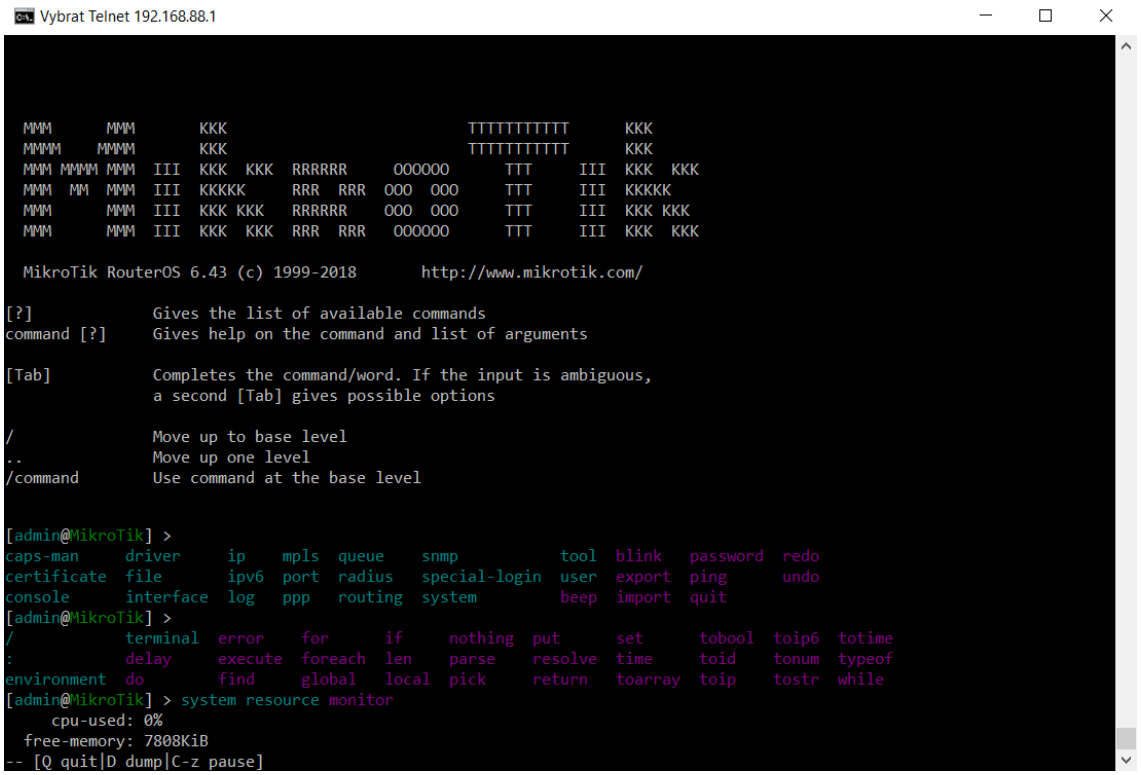
Společnost MikroTik [33] byla založena roku 1996 a zaměřuje se na tvorbu přepínačů, směrovačů a jiných zařízení pro ISP. Jejich aktivní prvky jsou určeny pouze pro menší a střední sítě. Nejznámější sérii výrobků této společnosti jsou směrovače nazývané MikroTik Routerboard. Jejich pořizovací cena je malá a nabízejí spolehlivost a na svou cenovou kategorii i velký výkon. V těchto zařízeních se využívá proprietární operační systém, rovněž vyvíjený MikroTikem, s názvem RouterOS. RouterOS je modulární operační systém. Tudíž nabízí možnost upravovat systém přidáváním nebo odebíráním balíčků s přídavnými funkcemi.

3.1 Komunikace se směrovačem

Existuje několik způsobů, jak komunikovat se zařízením za účelem jeho konfigurace nebo správy [34]. Komunikovat lze přes: konzoli, program WinBox, webové rozhraní WebFig a rozhraní API.

3.2 Konzole

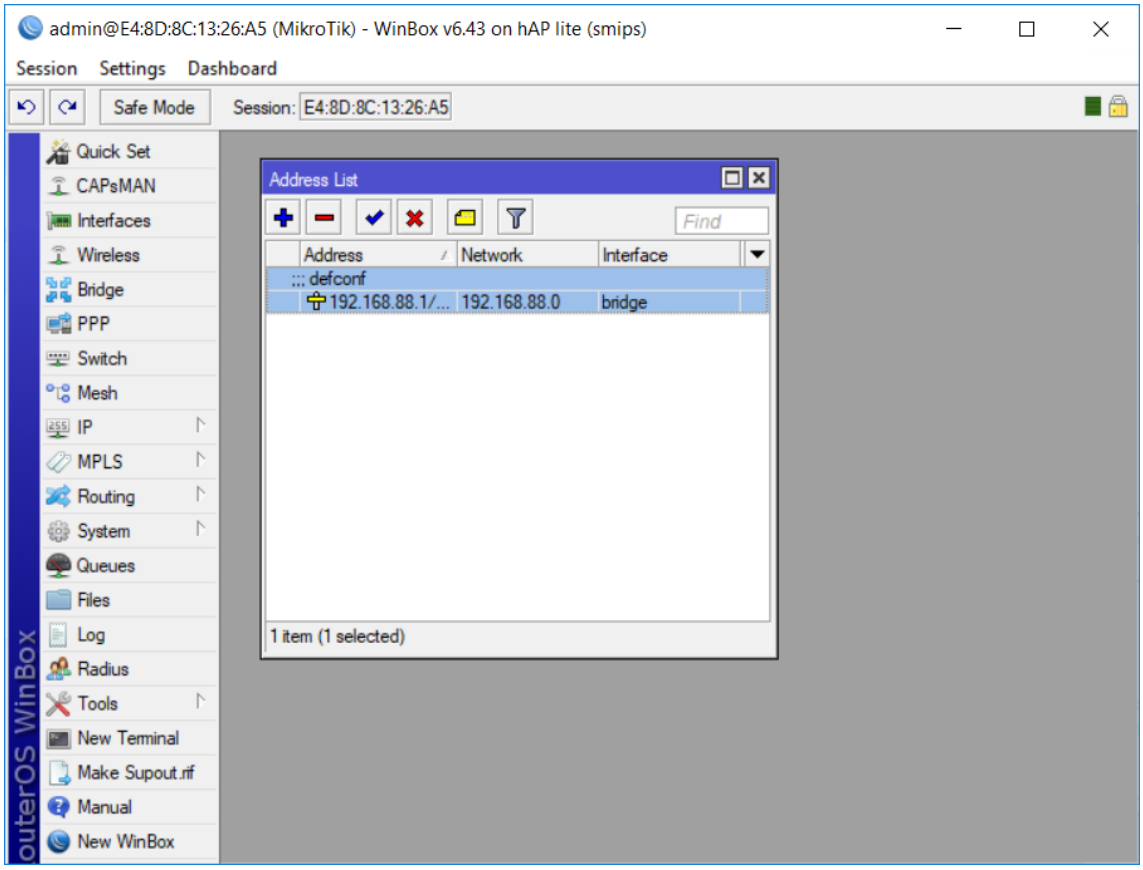
Pro připojení ke směrovači se využívá sériová linka, protokoly telnet nebo SSH (Secure Shell). SSH nabízí oproti telnetu zabezpečení dat formou šifrování. Telnet přenáší data v nešifrované podobě a kdokoliv, kdo odchytí komunikaci, může data přečíst. Konzolové zadávání příkazů uživateli nenabízí grafické prostředí. Příkazy jsou zadávány do příkazového řádku. Jelikož není zavedeno grafické prostředí, musí existovat velké množství příkazů pro konfiguraci. Příkazy jsou proto hierarchicky děleny. Příkladem je příkaz /ip. O úroveň níže jsou příkazy /ip route nebo /ip hotspot. Dalším ulehčením psaní příkazu je rychlé doplňování příkazů. Pokud začne uživatel psát příkaz a napíše jeho určitou část, která se neshoduje s žádným jiným příkazem, může zmáčknout klávesu TAB a příkaz se automaticky doplní. Příkladem je příkaz /interface, kdy uživateli stačí napsat /int a zmáčknout TAB. Příkaz se následně doplní na /interface. Jednou z funkcí, které RouterOS nabízí, je Safe Mode. Tato funkce slouží k tomu, aby uživateli nabídla možnost uložit změny, které provedl, nebo je neukládat. Na obrázku 3.1 je vidět výpis z konzole po přihlášení ke směrovači a příkaz pro vypsání vytížení procesoru spolu s dostupnou pamětí v reálném čase.



Obr. 3.1 Konzole

3.3 Winbox

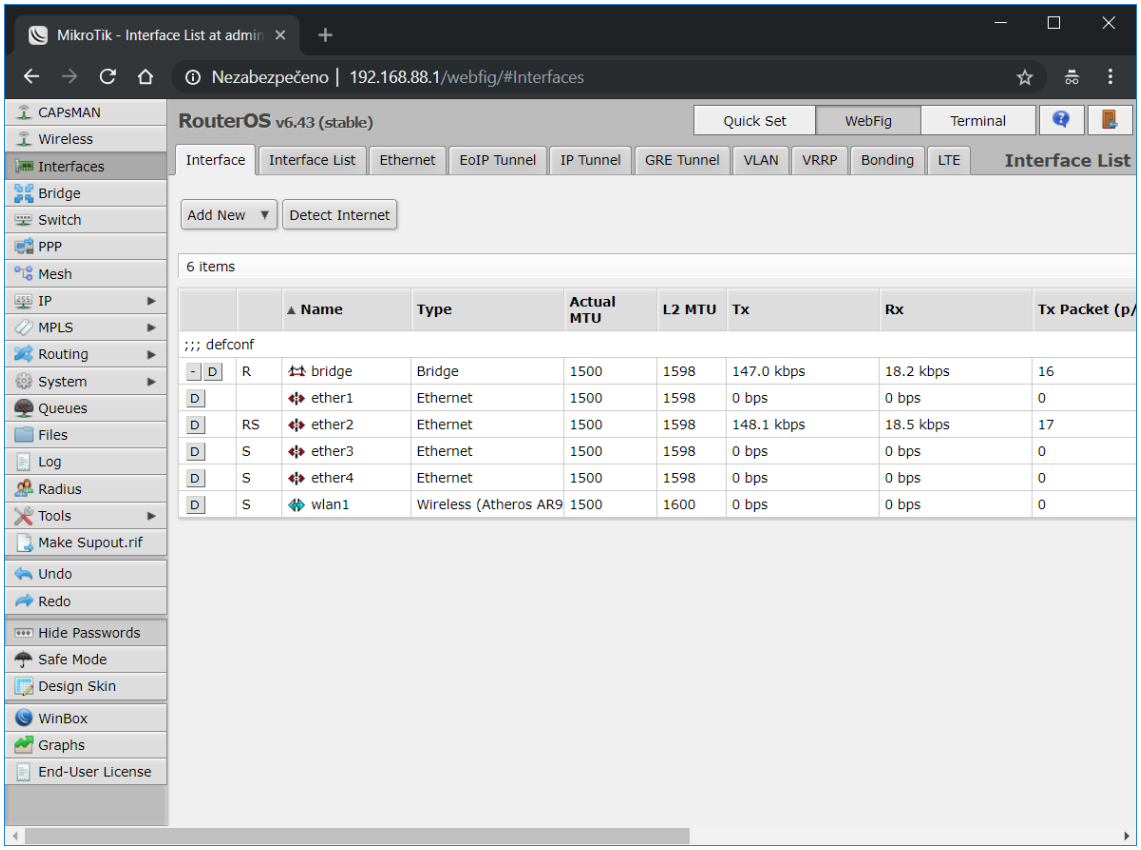
Winbox je program, který poskytuje uživatelům intuitivní grafické prostředí. Všechny jeho funkce je možné napsat příkazem v konzoli. Uživatelé si ale nemusí pamatovat všechny příkazy. Několik málo příkazů není možné provést v grafickém prostředí. Proto má Winbox implementovaný příkazový řádek, kam je možno zadávat příkazy ručně, stejně jako v případě konzolového přístupu. Z Winboxu je možné připojit se ke směrovači pomocí IP adresy nebo MAC adresy. Komunikace mezi programem a směrovačem je šifrovaná. Na obrázku 3.2 je vidět rozhraní programu Winbox.



Obr. 3.2 Winbox

3.4 WebFig

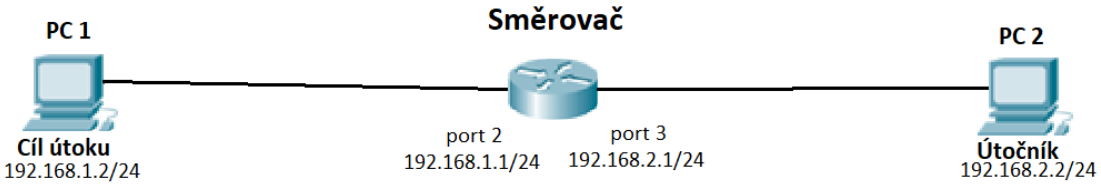
WebFig je webové rozhraní, které umožňuje nakonfigurovat směrovač stejným způsobem jako Winbox. Rovněž jeho grafické rozhraní je velmi podobné Winboxu. Do WebFigu je možné se dostat zadáním IP adresy směrovače do internetového prohlížeče. Proto není nutný žádný další SW. Na obrázku 3.3 je vidět rozhraní WebFig v internetovém prohlížeči (na obrázku byl použit Google Chrome).



Obr. 3.3 WebFig

4 VLASTNÍ EXPERIMENTÁLNÍ PRACOVIŠTĚ A SURICATA

K prvotnímu seznámení s útoky na odepření služeb bylo vytvořeno vlastní experimentální pracoviště, která je vidět na obrázku 4.1. V topologii byl použit směrovač MikroTik hAP lite RB941-2nD. Účelem bylo seznámit se s různými dostupnými programy k zátěžovému testování sítí, seznámit se s následky odepření služeb a seznámit se s programem Suricata.



Obr. 4.1 LAN schéma

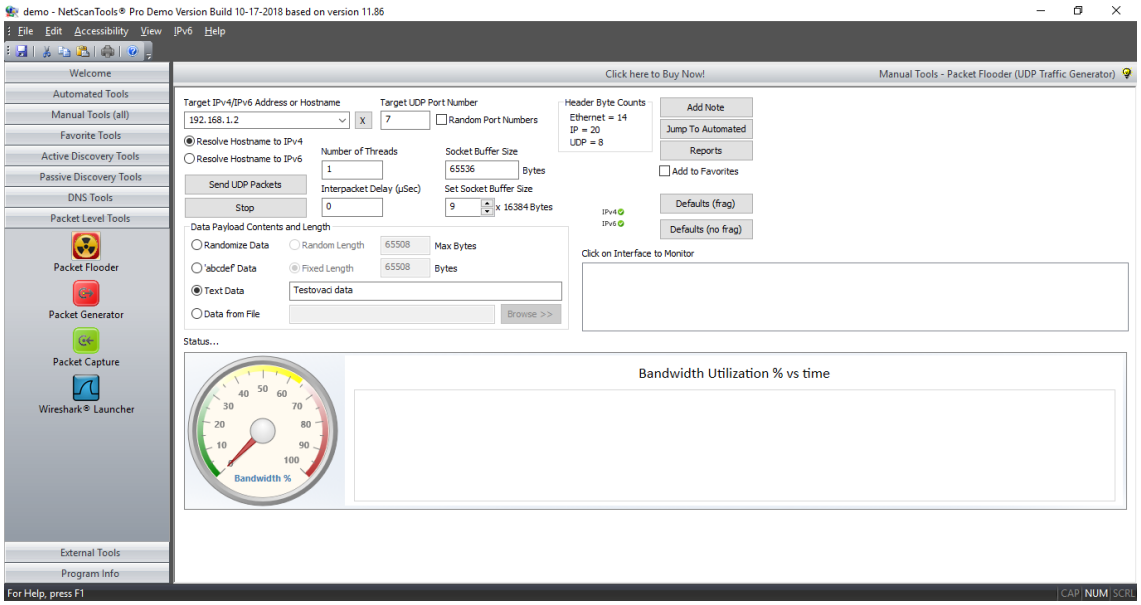
Před samotným generováním škodlivého datového toku bylo provedeno měření vytížení CPU při přihlášení, přes všechny tři výše zmíněné možnosti správy směrovače. Pokud bude cílem útoku infrastruktura sítě a některý ze směrovačů bude úplně vytížen, administrátorovi se vůbec nemusí podařit pokus o připojení k zařízení. V tabulce jsou uvedeny naměřené hodnoty pro jednotlivé způsoby přístupu. Hodnoty byly zaznamenávány jako maximální využití CPU v čase přihlášení k zařízení. Měření bylo provedeno dvacetkrát, výsledky byly zprůměrovány a vyneseny do tabulky 4.1. Nejnižší hodnoty dosahuje konzole, protože nemá žádné grafické rozhraní.

Tab. 4.1 Vytížení CPU při přihlášení

	Využití CPU [%]
Konzole	8,9
Webfig	21,6
Winbox	10,9

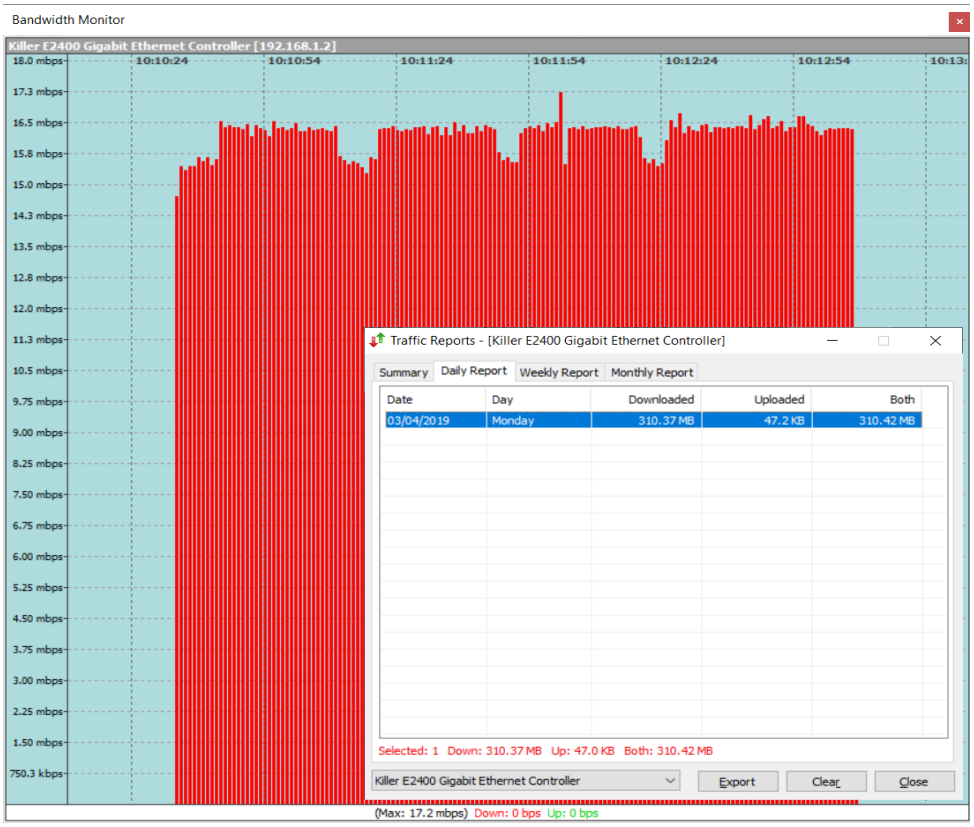
4.1 NetScan Tools

K testování byly využity dva generátory. Prvním z nich byl NetScan Tools [35], jehož rozhraní je vidět na obrázku 4.2. Jedná se o sadu programů pro OS Windows, sloužící pro testování různých vlastností sítě. Jmenovitě obsahuje nástroj pro detekci DHCP serveru, nástroj pro skenování portů, nástroje pro odchytávání a generování paketů a mnoho dalších. Množství nástrojů se odvíjí od zakoupené licence. Ke generování byl využit nástroj Packet Flooder. Slouží ke generování UDP provozu s možnou konfigurací parametrů daného provozu.



Obr. 4.2 Rozhraní NetScan Tool

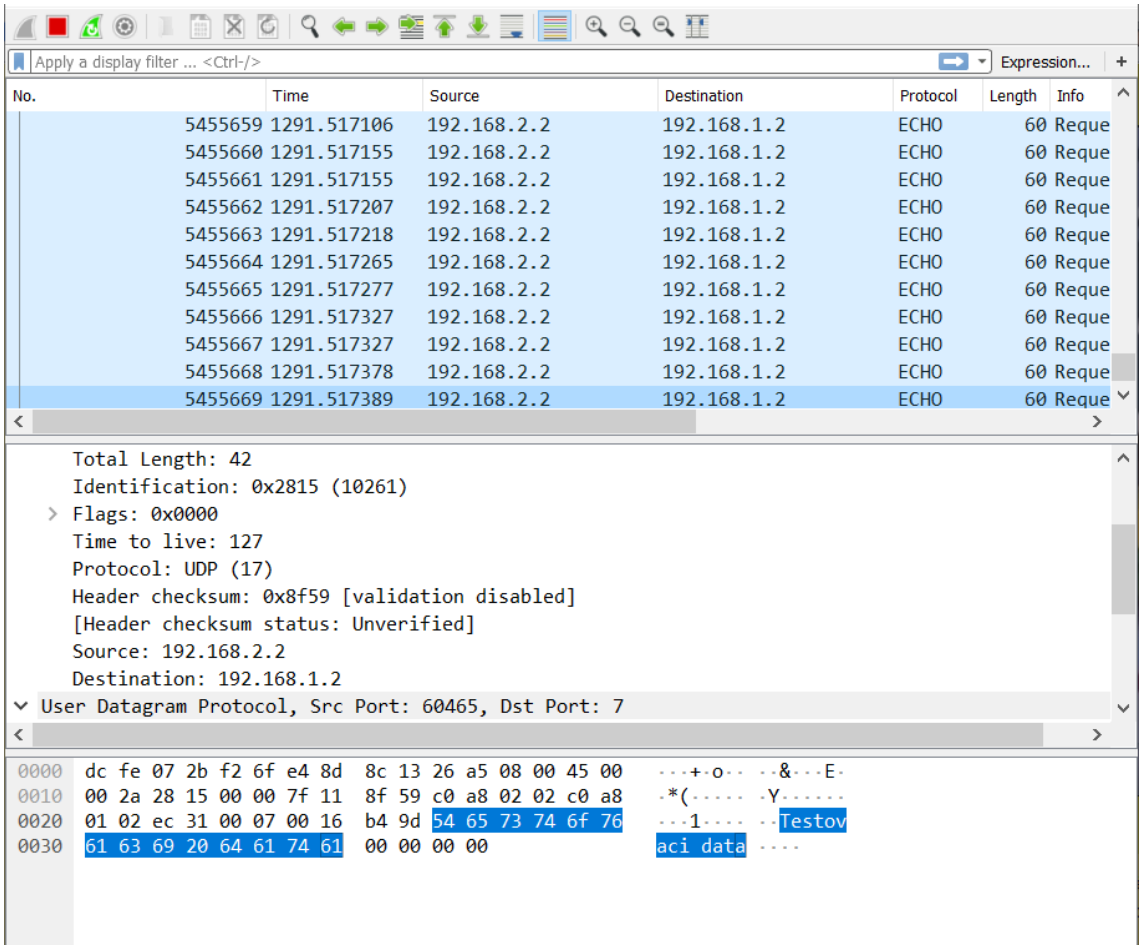
Po spuštění programu byl zaznamenán datový tok s rychlostí pohybující se kolem 16 Mb/s a během 2 minut bylo přeneseno 310 MB viz obrázek 4.3. Před testem a během testu byl z počítače oběti spuštěn příkaz ping (obrázek 4.4) na počítač útočníka. První dotaz na obrázku byl spuštěn před útokem, ostatní během útoku. Oproti pingu v klidovém stavu vykazoval ping při útoku větší odezvu, pokud nedošlo k úplné ztrátě daného ICMP paketu. Na obrázku 4.5 se nachází detail jednoho z paketů, odchyceného během útoku.



Obr. 4.3 Vygenerovaný datový tok programem NetScan Tools

```
C:\Users\Jakub>ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:
Reply from 192.168.2.2: bytes=32 time=1ms TTL=127
Reply from 192.168.2.2: bytes=32 time=1ms TTL=127
Reply from 192.168.2.2: bytes=32 time=1ms TTL=127
Reply from 192.168.2.2: bytes=32 time=1ms TTL=127

Ping statistics for 192.168.2.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 1ms, Average = 1ms

C:\Users\Jakub>ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.2.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\Users\Jakub>ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:
Request timed out.
Reply from 192.168.2.2: bytes=32 time=4ms TTL=127
Reply from 192.168.2.2: bytes=32 time=5ms TTL=127
Request timed out.
```

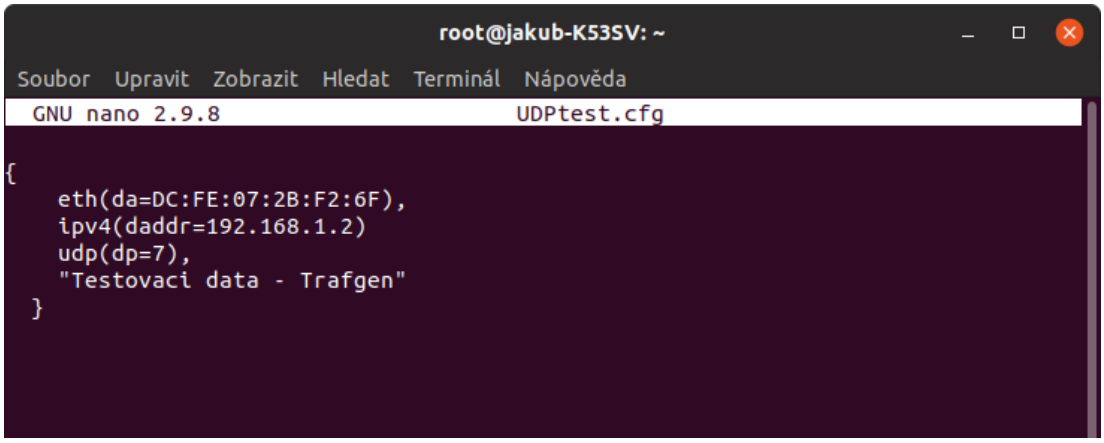
Obr. 4.4 Ping u prvního útoku



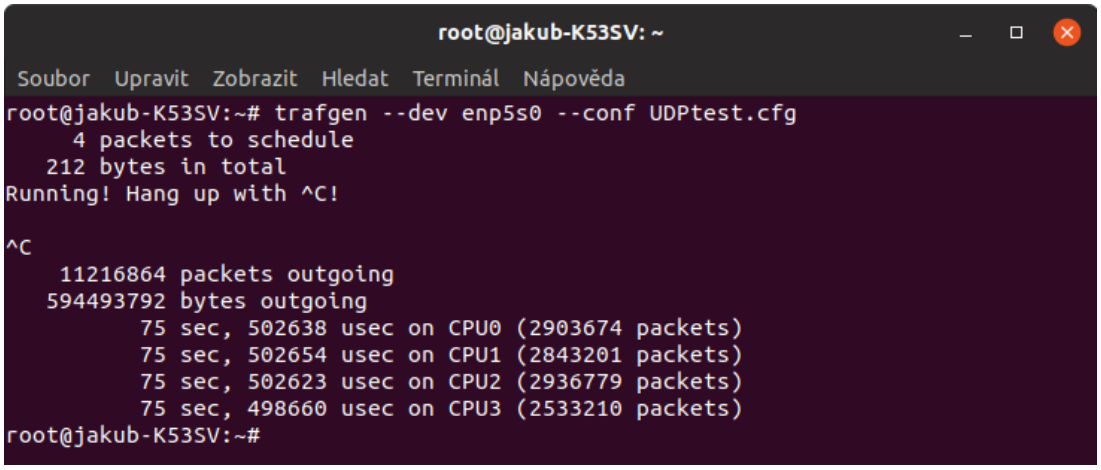
Obr. 4.5 Wireshark při prvním útoku

4.2 Trafgen

Druhým použitým generátorem byl Trafgen [36]. Jedná se o nástroj ke generování datového provozu z nástrojové sady s názvem Netsniff-ng. Je součástí balíku nástrojů NETSNIFF-ng, ale je možné ho využívat samostatně. Balíček nástrojů je určen pouze pro distribuce Linuxu, nikoliv pro OS Windows. Jeho výhodou je algoritmus „Zero-copy“. Algoritmus umožnuje rychlejší odesílaní tak, že nezatěžuje CPU kopírováním paketů z uživatelského prostoru do prostoru jádra a zpět. Stejně jako většina programů v Linuxu lze i Trafgenu přidávat velké množství parametrů, a tak ovlivnit jeho chování. Nejprve byl vytvořen paket sloužící k odesílání. Obsahuje pouze základní adresy potřebné pro odeslání, port číslo 7, který je určen pro ECHO a testovací sekvenci dat. Paket je vidět na obrázku 4.6. Pokud není definováno jinak, tak Trafgen odesílá neomezené množství paketů s využitím všech dostupných jader procesoru. Příkaz pro odesílaní vytvořeného paketu je na obrázku 4.7.

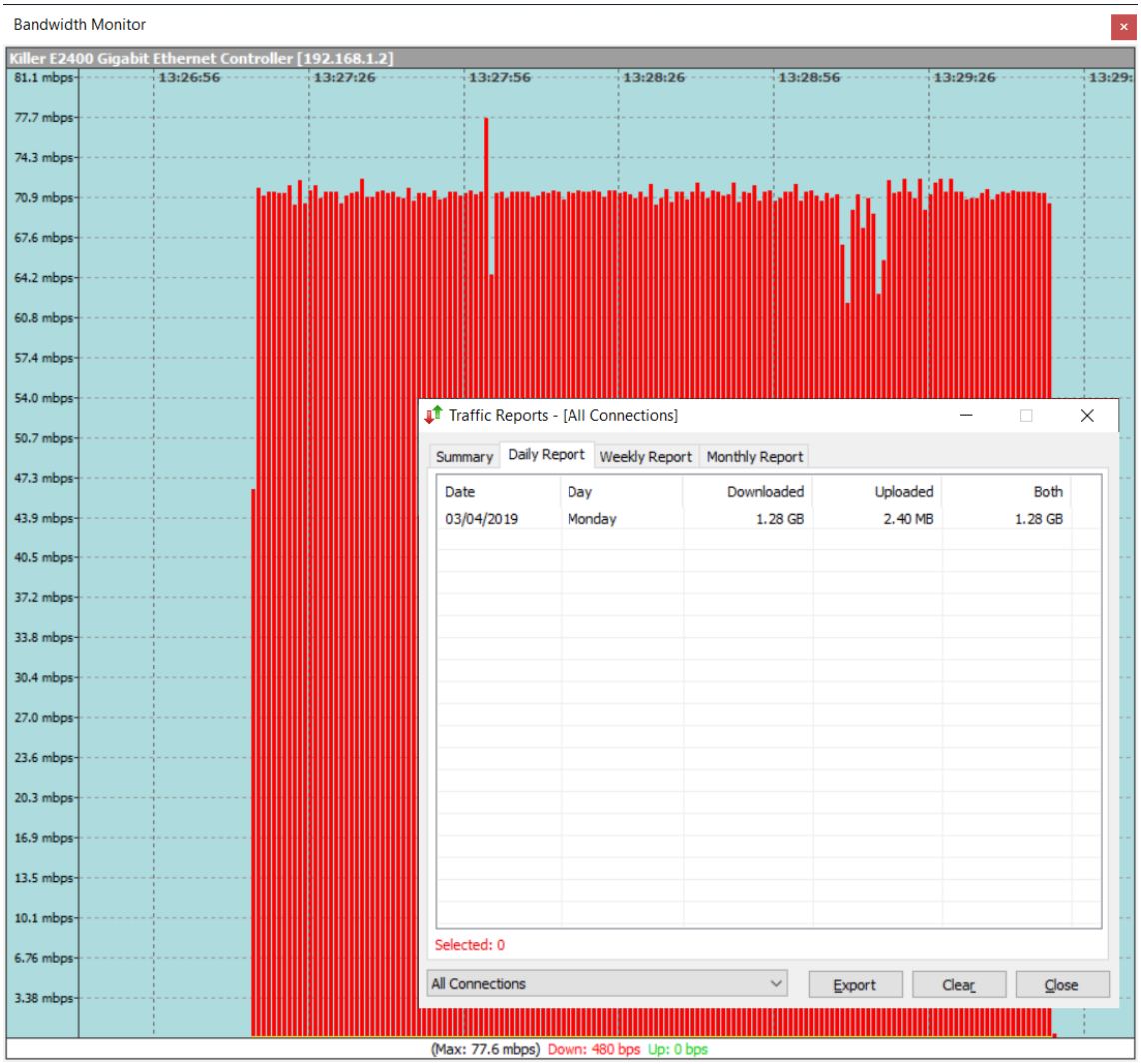


Obr. 4.6 UDP paket



Obr. 4.7 Trafgen

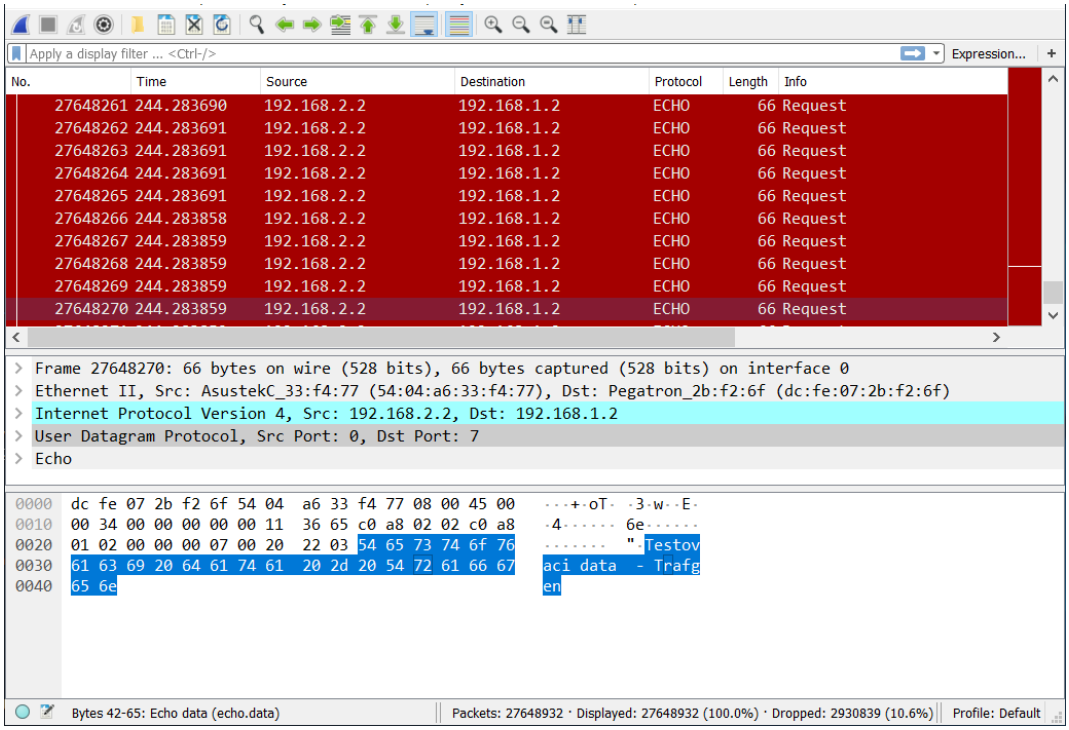
Ke generování byl využit stejný počítač jako v předchozím případě. Se čtyřjádrovým procesorem byl vygenerovaný tok několikrát vyšší než u předchozího generátoru NetScan Tools. Za 3 minuty se přeneslo 11216864 paketů. Datový tok se pohyboval kolem 71 Mb/s a za dvě a půl minuty bylo přeneseno 1,28 GB (obrázek 4.8). Stejně jako v případě předešlého útoku byl spuštěn příkaz ping (obrázek 4.9). Jsou vidět daleko větší odezvy než při prvním útoku, ale oběť nebyla úplně odříznuta od všech služeb. Stejně jako v předešlém případě byl i zde zachycen paket pomocí Wiresharku. Na obrázku 4.10 je vidět náhodně zvolený paket, jehož obsah se shoduje s obsahem paketu na obrázku 4.6.



Obr. 4.8 Vygenerovaný datový tok nástrojem Trafgen

```
C:\Users\Jakub>ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:
Reply from 192.168.2.2: bytes=32 time=1ms TTL=63
Reply from 192.168.2.2: bytes=32 time<1ms TTL=64
Reply from 192.168.2.2: bytes=32 time<1ms TTL=64
Reply from 192.168.2.2: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.2.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\Users\Jakub>ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:
Reply from 192.168.2.2: bytes=32 time=53ms TTL=64
Request timed out.
Request timed out.
Reply from 192.168.2.2: bytes=32 time=1041ms TTL=64

Ping statistics for 192.168.2.2:
    Packets: Sent = 4, Received = 2, Lost = 2 (50% loss),
Approximate round trip times in milli-seconds:
    Minimum = 53ms, Maximum = 1041ms, Average = 547ms
```

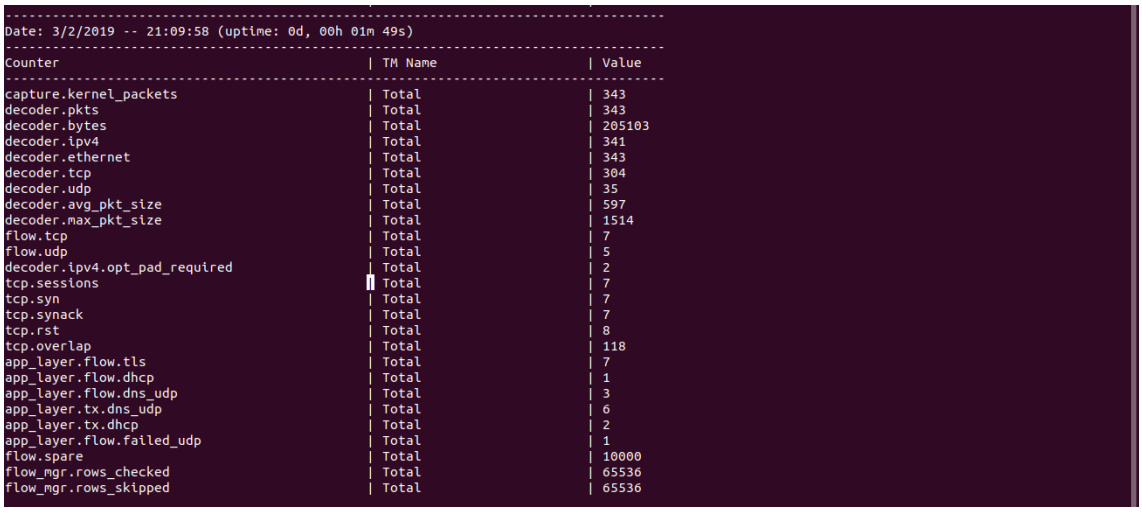
Obr. 4.9 Ping při druhém útoku



Obr. 4.10 Wireshark při druhém útoku

4.3 Suricata

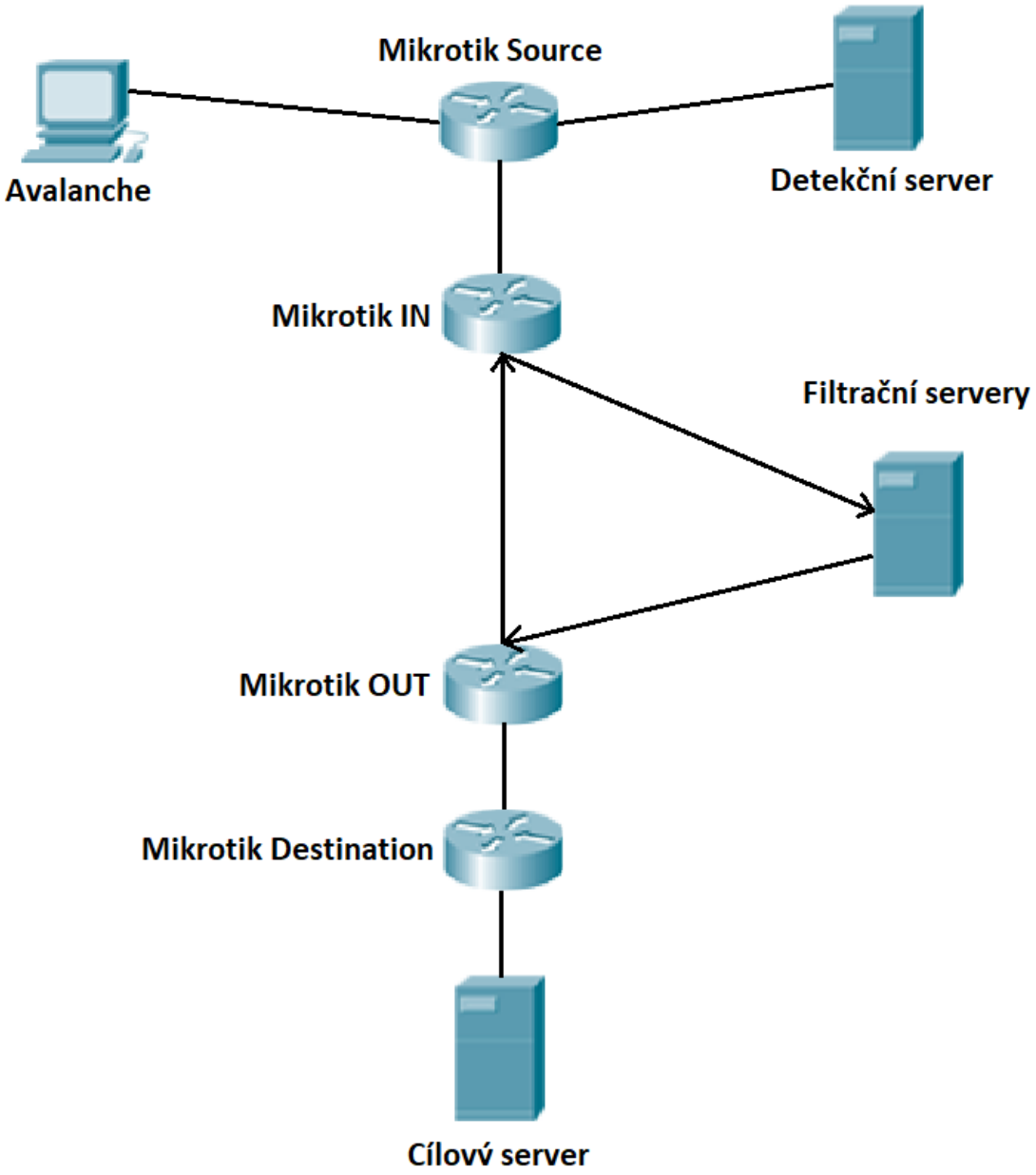
Program Suricata [37] je volně dostupný IDS/IPS pro OS Windows i Linux. Suricata nevyužívá grafické rozhraní, pro konfiguraci tohoto programu slouží konfigurační soubor surikata.yaml. Mimo jiné je v něm definováno, které soubory s pravidly budou, při spuštění programu, načteny. Tyto pravidla budou využita při kontrole datového toku. Samotná pravidla jsou v základní podobě distribuována společně s programem. Další pravidla je možné dodatečně stáhnout, nebo vytvořit a implementovat je do programu. Program jako IDS pouze zapisuje záznamy do logů (obrázek 4.11).



Obr. 4.11 Statistika zachycených paketů ze souboru stats.log

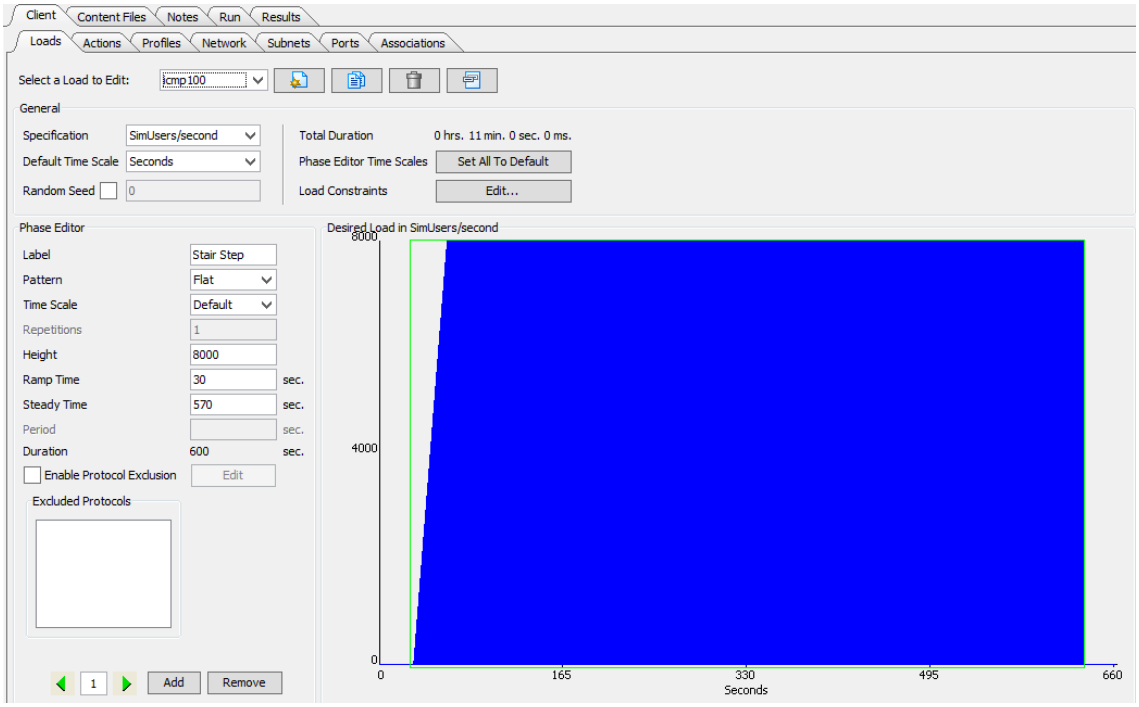
5 REALIZACE NA EXPERIMENTÁLNÍM PRACOVIŠTI VUT

K realizaci bakalářské práce bylo využito experimentální pracoviště (obrázek 5.1) na VUT, vytvořené k výzkumu a vývoji detekčních a filtračních technik k obraně proti DDoS útokům. Ke generování datového provozu již není využit počítač, ale specializovaný HW: Spirent Avalanche 3100B. Tento datový provoz je odesílán k cíli. Na cestě je veškerý datový provoz zrcadlen na detekční server, na kterém je IDPS Suricata. Veškerý provoz následně prochází přes filtrační servery k cíli útoku.

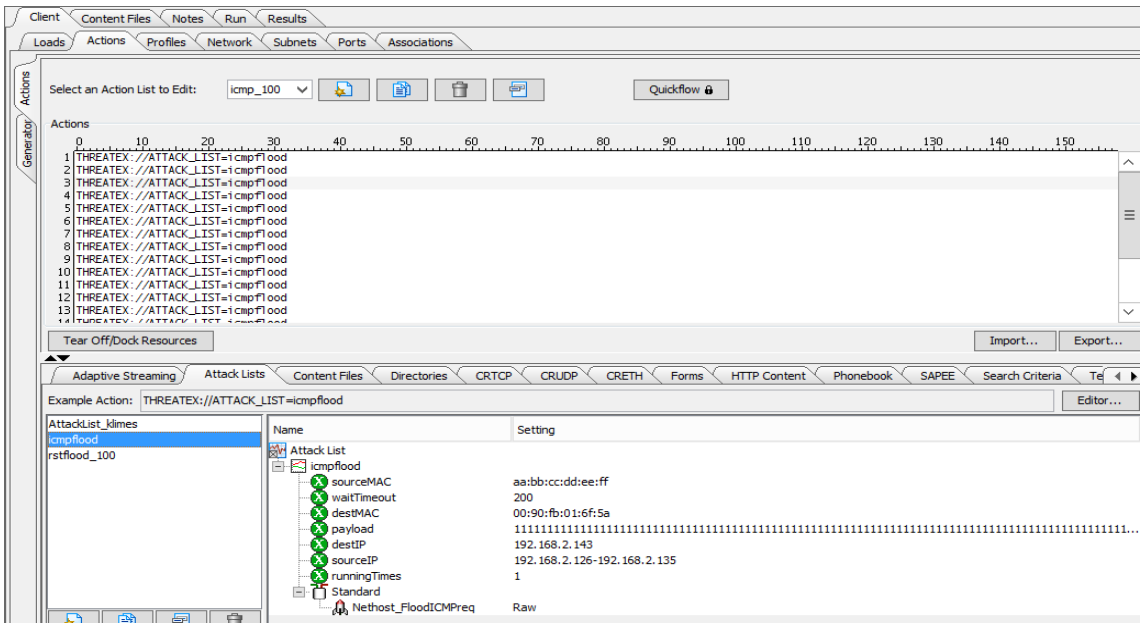


Obr. 5.1 Zjednodušené schéma pracoviště

K ovládání generátoru se využívá SW s názvem Spirent Avalanche Commander. Tímto rozhraním lze pomocí široké sady parametrů definovat požadovaný útok. Na obrázku 5.2 je vidět okno programu, ve kterém lze měnit parametry datového toku útoku. Nastavením parametru height lze měnit velikost datového toku. Dalšími parametry k nastaveni je například doba náběžné hrany, po kterou se bude datový tok stupňovat, dokud nedosáhne maximálního datového toku, nastaveného v parametru height. Dalším parametrem je steady time, který určuje čas, po který je útok prováděn s maximálním datovým tokem.



Obr. 5.2 Okno softwaru



Obr. 5.3 Okno pro nastavení parametrů obsahu

Na obrázku 5.3 je vidět nabídka programu, které slouží k nastavení obsahu paketů. Lze nastavit zdrojové a cílové adresy, port nebo obsah paketů. Na pracovišti byly předpřipraveny útoky ICMP flood a RST flood.

Detekce probíhá na detekčním serveru pomocí IDS Suricata. Při instalaci IDS jsou nainstalována i pravidla pro detekci nežádoucího provozu. Pravidla se musí nejprve povolit v konfiguračním souboru Suricaty. Z důvodu rychlejší detekce škodlivého provozu nejsou v souborech s pravidly povolena veškerá pravidla. Jednoduchou úpravou je možné povolit požadovaná pravidla. K detekci ICMP flood a RST flood byla vytvořena pravidla pomocí manuálových stránek k IDS [37].

K zamezení škod způsobených útokem je vhodné filtrovat tento útok co nejblíže u zdroje. Proto je k filtrování zvolen směrovač Mikrotik IN. K připojení ke směrovači je využit klasický telnet. Telnet byl zvolen pro testovací účely a samozřejmě je možno ho nahradit šifrovanou alternativou – SSH. Důvodem volby telnetu oproti SSH byla jeho jednoduchá implementace a případná možnost, odchytit a analyzovat data, které telnet odesílá nešifrovaně. Filtrace provozu je realizováno pomocí pravidel firewallu, jež je součástí směrovače. Filtrace je založena na zdrojových adresách uložených v seznamu adres. Tyto adresy jsou získány z logů (obrázek 5.4), které vytváří IDS na detekčním serveru, když dojde k porušení některého z pravidel. K tomuto byl vytvořen script v Bashi viz příloha.

```
03/18/2019-06:28:37.744888  [**] [1:1000001:3] ICMP zaplava [**] [Classification: (null)] [Priority: 3] {ICMP} 192.168.2.134:8 -> 192.168.2.142:0
03/18/2019-06:28:37.753727  [**] [1:1000001:3] ICMP zaplava [**] [Classification: (null)] [Priority: 3] {ICMP} 192.168.2.135:8 -> 192.168.2.142:0
03/18/2019-06:28:37.762699  [**] [1:1000001:3] ICMP zaplava [**] [Classification: (null)] [Priority: 3] {ICMP} 192.168.2.126:8 -> 192.168.2.142:0
03/18/2019-06:28:37.647869  [**] [1:1000001:3] ICMP zaplava [**] [Classification: (null)] [Priority: 3] {ICMP} 192.168.2.134:8 -> 192.168.2.142:0
03/18/2019-06:28:37.795634  [**] [1:1000001:3] ICMP zaplava [**] [Classification: (null)] [Priority: 3] {ICMP} 192.168.2.130:8 -> 192.168.2.142:0
03/18/2019-06:28:37.689384  [**] [1:1000001:3] ICMP zaplava [**] [Classification: (null)] [Priority: 3] {ICMP} 192.168.2.128:8 -> 192.168.2.142:0
03/18/2019-06:28:37.826752  [**] [1:1000001:3] ICMP zaplava [**] [Classification: (null)] [Priority: 3] {ICMP} 192.168.2.134:8 -> 192.168.2.142:0
```

Obr. 5.4 Výňatek z logu

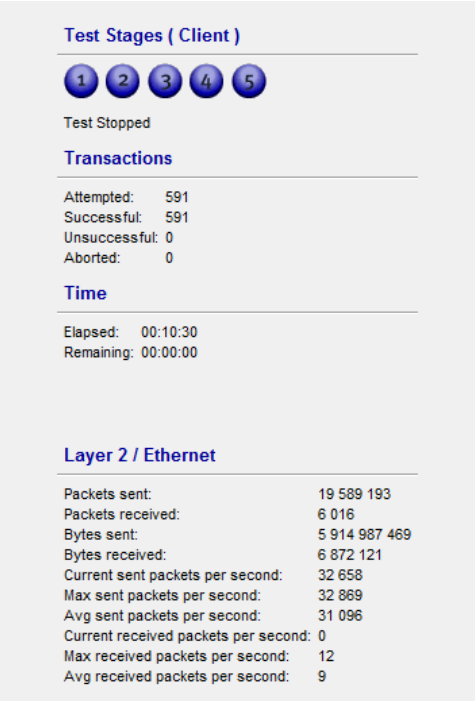
K tvorbě scriptu bylo využito rozšíření skriptovacího jazyka s názvem Expect [38]. Jedná se o rozšíření určené pro interakci s FTP, telnetem ale i jinými službami, které je možno automatizovat pro provádění z konzole. Expect dokáže simulovat uživatele zadávajícího příkazy z klávesnice. Jeho základním příkazem je expect“symboly“. Script naslouchá z konzole, jestli se objeví požadovaná sekvence symbolů. Jakmile se tak stane, provede se další příkaz, kterým může být například send“příkaz\r“. Příkaz send vypíše do konzole veškerý obsah nacházející se v uvozovkách. Aby došlo k jeho odeslání, musí být příkaz zakončen znakem \r.

```
spawn telnet 12.12.13.44
expect "Login:"
send "admin+t\r"
expect "Password:"
```

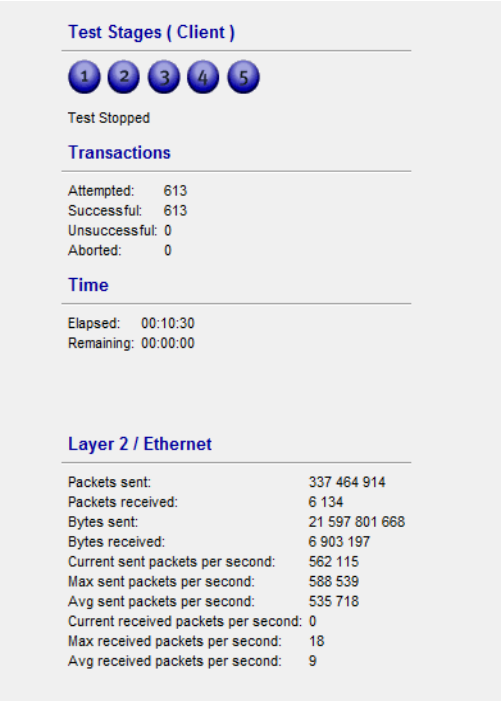
Obr. 5.5 Expect script

Na obrázku 5.5 je vidět část scriptu využívaná pro přihlášení k telnetu. Nejprve se započne relace telnetu. Následně script čeká, než se v konzoli objeví řetězec Login:, poté se odešle uživatelské jméno a obdobně se započne čekání na Password:. V některých případech se při přihlašování vypíší do konzole nechtěné znaky [39], které mohou narušit naslouchání řetězců z konzole. To je způsobeno tím, že se detekují možnosti terminálu. K zabránění tohoto jevu se přidává ke klasickému přihlašovacímu jménu ještě koncovka +t.

Byly provedeny útoky ICMP flood (obrázek 5.6) a RST flood (obrázek 5.7). Oba nejprve s datovým tokem o průměrné velikosti 79,09 Mb/s a následně RST flood s datovým tokem s průměrnou velikostí 294,1 Mb/s.



Obr. 5.6 ICMP flood

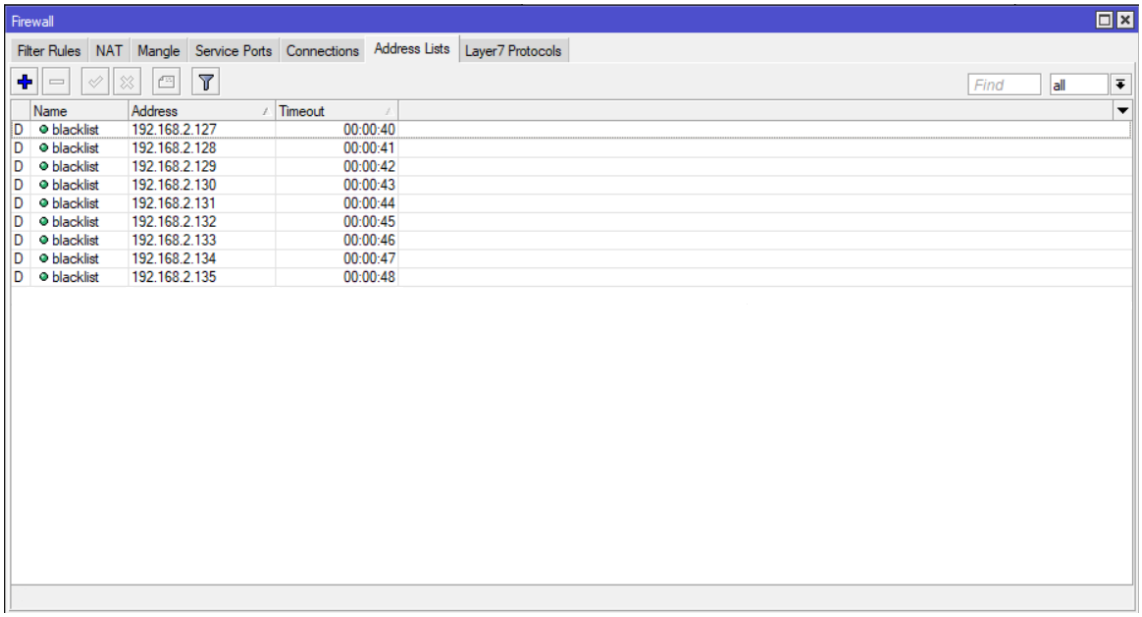


Obr. 5.7 RST flood s větším datovým tokem

Při spuštění scriptu byly zaznamenány zdrojové IP adresy a následně byly přidány do seznamu adres s názvem “blacklist“. Adresy se zdrojovou adresou z tohoto seznamu jsou podle vytvořeného pravidla ve firewallu zahazovány. Vlivem chyby se nebylo možno připojit pomocí dohledového systému k cílovému serveru, na který byl odesílán útočící datový tok. Proto nebylo možné získat odpovídající data o objemu dat útočícího toku, který na cíl přicházel a následně ani velikost datové tok po aktivaci filtru. Z tohoto důvodu nebylo možno otestovat efektivitu filtrování. Na obrázku 5.8 je vidět, jaké příkazy byly odeslány na základě adres, získaných od IDS. Na obrázku 5.9 je následně vidět seznam adres, které jsou filtrovány.

```
[admin@VT-AL44] > /ip firewall address-list
[admin@VT-AL44] /ip firewall address-list> add address=192.168.2.127 list=blacklist timeout=60s
[admin@VT-AL44] /ip firewall address-list> add address=192.168.2.128 list=blacklist timeout=60s
[admin@VT-AL44] /ip firewall address-list> add address=192.168.2.129 list=blacklist timeout=60s
[admin@VT-AL44] /ip firewall address-list> add address=192.168.2.130 list=blacklist timeout=60s
[admin@VT-AL44] /ip firewall address-list> add address=192.168.2.131 list=blacklist timeout=60s
[admin@VT-AL44] /ip firewall address-list> add address=192.168.2.132 list=blacklist timeout=60s
[admin@VT-AL44] /ip firewall address-list> add address=192.168.2.133 list=blacklist timeout=60s
[admin@VT-AL44] /ip firewall address-list> add address=192.168.2.134 list=blacklist timeout=60s
[admin@VT-AL44] /ip firewall address-list> add address=192.168.2.135 list=blacklist timeout=60s
[admin@VT-AL44] /ip firewall address-list>
```

Obr. 5.8 Script



Obr. 5.9 Seznam adres

6 ZÁVĚR

Cílem bakalářské práce byla realizace komunikace mezi systémem detekce a prevence průniku Suricata a přepínačem MikroTik. Dalším cílem bylo odfiltrovat probíhající útoky (ICMP flood a RST flood).

Na vlastním experimentálním pracovišti bylo provedeno měnění maximální hodnoty zatížení CPU v okamžiku přihlášení pomocí konzole, Winboxu a Webfigu. Po provedení dvaceti opakovaných připojení a odpojení byly hodnoty zprůměrovány. Nejnižších hodnot dosahovala konzole se svými 8,9 % (a srovnatelně Winbox se 10,9 %). Pokud by byl CPU směrovače silně vytížen, nemuselo by se vůbec podařit připojit. Z důvodu nízkého zatížení CPU a jednoduchosti byl konzolový přístup zvolen pro vytvoření scriptu.

Na vlastním experimentálním pracovišti byly rovněž testovány dva různé generátory DoS útoků, za účelem seznámení se s následky, jakých může útočící datový tok dosáhnout. Nejprve byt testován generátor pro OS Windows s názvem NetScan Tools. Tento generátor nevyužíval naplno potenciál CPU ke generování paketů, a proto vytvořený datový tok dosahoval pouze 16 Mb/s. I přesto docházelo při spuštění programu ping mezi obětí a útočníkem ke zvýšení odezvy a výpadkům. Následně byl testován generátor pro OS Linux (distribuce Ubuntu). Tento generátor dokázal využít CPU na maximum a vytvořit datový tok dosahující až 70 Mb/s. Takto velký datový tok měl za následek několikanásobné zvětšení odezvy pingu. Docházelo ke kolísání zpoždění od 53 ms do 1041 ms a jeho průměrná hodnota byla 547ms. Při prvním měření bylo změřeno zpoždění jako 1ms. Při útoku vzrostlo zpoždění 500x. Pořád se jedná o milisekundy, ale na jiné služby může mít toto zpoždění fatální dopad. Rovněž docházelo k častým výpadkům (přibližně polovina dotazů).

Bylo provedeno také testování na experimentální pracovišti VUT v Brně. Po instalaci neobsahuje IDPS Suricata pravidla pro detekci obou výše jmenovaných útoků. Proto musel být vytvořen a implementován nový soubor, obsahující nově pravidla specifická pro dané zadání.

Za účelem zprostředkování komunikace mezi IDPS Suricatou a zařízením od MikroTiku, byl vytvořen skript v Bashi. Pro komunikaci byl zvolen telnet. Jedná se o nejjednodušší způsob vzdáleného přístupu k zařízení a pro testovací účely je telnet plně dostačující. Zároveň umožnuje v případě potřeby odchytávat a analyzovat odeslaná data. K filtrování byl využit firewall směrovače. Tento již implementovaný mechanizmus směrovače umožnuje snadno definovat pravidla, podle kterých bude naloženo s příchozími pakety. S použitím scriptu došlo k úplnému odfiltrováni útoku nejblíže u jeho zdroje. Tato metoda filtrace co nejblíže u zdroje umožnuje minimalizovat nedostupnost služeb v dané síti. Škodlivý provoz musí být zahozen nebo odkloněn k zajištění dostupnosti služeb sítě.

Při testování scriptu došlo při všech provedených testech ke správnému získání zdrojových adres útočících zařízení a jejich zapsání do seznamu adres pro zablokování. V důsledku chyby v softwaru dohledového systému nebylo možno získat spolehlivé hodnoty vytížení CPU, respektive byly tyto hodnoty neustále nulové, a to jak v klidovém stavu, tak i při spuštění útoku. Tudíž nebylo možno otestovat efektivitu filtrace. Pokud by byl datový tok ještě větší a směrovač by byl plně vytížen, nebylo by již možné využít externího nástroje k přístupu ke směrovači a pomohlo by jen jeho odpojení od sítě.

Kritickým bodem řešení jsou pravidla, využívána k detekci útoků. Pokud je bude útočník znát, dokáže přizpůsobit svůj útok tak, aby nedošlo k jeho detekci. Proto je důležité udržovat pravidla v IDPS aktuální a stávající pravidla měnit nebo přidávat další na základě aktuálních trendů v útocích. Dalším možným krokem k posílením zabezpečení je přechod z telnetu na šifrované SSH.

LITERATURA

[1] MIRKOVIC, Jelena a Peter REIHER. A taxonomy of DDoS attack and DDoS defense mechanisms. ACM SIGCOMM Computer Communication Review. Duben 2004, 34(2), 39-53. DOI: 10.1145/997150.997156.

[2] PENG, Tao, Christopher LECKIE a Kotagiri RAMAMOHANARAO. Survey of network-based defense mechanisms countering the DoS and DDoS problems. ACM Computing Surveys (CSUR). Duben 2007, 39(1), Článek 3. DOI: 10.1145/1216370.1216373.

[3] BAŠTA, Pavel a Zuzana DURAČINSKÁ. DDoS – sofistikovaný útok nebo služba na objednávku?. IT SYSTEMS. 2015, 2015(4), 38-39.

[4] ASOSHEH, ABBASS a NAGHMEH RAMEZANI. A Comprehensive Taxonomy of DDoS Attacks and Defense Mechanism Applying in a Smart Classification. WSEAS Transactions on Computers. Duben 2008, 7(4), 281-290. ISSN 1109-2750.

[5] ROUSE, Margaret a Brien POSEY. Computer exploit. TechTarget [online]. Newton, 2017 [cit. 2018-10-06]. Dostupné z: https://searchsecurity.techtarget.com/definition/exploit

[6] PHILLIPS, Gavin. What Is a Botnet and Is Your Computer Part of One?. MakeUseOf [online]. Sheung Wan: MakeUseOf, 2018 [cit. 2018-10-07]. Dostupné z: https://www.makeuseof.com/tag/what-is-botnet/

[7] Pierluigi Paganini and Odysseus. Linux/Mirai ELF, when malware is recycled could be still dangerous [online]. Security Affair, Říjen 5, 2016 [cit. 2018-10-07]. Dostupné z: http://securityaffairs.co/wordpress/50929/malware/linux-mirai-elf.html

[8] REDDY, Rajini a Siva SUBRAMANIYAM. Techniques to Differentiate DDOS Attacks from Flash Crowd. International Journal of Advanced Research in Computer Science and Software Engineering. Červen 2013, 3(6), 295-299. ISSN 2277 128X.

[9] COCHRAN, Jaime. The WireX Botnet: How Industry Collaboration Disrupted a DDoS Attack. Cloudflare [online]. 2017 [cit. 2018-10-09]. Dostupné z: https://blog.cloudflare.com/the-wirex-botnet/

[10] ABLIZ, MEHMUD. Internet Denial of Service Attacks and Defense Mechanisms. Pittsburgh, květen 2011. Technická zpráva. Univerzita Pittsburgh.

[11] KRIŠOVÁ, Zdeňka a Jiří MARTINŮ. POČÍTAČOVÉ SÍTĚ. Moravská vysoká škola Olomouc, 2017. Dostupné také z: https://mvso.cz/wp-content/uploads/2018/02/Počítačové-sítě-studijní-text.pdf

[12] DDoS Quick Guide. US-CERT: United States [online]. Washington: National Cybersecurity and Communications Integration Center, 2014 [cit. 2018-10-9]. Dostupné z: https://www.us-cert.gov/sites/default/files/publications/DDoS%20Quick%20Guide.pdf

[13] DROST, Lars. A DDoS Security Control Framework. Amsterdam, 2015. Postgraduální práce. Vrije Univerzita Amsterdam. Vedoucí práce Paul Harnzeb.

[14] ČMELÍK, Martin. Seznamte se – DoS a DDoS útoky. Security-Portal.cz [online]. Praha, 2013 [cit. 2018-10-11]. Dostupné z: https://www.security-portal.cz/clanky/seznamte-se-%E2%80%93-dos-ddos-%C3%BAtoky

[15] POSTEL, J., "Internet Control Message Protocol", [online]. STD 5, RFC 792, DOI 10.17487/RFC0792, September 1981, [cit. 2018-10-16]. Dostupné z: https://www.rfc-editor.org/info/rfc792

[16] HALLER, Martin Denial of Service (DoS) útoky: záplavové typy. Lupa.cz [online]. CZ: Internet Info, 2006 [cit. 2018-10-16]. Dostupné z: https://www.lupa.cz/clanky/denial-of-

service-dos-utoky-zaplavove-typy/?ic=serial-box&icc=text-title

[17] 1996 CERT Advisories. Carnegie Mellon University: Software Engineering Institute [online]. Pittsburgh: CERT Division, 1996 [cit. 2018-10-16]. Dostupné z: https://resources.sei.cmu.edu/library/asset-view.cfm?assetID=496170

[18] UDP Flood Attack. Cloudflare [online]. Cloudflare [cit. 2018-10-17]. Dostupné z: https://www.cloudflare.com/learning/ddos/udp-flood-ddos-attack/

[19] Denial of Service (DoS) útoky: typy využívající chyb a vyčerpání systémových prostředků (2.). Lupa.cz [online]. CZ: Internet Info, 2006 [cit. 2018-10-17]. Dostupné z: https://www.lupa.cz/clanky/typy-vyuzivajici-chyb-a-vycerpani-systemovych-prostredku-2/?ic=serial-box&icc=text-title

[20] HALLER, Martin Denial of Service útoky: reflektivní a zesilující typy. Lupa.cz [online]. CZ: Internet Info, 2006 [cit. 2018-10-21]. Dostupné z: https://www.lupa.cz/clanky/denial-of-service-utoky-reflektivni-a-zesilujici-typy/

[21] ČMELÍK, Martin. Obrana před útokem DRDoS. Security-Portal.cz [online]. Praha, 2004 [cit. 2018-10-21]. Dostupné z: https://www.security-portal.cz/clanky/obrana-před-útokem-drdos

[22] BUKAČ, Vít. Small scale denial of service attacks. Brno, 2015. Dizertační práce. Masarykova Univerzita.

[23] HALLER, Martin. Denial of Service (DoS) útoky: typy využívající chyb a vyčerpání systémových prostředků (1.). Lupa.cz [online]. Praha: Internet Info, 2006 [cit. 2018-10-21]. Dostupné z: https://www.lupa.cz/clanky/typy-vyuzivajici-chyb-a-vycerpani-systemovych-prostredku-1/

[24] LEWIS, Jonathan. DDoS Attacks on SSL: Something Old, Something New. Netscout [online]. NA: ARBOR NETWORKS, 2012 [cit. 2018-10-22]. Dostupné z: https://asert.arbornetworks.com/ddos-attacks-on-ssl-something-old-something-new/

[25] SIKORA, Marek a Petr BLAŽEK. Systém prevence průniku Slow HTTP DoS a DDoS útoků. Elektrorevue. 2017, 19(4), 110-118. ISSN 1213-1539.

[26] KHANDELWAL, Swati. Memcached Servers Abused for Massive Amplification DDoS Attacks. The Hacker News Logo [online]. The Hacker News, 2018 [cit. 2018-10-23]. Dostupné z: https://thehackernews.com/2018/02/memcached-amplification-ddos.html

[27] HTTP Flood Attack. Cloudflare [online]. Cloudflare [cit. 2018-10-23]. Dostupné z: https://www.cloudflare.com/learning/ddos/http-flood-ddos-attack/

[28] K. S. Bhosale, M. Nenova a G. Iliev, "The distributed denial of service attacks (DDoS) prevention mechanisms on application layer," 2017 13th International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS), Nis, 2017, 136-139. doi: 10.1109/TELSKS.2017.8246247

[29] Mukhopadhyay, Debajyoti & Oh, Byung-Jun & Shim, Sang-Heon & Kim, Young-Chon. (2010). A Study on Recent Approaches in Handling DDoS Attacks.

[30] David G. Andersen, Hari Balakrishnan, Nick Feamster, Teemu Koponen, Daekyeong Moon, and Scott Shenker. 2008. Accountable internet protocol (aip). In Proceedings of the ACM SIGCOMM 2008 conference on Data communication (SIGCOMM '08). ACM, New York, NY, USA, 339-350. DOI: 10.1145/1402958.1402997

[31] ROUSE, Margaret. Intrusion detection system (IDS). TechTarget [online]. Newton, 2018 [cit. 2018-10-28]. Dostupné z: https://searchsecurity.techtarget.com/definition/intrusion-detection-system

[32] LASEK, Petr. Intrusion prevention system: Nová kategorie bezpečnostního řešení. IT Systems. 2005, 6(1-2), 62-63.

[33] About us. MikroTik [online]. [cit. 2019-03-05]. Dostupné z: https://mikrotik.com/aboutus

[34] Kolektiv autorů Kompletní manuálové stránky společnosti Mikrotik [online]. 2015, poslední aktualizace 20. 04. 2018 [cit. 18. 11. 2018]. Dostupné z URL: http://wiki.mikrotik.com/wiki/Manual:TOC

[35] NetScanTools® Pro. NetScanTools.com [online]. 2019 [cit. 2019-03-05]. Dostupné z: https://www.netscantools.com/nstpromain.html

[36] Netsniff-ng toolkit. Netsniff-ng [online]. [cit. 2019-03-05]. Dostupné z: http://netsniff-ng.org/

[37] Suricata User Guide. Suricata [online]. [cit. 2019-04-07]. Dostupné z: https://suricata.readthedocs.io/en/suricata-4.1.2/index.html

[38] Expect. Tcl/Tk Development [online]. [cit. 2019-04-08]. Dostupné z: https://core.tcl.tk/expect/index

[39] Manual:Console login process. MikroTik Wiki [online]. [cit. 2019-04-08]. Dostupné z: https://wiki.mikrotik.com/wiki/Manual:Console_login_process#FAQ

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

AD	Accountable Domains
AIP	Accountable Internet Protocol
ARP	Address Resolution Protocol
AS	Autonomous Systém
BGP	Border Gateway Protocol
C&C	Command and Control
CBQ	Class-Based Queuing
CPU	Central Processor Unit
DDoS	Distributed Denial of Services
DDoSaaS	DDoS-as-a-Service
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name Systém
DoS	Denial of Services
DRDoS	Distributed Reflektor Denial of Service
EID	Endpoint Idenfiticator
FTP	File Transfer Protocol
HIDS	Host-based Intrusion Detection Systems
HTTP	Hypertext Transfer Protocol
HW	Hardware
ICMP	Internet Control Message Protocol
IDPS	Intrusion detection and prevention systém
IDS	Intrusion Detection Systems
IPD	Intrusion Prevention Systém
IRC	Internet Realy Chat
MAC	Media Access Control
MB	Megabajty
Mb/s	Megabity za sekundu
ms	Milisekundy
NIDS	Network Intrusion Detection Systems
NTP	Network Time Protocol
OS	Operační systém

POP3	Post Office Protocol 3
RAM	Random Access Memory
SAVE	Source Address Validity Enforcement
SMTP	Simple Mail Transfer Protocol
SNTP	Simple Network Time Protocol
SOS	Secure Overlay Service
SSH	Secure Shell
SSL	Secure Sockets Layer
SW	Software
TCP	Transmission Control Protocol
UDP	User Datagram Protocol

SEZNAM PŘÍLOH

A Příloha 59
A.1 Pravidla iptables..……..……59
A.2 Obsah přiloženého CD…….…… 59

A PŘÍLOHA

A.1 Pravidla ve firewallu

Pravidla používaná firewallem směrovače.

alert icmp any -> $HOME_NET any (msg:"ICMP zaplava"; itype:8; threshold: type threshold, track by_dst, count 10, seconds 60; sid:1000001; rev:3;)

alert tcp any -> $HOME_NET 80 (msg:"RST zaplava"; flags: R; threshold: type threshold, track by_dst, count 10, seconds 60; sid:1000002; rev:2;)

A.2 Obsah přiloženého CD

Na přiloženém CD je umístěna elektronická verze bakalářské práce, všechny použité obrázky, soubor s pravidly pro Suricatu a vytvořený script.