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

OBRANA PŘED VOLUMETRICKÝMI DDOS ÚTOKY V PROSTŘEDÍ SDN
MITIGATION OF VOLUMETRIC DDOS ATTACKS IN SDN ENVIRONMENT

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. Vojtěch Hodes

VEDOUCÍ PRÁCE
SUPERVISOR

doc. Ing. Vladislav Škorpil, CSc.

BRNO 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**
Ústav telekomunikací

Student: Bc. Vojtěch Hodes ***ID:*** 151640
Ročník: 2 ***Akademický rok:*** 2016/17

NÁZEV TÉMATU:

Obrana před volumetrickými DDoS útoky v prostředí SDN

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s technologií NetFlow/IPFIX, řešením Flowmon, technologií softwarově definovaných sítí (SDN) a standardem OpenFlow ve verzi 1.3. Popište metody monitorování a detekce volumetrických útoků v prostředí páteřních datových sítí. Navrhněte techniky obrany před těmito útoky s využitím moderního konceptu SDN umožňujících dynamicky řídit datový tok. Výše uvedené koncepty ověřte v praxi s využitím řešení Flowmon pro monitorování provozu datových sítí a detekci útoků typu DDoS. Následující praktická část práce bude zahrnovat ověření možnosti řídit a omezovat datový tok prostřednictvím protokolu OpenFlow v prostředí SDN. Dosažené výsledky budou demonstrovány prakticky v laboratorním prostředí. Dále vytvořte laboratorní prostředí, kde je možné efektivně útoky typu DDoS simulovat včetně konfigurace pro SDN přepínač ve standardu OpenFlow verze 1.3 pro eliminaci DDoS útoku typu UDP flood.

DOPORUČENÁ LITERATURA:

[1] Flowmon Networks a.s. [online]. Brno: Flowmon Networks a.s., ©2016 [cit. 2016-09-13]. Dostupné z: www.flowmon.com

[2] Open Networking Foundation [online]. Kalifornie: Open Networking Foundation, ©2016 [cit. 2016-09-13]. Dostupné z: www.opennetworking.org

Termín zadání: 1.2.2017 ***Termín odevzdání:*** 24.5.2017

Vedoucí práce: doc. Ing. Vladislav Škorpil, CSc.
Konzultant: RNDr. Pavel Minařík, Ph.D.

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

ABSTRAKT

Cílem diplomové práce je prostudovat různé přístupy a navrhnout koncepty monitorování a detekce volumetrických DDoS útoků v prostředí páteřních sítí. Je v ní pojednáno o protokolech řízení datových toků s důrazem na nejmodernější technologii softwarově definovaných sítí. Závěrečná část zahrnuje ověření získaných teoretických poznatků v podobě sestavení vlastního laboratorního prostředí, simulaci volumetrického DDoS útoku typu UDP Flood a automatizovanou obranu před ním.

KLÍČOVÁ SLOVA

Detekce volumetrických DDoS útoků, obrana, monitorování provozu, NetFlow, IPFIX, SDN, OpenFlow, Flowmon

ABSTRACT

The aim of this Master‘s thesis is to explore different attitudes and to design various monitoring and detection concepts of volumetric DDoS attacks in core networks. The thesis deals with data flow control protocols with an emphasis on a modern technology of Software Defined Networks. The last part of the thesis describes verification of the theory by setting up a laboratory environment for volumetric DDoS UDP Flood simulation, detection and automated mitigation.

KEY WORDS

Volumetric DDoS detection, mitigation, traffic monitoring, NetFlow, IPFIX, SDN, OpenFlow, Flowmon

HODES, V. *Obrana před volumetrickými DDoS útoky v prostředí SDN.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2017. 51 s. Vedoucí diplomové práce doc. Ing. Vladislav Škorpil, CSc.

Prohlášení

Prohlašuji, že svou diplomovou práci na téma „Obrana před volumetrickými DDoS útoky v prostředí SDN“ jsem vypracoval samostatně pod vedením vedoucího diplomové 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 diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové 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 10.5.2017 .....

podpis autora

SIX research centre
sensor, information and communication systems

Faculty of Electrical Engineering
and Communication

Brno University of Technology
Technicka 12, CZ-61600 Brno, Czechia

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

Výzkum popsaný v této diplomové práci byl realizovaný v laboratořích podpořených projektem Centrum senzorických, informačních a komunikačních systémů (SIX); registrační číslo CZ.1.05/2.1.00/03.0072, operačního programu Výzkum a vývoj pro inovace.

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

2007–13
OP Výzkum a vývoj pro inovace

Poděkování

Děkuji vedoucímu práce doc. Ing. Vladislavu Škorpilovi, CSc. a konzultantu práce RNDr. Pavlu Minaříkovi, Ph.D. za velmi užitečnou metodickou pomoc a potřebné rady při zpracování diplomové práce. Rovněž bych chtěl poděkovat za cenné konzultace pracovnímu kolektivu společnosti Flowmon Networks a.s., zvláště pak Jankovi Pekl'anskému a Ing. Jiřímu Knapkovi.

V Brně dne 10.5.2017 .....

podpis autora

Obsah

Úvod

Volumetrické útoky jsou asi ty nejfrekventovanější distribuované útoky odepření služeb, kterým poskytovatelé připojení a služeb ve svých páteřních datových sítích musí čím dál tím častěji čelit. S rozvojem síťových technologií a infrastruktury tyto útoky nabývají na četnosti a intenzitě. Dopady úspěšně vedených útoků mohou být velmi závažné, a to v dnešní době především ekonomické. Jako příklad lze jistě uvést nedostupnost internetových obchodů, internetových bankovnictví až po výpadek klíčových aplikací velkých nadnárodních korporací. Tato semestrální práce si klade za cíl prostudovat různé přístupy a navrhnout koncepty monitorování a detekce volumetrických DDoS útoků v prostředí páteřních sítí. Bude pojednáno o protokolech řízení datových toků s důrazem na nejmodernější technologii softwarově definovaných sítí. Závěrečná praktická část bude zahrnovat ověření teoretických poznatků sestavením laboratoře pro simulaci útoku typu UDP Flood. Dále pak jeho detekci a obranu před ním.

1 Metody monitorování a detekce volumetrických DDoS útoků

V této kapitole budou prozkoumány a navrženy možné metody monitorování a detekce volumetrických DDoS útoků v prostředí páteřních datových sítí a jejich vzájemné srovnání. Tyto metody jsou monitorování pomocí SNMP protokolu, záchyt celého provozu a monitorování datových toků.

1.1 Monitorování pomocí SNMP protokolu

SNMP je protokol sedmé, tedy aplikační, vrstvy ISO/OSI síťového modelu definovaný nad nespolehlivým transportním protokolem UDP. SNMP protokol umožňuje monitoring mnoha částí síťových prvků. Těmito částmi může být vytížení procesoru systému, operační paměti, vytížení síťových rozhraní aj. V současnosti SNMP protokol existuje ve třech standardy: SNMP, SNMPv2, SNMPv3.

Protokol SNMP definuje dvě části. Část monitorovanou (tzv. Agent), typicky několik síťových zařízení, a část monitorující (tzv. NMS). Dle rozsahu a složitosti síťové infrastruktury může být monitorujících prvků více. Obecně však platí, že NMS spravuje více agentů.

Princip SNMP monitoringu spočívá v tom, že NMS se periodicky dotazuje na sledované statistiky na straně agentů. Agenti na tyto dotazy odpovídají zprávami s obsahem požadovaných statistik. Výjimkou je pak odpověď typu SNMP Trap. Zde totiž agent s odpovědí nečeká na dotaz NMS a posílá ji ihned, jakmile je splněna dříve nakonfigurovaná podmínka na straně agenta. [1] Jako příklad může být uvedeno překročení definovaného prahu vytížení operační paměti. To může mít např. u mnoha serverových systémů založených na platformě Linux/UNIX za následek vynucené ukončování (zabíjení) kritických procesů.

Pro účely monitorování a detekce volumetrických DDoS útoků lze SNMP protokol s výhodou využít pro sledování volumetrických statistik indikujících možný útok. Tento koncept počítá s konfigurací SNMP takovým způsobem, že na straně agentů je nakonfigurováno zasílání SNMP Trap zpráv v případě vytížení síťových rozhraní nad definovanou očekávanou hranici. Určení této hranice pak závisí na uvážení síťového administrátora na základě informací z dlouhodobého SNMP monitoringu. Současně tento koncept monitoringu počítá s SNMP monitoringem ze strany monitorovacího prvku NMS, který se bude periodicky dotazovat agentů na vytížení jejich procesorových jednotek a operační paměti. Volumetrický distribuovaný útok odepření služeb se totiž typicky vyznačuje zvýšeným nárokem na výpočetní prostředky aktivního prvku. SNMP dotazování ze strany NMS navíc umožňuje zjištění nedostupnosti síťového prvku a následnou detekci DDoS útoku v případě, že pod jeho vlivem není možné odeslat zprávu SNMP Trap. To může být typicky způsobeno díky tomu, že primárním účelem aktivního prvku je přepínat pakety v síti, SNMP monitoring má tedy daleko nižší prioritu. Dalším důvodem může být úplný pád aktivního prvku pod vlivem DDoS útoku.

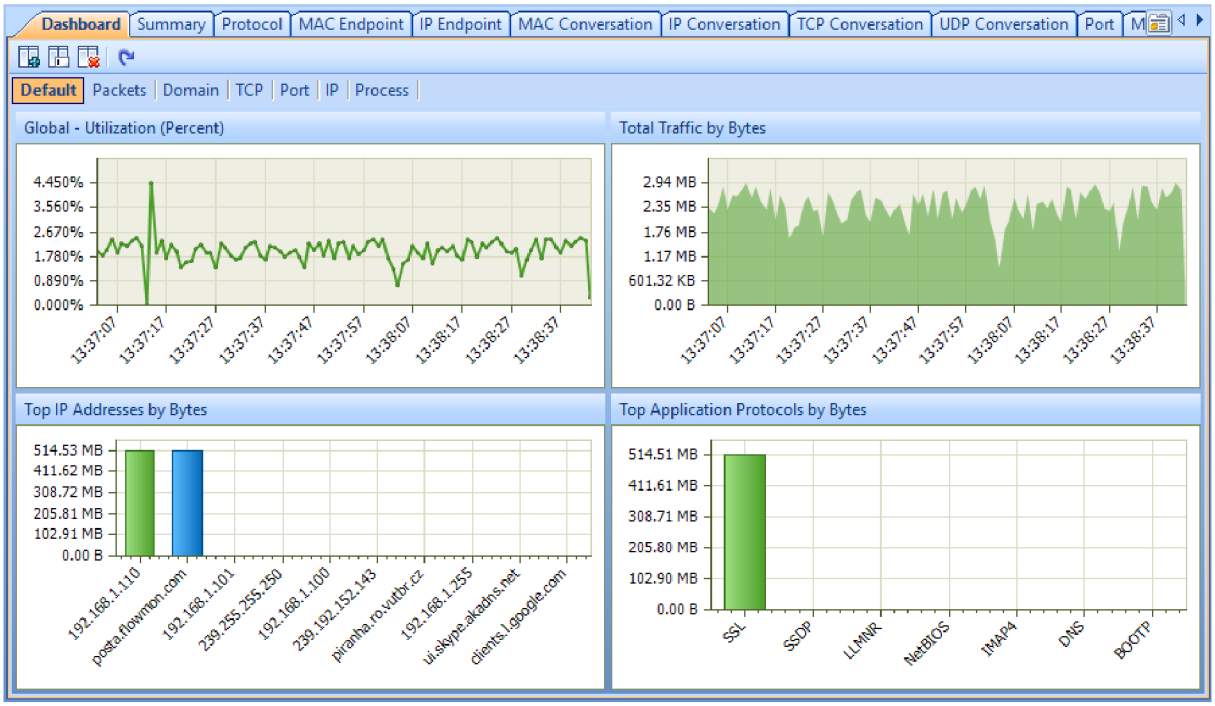
Z výše uvedených metod monitorování a detekce volumetrických DDoS útoků je technologie SNMP monitoringu asi nejstarším, tedy spolehlivým, široce rozšířeným a jednoduchým nástrojem. Umožňuje však získání pouze základních statistik. Kromě vytíženosti jednotlivých částí aktivních prvků a volumetrických statistik o útoku nelze zjistit bližší informace z hlaviček nebo obsahu paketů.

1.2 Záchyt celého provozu (Full Packet Capturing)

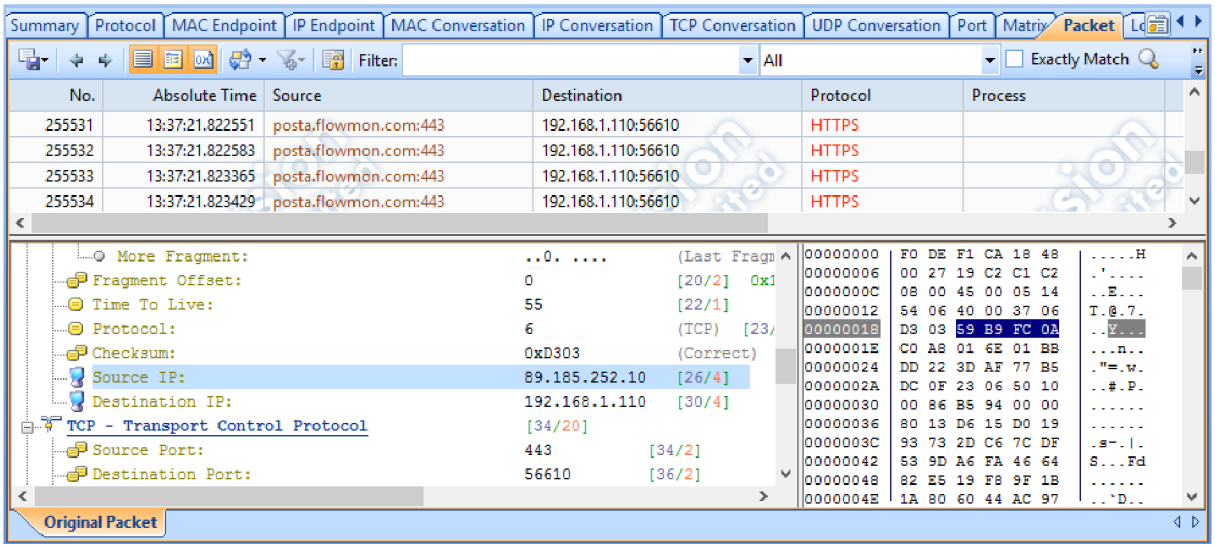
Další možností, jak monitorovat a detekovat volumetrické distribuované útoky odepření služeb, je záchyt veškerého provozu na síti, tedy každého paketu. Principiálně se jedná velmi jednoduchý přístup.

Možný scénář počítá s nasazením zařízení s paketovým analyzátorem schopného v reálném čase tímto způsobem monitorovat provoz.

Jedním z nástrojů, které lze použít pro paketovou analýzu, je nástroj Capsa společnosti Colasoft. Jedná se o velmi sofistikovaný nástroj umožňující zachycení a analýzu všech paketů se zobrazením mnoha pohledů na monitorovaný provoz v reálném čase. Mezi tyto pohledy např. patří vykreslování přehledných grafů jednak s volumetrickými informacemi o monitorovaném provozu (počty paketů, bytů za sekundu), ale rovněž i informace o nejvyužívanějších IP adresách a protokolech z hlediska provozu. Ukázka uživatelského prostředí paketového analyzátoru Capsa je na obr. 1.1 a 1.2.



Obr. 1.1: Analýza síťového provozu v reálném čase nástrojem Capsa



Obr. 1.2: Paketová analýza v reálném čase v nástroji Capsa

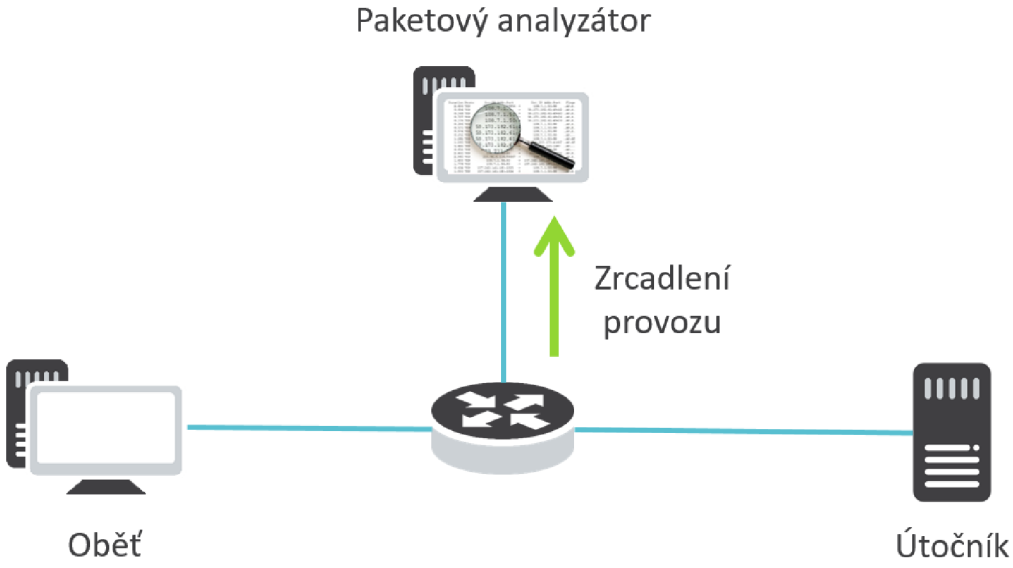
Uvedený koncept monitorování a detekce volumetrických DDoS útoků dokáže díky zachycení celých paketů síťovému administrátorovi poskytnout v reálném čase veškeré informace až do sedmé vrstvy síťového modelu ISO/OSI. Lze tedy jednoznačně odhalit odkud a kam je útok veden, na kterých službách je veden, o jaký transportní protokol se jedná atp. K analýze tohoto provozu lze použít více či méně automatizované nástroje. Nevýhodou je však zejména při volumetrických DDoS útocích velká náročnost na výpočetní prostředky analyzátoru. Důvodem k tomu je velké množství provozu, které podléhá detailní paketové analýze. Dochází totiž i ke zkoumání datové části paketů, která pro detekci volumetrických DDoS útoků není příliš relevantní.

Dalším významným aspektem je potřeba uložení každého paketu na pevný disk analyzátoru. Pokud bychom uvažovali nad volumetrickým DDoS útokem o velikosti 10 Gb za sekundu, je potřeba na pevný disk uložit 1,25 GB za sekundu, což je při standardních rychlostech magnetických disků kolem 65 MB za sekundu obtížné. Řešením tohoto problému by mohl být systém s nasazením rychlých SSD disků společně s technologií RAID. Tímto by však kvůli vysoké ceně SSD technologie došlo k velmi významným zvýšením nákladů na celý monitorovací a detekční systém.

Aby však bylo možné paketovým analyzátorem provoz analyzovat, je ze všeho nejdříve nutné nějakým způsobem k tomuto analyzátoru monitorovaný provoz přivést.

1.2.1 Zrcadlení provozu

Obecný koncept tohoto typu monitoringu je uveden na obr. 1.3. Spočívá ve vhodné konfiguraci tzv. mirror portu na centrálním aktivním prvku, na který dochází ke kopírování (zrcadlení, mirroringu) žádaného monitorovaného provozu. Tento provoz je pak přiváděn na síťové rozhraní analyzujícího zařízení. Lze tak tedy jednoduše monitorovat veškeré dění v síti a rychle detekovat zvýšené přenosy indikující volumetrické DDoS útoky.



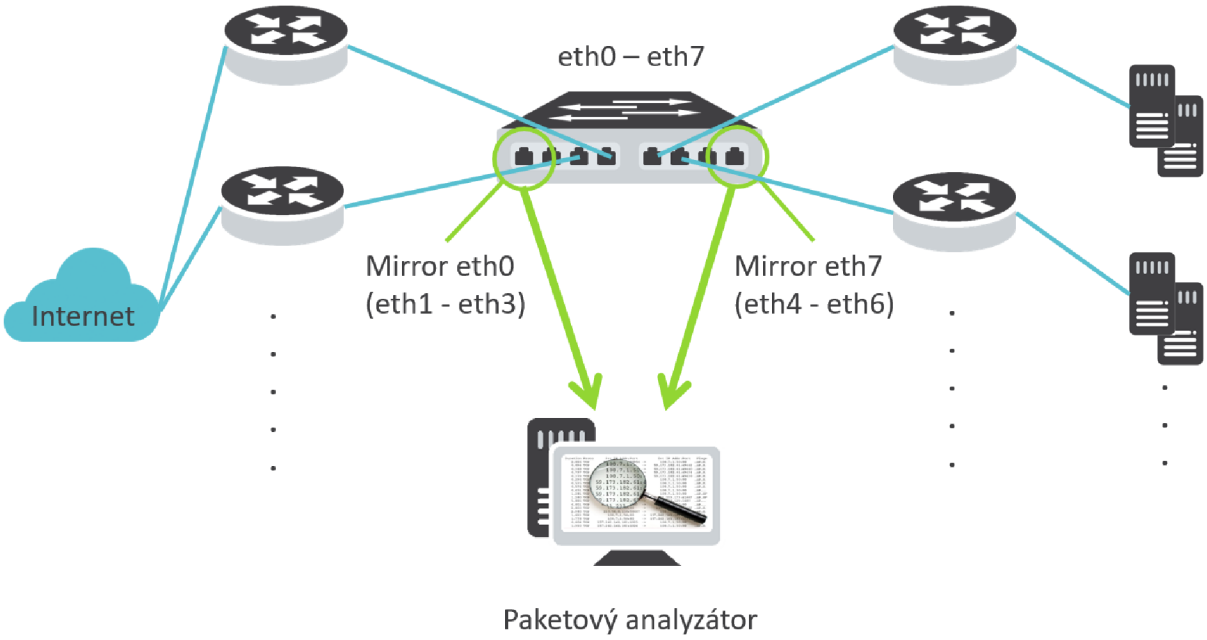
Obr. 1.3: Koncept síťového monitoringu analýzou celého provozu (paketů)

V prostředí páteřních datových sítí však toto řešení může být problematické. Pokud uvážíme, že v infrastruktuře poskytovatelů připojení se běžně můžeme setkat s datovými přenosy v řádu stovek

gigabit za sekundu, je zřejmé že vytvoření mirror portu, který by zrcadlil takové velké množství dat, je prakticky nereálné, neboť celkový replikovaný provoz by jistě překročil propustnost mirror portu na aktivním prvku. Špičkově technologicky vybavené aktivní prvky totiž v současnosti zpravidla disponují síťovými rozhraními o propustnosti do sta gigabitů za sekundu. Řešením by mohlo být vytvoření několika takovýchto mirror portů, mezi něž by byl provoz rozdělován na základě nějakého pravidla.

První možností by mohla být konfigurace každého z mirror portů takovým způsobem, že by zrcadlil provoz jen z určité části portů aktivního prvku, jimiž prochází monitorovaný datový provoz. Na obr. 1.4 je zachycena tato situace, kdy na centrálním přepínači s celkem osmi síťovými rozhraními (eth0 – eth7), který je umístěn v páteřní infrastruktuře poskytovatele připojení, je zrcadlení provozu nakonfigurováno tak, že mirror port eth0 zrcadlí veškerý provoz z hraničních směrovačů v příchozím směru, tedy porty eth1 – eth3.

Druhý mirror port eth7 je nakonfigurován totožně s tím rozdílem, že zrcadlí provoz pro porty eth4 – eth6, na které jsou připojeny směrovače k pomyslným koncovým zákazníkům. Výše zmíněné mirror porty jsou pak přímo připojeny do dedikovaného zařízení s minimálně dvěma síťovými rozhraními, na němž je spuštěn paketový analyzátor.



Obr. 1.4: Zrcadlení provozu pomocí dvou mirror portů

Obr. 1.5 popisuje jednoduchý příklad nastavení mirror portů pro případ uvedený na obr. 1.4. Jedná se o ukázku z přepínače Catalyst 4500 společnosti Cisco. Nejprve je v příkladu specifikováno, které porty budou zdrojem pro zrcadlení, v terminologii operačního systému přepínače se jedná o source porty. Takto nakonfigurované porty jsou eth1 – eth3 a eth4 – eth6. Po provedení této konfigurace bude jejich provoz kopírován do mirror portu, v terminologii operačního systému přepínače se jedná o destination port nebo porty. Z uvedeného příkladu je zřejmé, že pro zrcadlení provozu jsou v rámci přepínače použity jednotlivé, na sobě nezávislé, instance zrcadlení, tzv. monitor session s příslušným pořadovým číslem. Počet těchto instancí bývá společností Cisco zpravidla licenčně omezen. V našem příkladu jsou použity instance dvě, pro každý mirror port jedna.

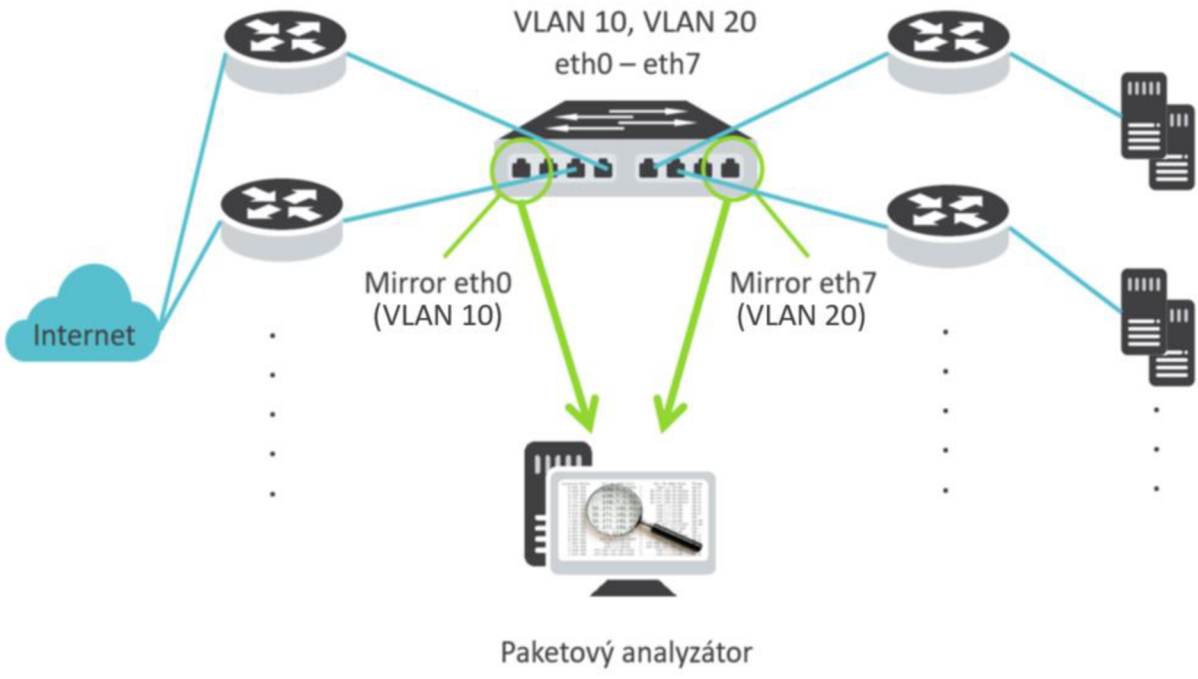
```
# specifikace portů, které budou kopírovat svůj provoz
Switch(config)# monitor session 1 source interface fastethernet 1/1-3
Switch(config)# monitor session 2 source interface fastethernet 1/4-6

# přiřazení portů výše jednotlivým mirror portům
Switch(config)# monitor session 1 destination interface fastethernet 1/4 encapsulation replicate
Switch(config)# monitor session 2 destination interface fastethernet 1/7 encapsulation replicate
```

Obr 1.5: Nastavení zrcadlení provozu využitím zrcadlení jednotlivých portů

Posledním konfiguračním krokem je specifikace samotného mirror portu, do něhož bude kopírován veškerý provoz z vybraných portů přepínače, a to formou přiřazení monitor session příslušnému portu. Pro mirror port eth0 jsou to porty eth1 – eth3 spadající pod monitor session s pořadovým číslem 1 a pro mirror port eth7 to jsou porty eth4 – eth6 spadající pod monitor session s pořadovým číslem 2. Součástí konfiguračního příkazu je i definice, jestli daný mirror port ponechá všem paketům jejich VLAN značku (VLAN tag) nebo naopak tuto značku z rámců odebere. Pro účely síťového monitoringu je samozřejmě žádoucí, aby monitorovaný provoz zůstal nezměněn a tím poskytnul maximálně autentické informace o dění v síti. Je proto nutné v příslušném příkazu uvést „encapsulation replicate“. V opačném případě by všechny rámce byly zrcadleny bez VLAN značky.

Druhým řešením problematiky možnosti zahlcení mirror portu extrémními přenosy by mohl být ve své podstatě totožný koncept s tím rozdílem, že by nedocházelo k zrcadlení provozu jednotlivých portů páteřního síťového prvku, nýbrž bychom s výhodou mohli využít toho, že celkový provoz je často logicky dělen do VLAN. Při konfiguraci mirror portů lze totiž specifikovat zrcadlení jednotlivých VLAN. Je samozřejmě potřeba vzít v úvahu provoz v jednotlivých VLAN ve vztahu k maximální propustnosti portů, které budou provádět mirroring. Na obr. 1.6 je příklad takovéto situace. Na páteřním přepínači je port mirroring nastaven takovým způsobem, že port eth0 zrcadlí veškerý provoz spadající do VLAN 10, eth7 pak veškerý provoz spadající do VLAN 20.



Obr. 1.6: Zrcadlení provozu pomocí členění na VLAN

Nastavení port mirroringu pro případ mirroringu jednotlivých VLAN staví na stejných principech, jak bylo popsáno výše. Jediný rozdíl spočívá v tom, že jako zdroj mirroringu jsou nastaveny jednotlivé VLAN, nikoli rozsahy portů. Mirror port eth4 tedy bude zrcadlit veškerý provoz z VLAN 10 a mirror port eth7 pak veškerý provoz spadající pod VLAN 20. Příklad, který tuto konfiguraci ilustruje, je uveden na obr. 1.7.

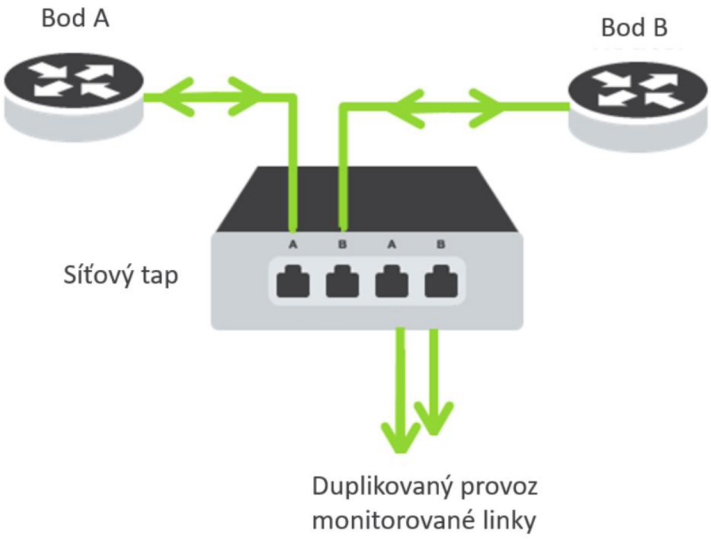
```
# specifikace VLAN, které budou kopírovat svůj provoz
Switch(config)# monitor session 1 source vlan 10
Switch(config)# monitor session 2 source vlan 20

# přiřazení VLAN výše jednotlivým mirror portům
Switch(config)# monitor session 1 destination interface fastethernet 1/4 encapsulation replicate
Switch(config)# monitor session 2 destination interface fastethernet 1/7 encapsulation replicate
```

Obr. 1.7: Nastavení zrcadlení provozu využitím zrcadlení jednotlivých VLAN

1.2.2 Použití síťových tapů

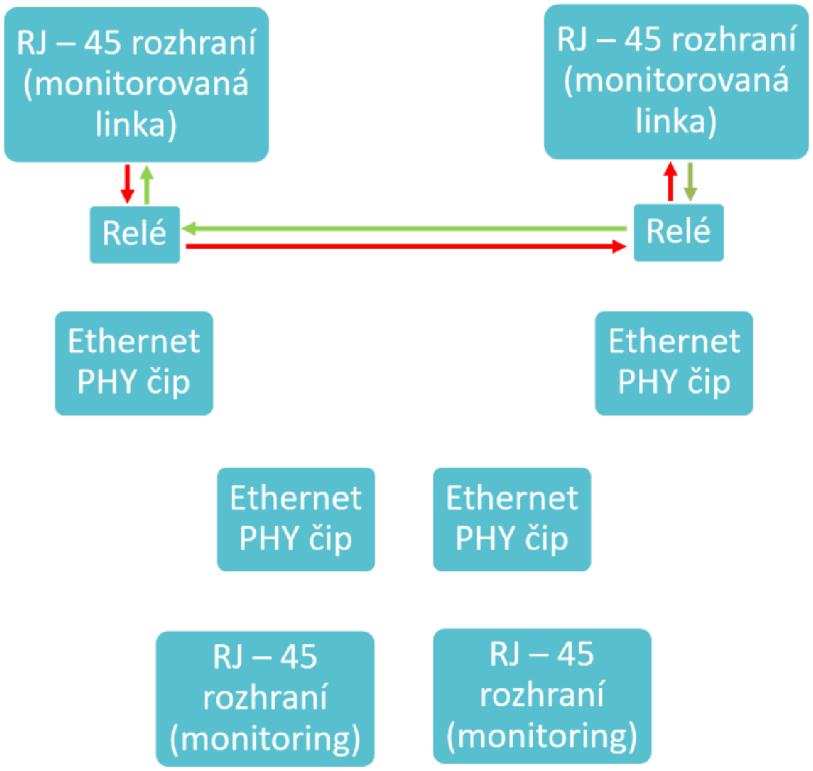
Jinou možností, jak dosáhnout zrcadlení síťového provozu pro účely síťového monitoringu, je použití tzv. síťových tapů. Jejich funkce by se dala analogicky přiblížit štěnicím pro odposlech. Základní princip těchto tapů je značně jednoduchý. V podstatě se jedná o „napíchnutí" sledované síťové linky, kdy pak tap kopíruje data monitorované linky na své porty určené pro monitoring. Provoz na sledované lince pak není nijak narušen a běžně pokračuje. Z obr. 1.8 je zřejmé to, že pokud budeme chtít monitorovat jednu plně duplexní metalickou linku mezi bodem A a bodem B, budeme muset fyzickou vrstvu této linky rozšířit o jeden kabel navíc. Z principu technologie tapu navíc vyplývá, že použití tapu pro monitoring jedné plně duplexní linky rozdělí oba její směry provozu.



Obr. 1.8: Monitoring plně duplexní metalické linky použitím síťového tapu

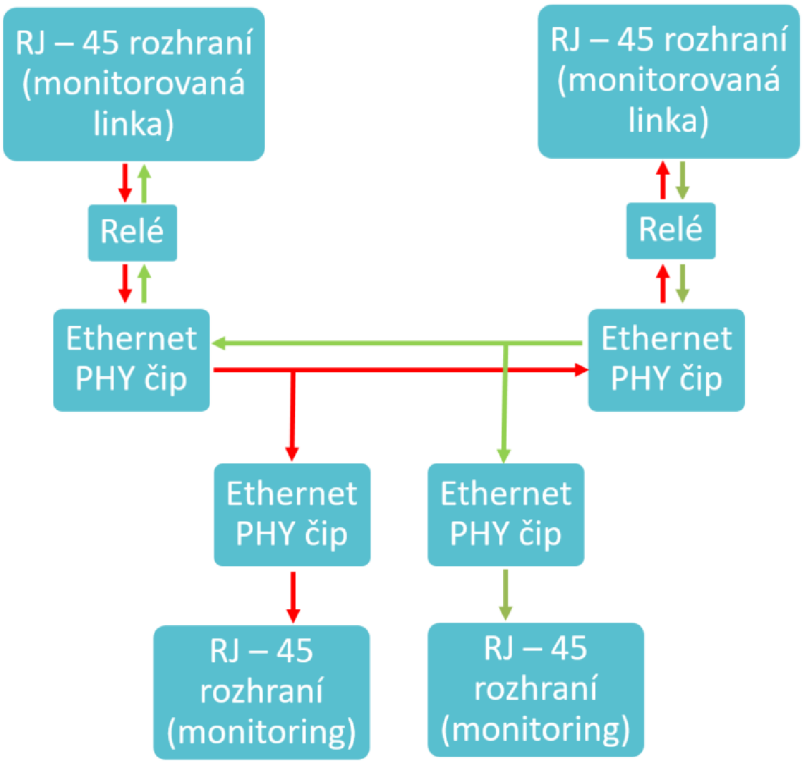
Na obr. 1.9 je zobrazeno vnitřní blokové schéma tapu přizpůsobeného monitorování dnes běžně používaných linek 1000BASE-T. Blok označený jako „RJ – 45 rozhraní (monitorovaná linka)" slouží k připojení síťových kabelů monitorované linky. Dále následují dva bloky plnící funkci jednoduchého elektromagnetického relé. Bloky „Ethernet PHY čip" jsou integrované obvody zabezpečující přijímání a odesílání ethernetových rámců. Poslední bloky s RJ – 45 přípojkami pak slouží k zapojení kabelů, které dále odvádějí do monitorovacího zařízení monitorovaný provoz.

Obr. 1.9 současně zachycuje situaci, kdy síťový tap vybaven bloky elektromagnetických relé není připojen k elektrické síti, tento typ tapu tedy v tomto smyslu není pasivní. V uvedeném případě nedojde k sepnutí jednotlivých relé v tapu a tap jednoduše předává veškerý provoz z jednoho RJ – 45 rozhraní určeného pro monitorovanou linku na druhé. Plní tedy propojovací funkci linky fyzické vrstvy. Na síťová rozhraní tapu, která jsou určena pro odvádění monitorovaných dat do síťového analyzátoru, nejsou kopírována žádná data. Odposlech dat je tedy neaktivní.



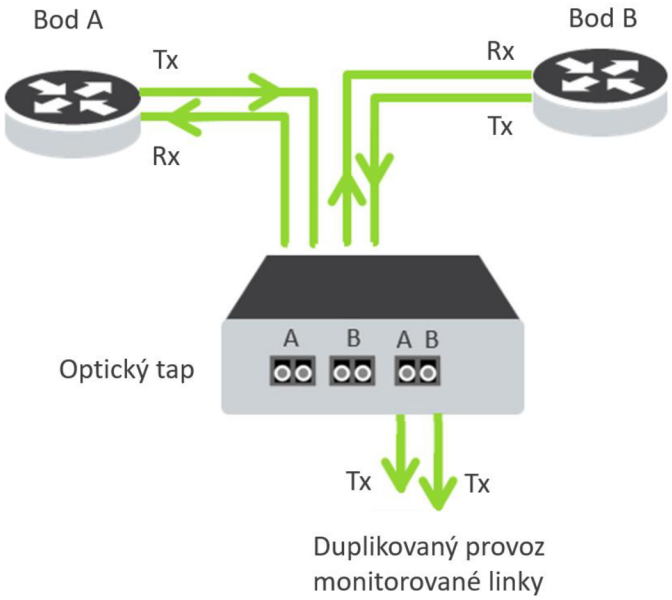
Obr. 1.9: Blokové schéma tapu pro monitoring 1000BASE-T linky ve stavu bez napětí

Pokud však dojde k připojení tapu do elektrické sítě, jednotlivá relé sepnou obvod k příslušným ethernet PHY čipům a dochází jednak k předávání provozu z jednoho RJ – 45 rozhraní určené pro monitorovanou linku na druhé a jednak i k odposlechu a kopírování dat na RJ – 45 rozhraní na cestě k síťovému analyzátoru. Z obr. 1.10 je zřejmé, že síťový tap rozděluje monitorovaný provoz takovým způsobem, že jeden směr provozu monitorované linky kopíruje na jedno RJ – 45 rozhraní směřující k síťovému analyzátoru a druhý směr na druhé. Tyto dva směry provozu jsou na obr. 1.10 rozlišeny různými barvami. [2]



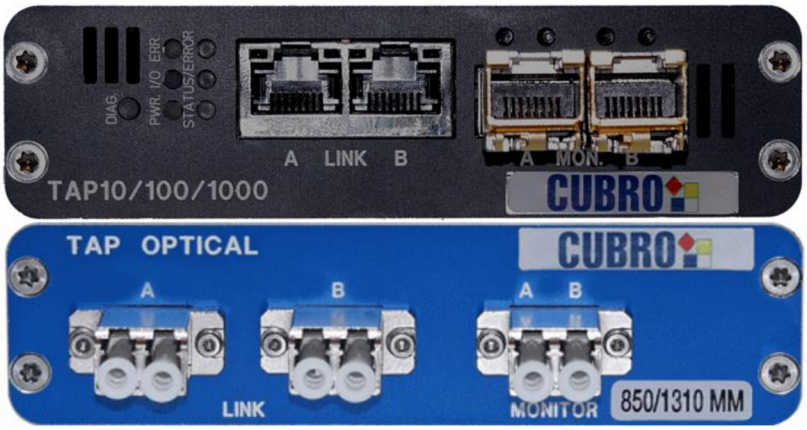
Obr. 1.10: Blokové schéma tapu pro monitoring 1000BASE-T linky ve stavu pod napětím

Pro případ monitorování počítačových sítí využívajících technologii optického přenosu dat je možné použít i optickou variantu síťového tapu. Princip monitoringu jedné plně duplexní optické linky je velmi podobný metalické variantě, viz obr. 1.11.



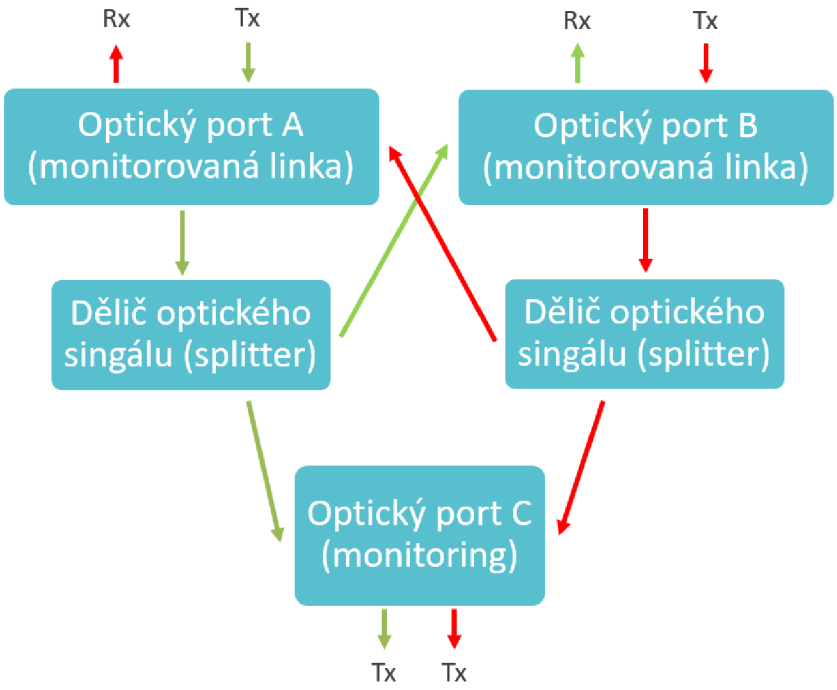
Obr. 1.11: Monitorování jedné plně duplexní optické linky použitím optického síťového tapu

Vlivem toho, že běžná optická linka je složena ze dvou vláken, jednoho pro příjem (Tx) a druhého pro odesílání dat (Rx), je potřeba přizpůsobit i konstrukci optického tapu. Běžný optický tap disponuje v porovnání s tapem pro monitoring optických linek jen třemi porty pro příjem a odesílání dat. Obr. 1.12 ukazuje srovnání přední strany obou jmenovaných tapů. V horní části je zachycen metalický tap, v dolní pak optický.



Obr 1.12: Srovnání čelní strany optického a metalického tapu [3]

Princip funkce síťových tapů pro monitoring optických linek je o něco jednodušší, než je tomu u metalických tapů. Optický tap je totiž zcela pasivní, využívá jednoduchého dělení optického signálu a není tedy třeba u něj uvažovat připojení do elektrické sítě. Stejně jako u metalického tapu je potřeba pouze připojit monitorovanou optickou linku do příslušných optických portů. Optický signál z každého portu je následně rozdělen pomocí děliče optického signálu (splitteru). Část energie optického signálu tedy pokračuje do další části monitorované linky, část je pak odkloněna do optického portu určeného pro monitoring a odvedena do analyzátoru síťového provozu. Tento technologický přístup je znázorněn na Obr. 1.13.



Obr. 1.13 - Blokové schéma síťového tapu pro monitoring optické linky

Z výše uvedeného je zřejmé, že při použití tapů pro monitoring optických sítí je nutné počítat s útlumem optického signálu na monitorované lince vlivem jeho dělení ve splitteru. Optické tapy se vyrábějí v mnoha variantách s dělícím poměrem pohybujícím se v rozmezí od 50/50 po 90/10.

Ve srovnání s metalickými tapy jsou optické technologicky jednodušší, neobsahují žádné elektrické nebo mechanické části a jsou tedy robustnější a velmi spolehlivé. Při použití metalických tapů zase naopak není potřeba uvažovat zvýšený útlum signálu na monitorované lince.

Z výše uvedených přístupů přivedení monitorovaného provozu na síťový analyzátor se jako nejlepší varianta jednoznačně jeví použití síťových tapů. Je to dáno tím, že tapy lze prakticky nasadit v jakémkoli místě sledované sítě a monitorovat celou kapacitu monitorované linky. Nehrozí zde tedy zahlcení mirror portů a potažmo celého aktivního prvku, jenž v sítí zastává daleko kritičtější úlohu, kterou je směrování a přepínání datových jednotek. Nevýhodu tohoto přístupu však může představovat nákladnost takovéhoto řešení. Do sítě je totiž potřeba dodatečný hardware v podobě tapů a alokovat dostatečné množství síťových rozhraní na síťovém analyzátoru.

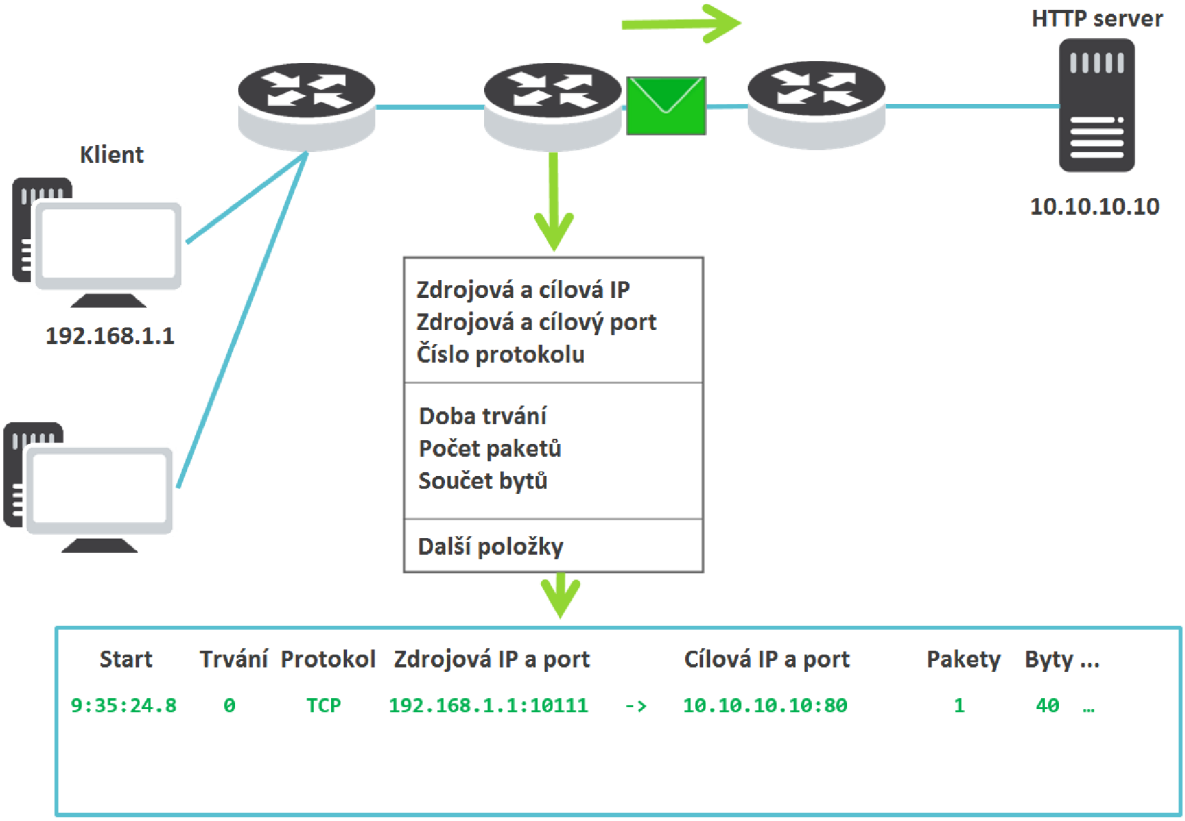
1.3 Monitorování datových toků (Flow Monitoring)

Princip monitorování datových toků spočívá v tom, že síťový prvek zkoumá pouze hlavičku paketu. Informace, které jsou z hlavičky paketu exportérem aktivního prvku parsovány, závisí na použitém standardu monitorování datových toků. Mezi základní informace, které zpravidla všechny standardy monitorují, patří zdrojová a cílová IP adresa, zdrojový a cílový port, TCP příznaky, číslo protokolu a velikost paketu. Tab 1.1 stručně popisuje některé nejrozšířenějších standardy pro monitorování datových toků.

Tab. 1.1: Popis některých standardů monitorování datových toků

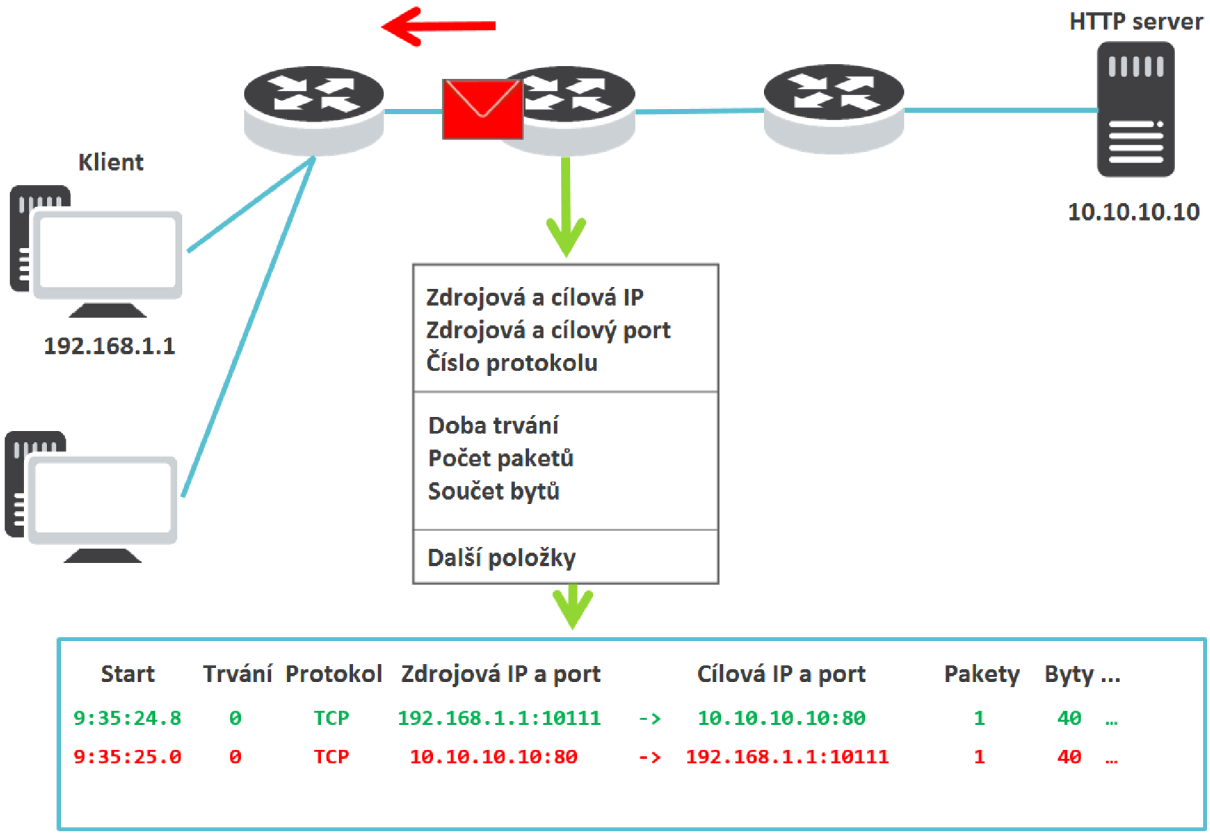
Název standardu	Popis standardu
NetFlow v5	Vyvinut společností Cisco Jen základní pole (zdrojová a cílová IP a port, číslo protokolu, ToS pole,... Neobsahuje informace 2. vrstvy ISO/OSI modelu (MAC, VLAN)
Netflow v9 (Flexibilní NetFlow)	Flexibilní rozšířený formát Informace o IPv6, VLAN, MAC,... Asi nejrozšířenější
IPFIX	Standard s otevřeným zdrojovým kódem Trend v monitoringu datových toků Možnost inspekce až do 7. vrstvy ISO/OSI
NetStream	Vyvinut společností Huawei Téměř totožný s NetFlow v9
jFlow	Vyvinut společností Juniper Podobný NetFlow v9
sFlow (Sampled Flow)	Odvozený standard Vzorkované NetFlow/IPFIX na úrovni datových toků
NEL (Network Event Logging)	Vyvinut společností Cisco Odvozený standard Informace o průběhu procesu NAT
NSEL (Network Secure Event Logging)	Vyvinut společností Cisco Odvozený standard Informace o procesu firewallingu

Jakmile paket dorazí na síťové rozhraní prvku podporujícího technologii exportu datových toků, je porovnáván s tabulkou již existujících datových toků, kterou směrovač po určitou dobu udržuje v paměti. Pokud výše zmíněné informace v hlavičce paketu odpovídají již existujícímu toku, je paket k tomuto toku asociován a dojde pouze k úpravě statistik toku jako je počet přenesených bytů, paketů a délka trvání toku. Pokud se však informace v hlavičce paketu s žádným již existujícím tokem neshodují, je vytvořen tok nový, kde opět v závislosti na použitém standardu monitorování datových toků jsou typicky zaznamenány následující informace: časová známka začátku datového toku, délka trvání toku, zdrojová a cílová IP adresa, zdrojový a cílový port, použitý transportní protokol a agregovaný počet paketů a bytů. Vytvoření a zaznamenání datového toku při pokusu o navázání TCP spojení (three-way handshake) ilustruje obr. 1.14.



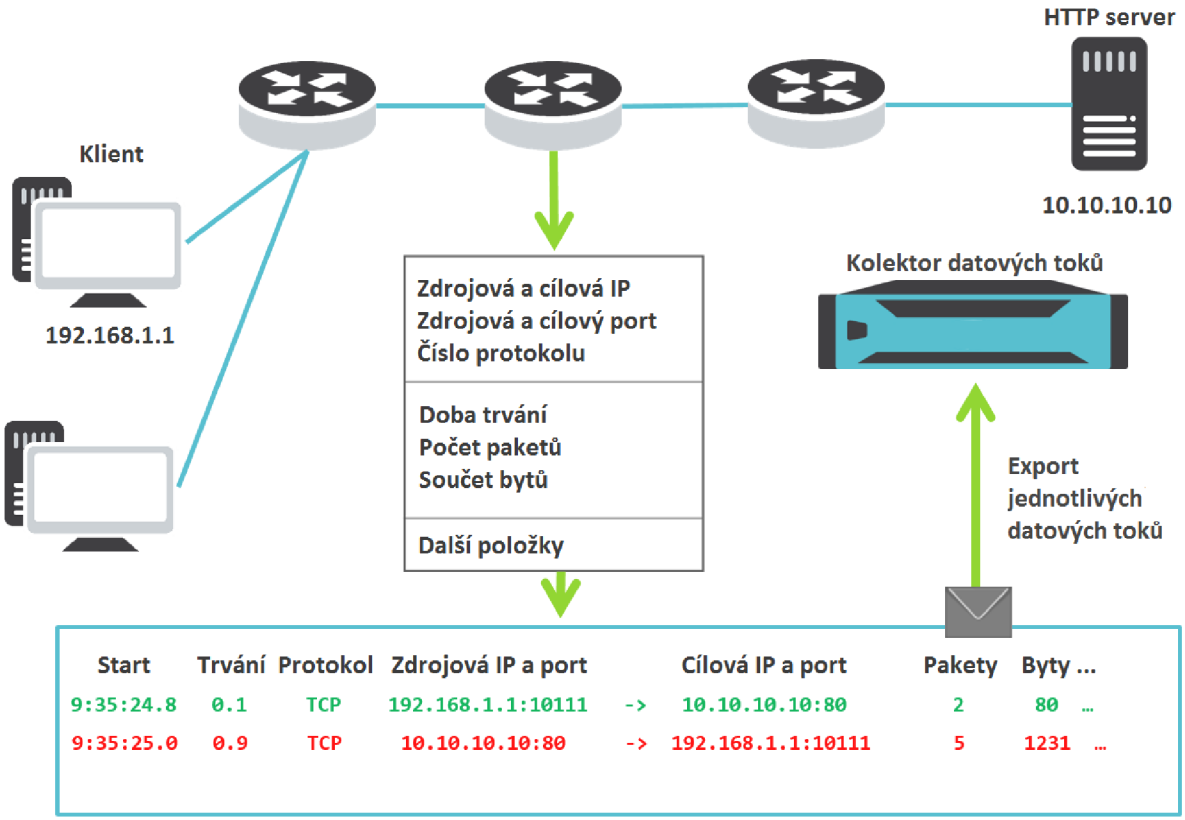
Obr. 1.14: Navázání spojení v cestě klient-server

Pokus o navázání TCP spojení ze strany serveru a následné vytvoření dalšího datového toku je pak znázorněno na obr. 1.15. Z obr. 1.14 a 1.15 je tedy zřejmé, že jeden datový tok reprezentuje jednosměrný tok paketů se společnými parametry v hlavičce paketu.



Obr. 1.15: Navázání spojení v cestě server-klient

Dle konfigurace aktivních a neaktivních časovačů síťového prvku podporujícího technologii monitorování datových toků, jsou udržované informace o jednotlivém datovém toku za určitý časový úsek exportovány na kolektor datových toků. S exportem datového toku daný tok v tabulce datových toků síťového prvku zaniká. Celý proces se pak periodicky opakuje. Kolektorem se myslí typicky dedikované zařízení uzpůsobené na ukládání a dlouhodobé udržování informací o všech datových tocích, které byly zdroji datových toků exportovány. Na kolektoru lze pak informace o síťovém provozu v podobě datových toků analyzovat. Popsaná situace je zachycena na obr. 1.16.



Obr. 1.16: Export datových toků po uplynutí aktivních nebo neaktivních časovačů

Mezi výhody monitorování datových toků patří nenáročnost na hardwarové prostředky úložiště dat – kolektoru. Je to dáno tím, že kolektor datových toků nemusí zpracovávat a ukládat celé pakety včetně jejich datového obsahu, nýbrž jen informace z hlavičky paketu. Lze tedy najednou analyzovat mnohonásobně více dat, které jsou skutečně podstatné pro detekci volumetrických DDoS útoků. Nevýhodou může být rychlost detekce. Analýza datových toků totiž zpravidla neprobíhá v reálném čase, což je způsobeno existencí výše zmíněných aktivních a neaktivních časovačů.

Jeden z možný scénářů monitorování a detekce volumetrických DDoS útoků s využitím technologie monitorování datových toků počítá s exportem datových toků z aktivního prvku na kolektor datových toků. Jako vhodným nástrojem se pro tyto účely jeví Flowmon kolektor s rozšiřujícím modulem DDoS Defender, o kterém pojednává kapitola 3.

Na základě zjištěných poznatků v této kapitole se pro monitorování a detekce volumetrických DDoS útoků v prostředí páteřních datových sítí s velkými přenosy v řádech desítek a stovek Gb za sekundu jeví metoda monitorování datových toků jako nejlepší. Pro část této práce pojednávající o praktickém řešení detekce a obrany před volumetrickými DDoS útoky byla tedy vybrána tato metoda.

2 Protokoly řízení datových toků

Tato kapitola si bere za cíl prozkoumat a srovnat známé protokoly pro řízení datových toků v prostředí páteřních datových sítí za účelem obrany před volumetrickými DDoS útoky.

2.1 BGP Flowspec

BGP Flowspec (Flow Specification) protokol je v současnosti v prostředí poskytovatelů připojení poměrně široce rozšířeným protokolem, jehož primárním účelem je obrana před distribuovanými útoky odepření služeb. Oficiálně byl definován v roce 2009 a popisuje jej dokument RFC5575.

V podstatě se jedná o upravený protokol BGP. Motivací k tomu byla jednoduchost protokolu BGP a jeho široká rozšířenost v prostředí páteřních datových sítí. Technologie BGP Flowspec tedy zaručuje snadné a rychlé nasazení do již existujících produkčních prostředí.

2.1.1 Princip BGP Flowspec protokolu

BGP Flowspec umožňuje v síti nakládat s provozem na základě definovaných pravidel. Pravidla se mohou skládat z několika dílčích položek, které popisuje tab. 2.1. Pro IPv4 protokol organizace IANA registruje několik typů položek Flow Spec (Flow Spec Component Types). Pro IPv6 dosud nedošlo k oficiálnímu schválení organizací IANA. Návrh pro IPv6 protokol popisuje „Dissemination of Flow Specification Rules for“ organizace IETF. [4]

Tab. 2.1: Typy BGP Flowspec položek pro IPv4 a IPv6

Typ	IPv4	IPv6
1	Cílový prefix	Cílový IPv6 prefix
2	Zdrojový prefix	Zdrojový IPv6 prefix
3	IP protokol	Další hlavička (poslední oktet položky Next Header v IPv6 paketu)
4	Port	Port
5	Cílový port	Cílový port
6	Zdrojový port	Zdrojový port
7	ICMP typ	ICMP typ
8	ICMP kód	ICMP kód
9	TCP příznaky	TCP příznaky
10	Délka paketu	Délka paketu
11	DSCP	DSCP
12	Fragment	Fragment
13	-	Flow Label (v záhlaví IPv6 paketu)

Další položkou BGP Flowspec pravidla je akce, která se má provést s příchozím paketem v případě, že se informace v hlavičce paketu shodují s daným pravidlem. Organizace IANA jednotlivé položky akce standardizuje mezi rozšířené komunitní položky (BGP extended community values), viz tab. 2.2.

Tab. 2.2: Vysvětlení funkci rozšířených komunitních položek protokolu BGP Flowspec

Typ	Rozšířená komunitní položka	Pravidlo sleduje/provádí akci
0x8006	traffic-rate	Maximální povolené množství provozu v bytech za sekundu
0x8007	traffic-action	Např. Vzorkování provozu na úrovni paketů
0x8008	Redirect	Přesměrování 2B a 4B autonomních systémů, přesměrování IPv4 a IPv6 adres
0x8009	traffic marking	Značkování provozu (DSCP hodnot)

Podle nastavených pravidel je pak zacházeno s každým paketem, který aktivní prvek musí zpracovat. Lze to tedy do jisté míry přirovnat k nastavení pravidel firewallu. Typicky využívanou akcí může být zahazování provozu v případě velkých datových přenosů (položka traffic-rate). Směrovače, které podporují BGP Flowspec technologii, se pak v síti provozovatele stávají svou funkčností podobné čističkám provozu, což významně snižuje náklady provozovatelů na využívání služeb obrany proti DDoS útokům.

Jednotlivé položky uvedené v tab. 2.1 a 2.2 jsou v síti propagovány v podobě BGP Update zpráv v poli NLRI variabilní délky.

2.2 OpenFlow protokol

OpenFlow je protokol počítačových sítí pracující na třetí vrstvě ISO/OSI síťového modelu. Je definován nad spolehlivým transportním protokolem TCP. Rané verze OpenFlow používaly port 6633, dnes je protokol OpenFlow standardně provozován nad portem 6653. Zabezpečenou komunikaci může zajišťovat sada protokolů TLS. OpenFlow standard je spravován a vyvíjen nezávislou skupinou Open Networking Foundation. [5]

Historie vzniku OpenFlow protokolu sahá do roku 2006. Tehdy měl projekt název „Ethane“ a stál za ním student doktorského studia Martin Casado na univerzitě ve Stanfordu. [6] V současnosti existuje protokol OpenFlow v nejaktuálnější verzi 1.5.1 definované v březnu roku 2015. [5] Výrobci, mezi něž patří např. společnost Extreme Networks nebo A10 Networks, však přecházejí na nové verze OpenFlow protokolu s jistou setrvačností. [7] [8]

Motivací k jeho vzniku a rozvoji je koncept programovatelných softwarově definovaných sítí SDN. Ten vychází z toho, že se snaží síťovým administrátorům podstatně zjednodušit správu zejména velmi rozsáhlých sítí o desítkách a stovkách aktivních síťových prvků. Takovéto množství síťových prvků většinou není stejného typu a není od stejného výrobce. To znamená, že velmi často disponují různými operačními systémy, různými skriptovacími jazyky, proprietárními síťovými rozhraními a v neposlední řadě i konfiguračními příkazy. Tímto nároky na kvalifikaci správce sítě a časové nároky na administraci mnoha zařízení exponenciálně stoupají. Koncept SDN tedy umožňuje centralizovanou administraci velkého množství síťových prvků pomocí jednoho standardu s otevřeným zdrojovým kódem.

Myšlenka SDN definuje tři funkčně odlišné vrstvy. Hierarchicky nejvýše je aplikační vrstva, která komunikuje s jí vrstvou podřízenou, tu zastává řídící vrstva s SDN kontrolérem. Aplikační vrstvu si lze představit jako samostatný software pro virtualizaci, monitoring aj. SDN kontrolér pak rozhoduje o podobě směrování provozu v síti. K tomu zasílá instrukce poslední vrstvě, kterou představují samotné síťové prvky, tedy směrovače či přepínače, hardwarové nebo virtualizované. V SDN tedy dochází k rozdělení funkcí přepínání paketů a dalšího směrování paketů (zacházení s pakety). To je odlišný přístup než u tradičních sítí, kde síťové prvky zastávají obě tyto funkce. Koncept SDN shrnuje obr. 2.1. [9]

Obr. 2.1: Diagram funkčních prvků softwarově definovaných sítí SDN

Článkem, který zajišťuje komunikaci mezi řídící vrstvou a vrstvou síťové infrastruktury, jsou specializované protokoly, mezi něž patří i protokol OpenFlow. [10]

2.2.1 Princip OpenFlow protokolu

OpenFlow přepínač je charakterizován třemi hlavními komponentami. První je tabulka toků a tabulka skupin, které zabezpečují vyhledávání a porovnávání paketů s pravidly v jejich záznamech. Další funkcí tabulek toků a skupin je přepínání paketů na základě těchto pravidel. Tabulka toků se skládá z jednotlivých záznamů toku, každý záznam toku obsahuje položky uvedené na obr. 2.2.

Obr. 2.2: Položky záznamu toku

Informace v Polích shody (např. verze 1.1.0 protokolu OpenFlow definuje 44) se porovnávají s informacemi v příchozích paketech. Jsou to vstupní port OpenFlow přepínače a data v hlavičkách paketů, mezi které patří např. zdrojová a cílová MAC a IP adresa, zdrojová a cílová síťová maska, pole ToS, TCP příznaky aj. Volitelně jsou porovnávány i další informace jako třeba metadata specifikovaná předchozí tabulkou toků.

Priorita v záznamu toku definuje přednost při zpracování paketů. Jednotlivý záznam toku jednoznačně popisují informace v položce Pole shody a Priorita.

Počitadla uchovávají informaci o počtu paketů, které se dosud shodly s pravidly záznamu toku. Pokud tedy dojde ke shodě, hodnota v položce Počitadlo je inkrementována.

Položka instrukce obsahuje sadu příkazů, které se provedou, pokud paket vyhovuje záznamu toku. Takovou instrukcí typicky může být zahození paketu nebo jeho přesměrování.

Časovače definují maximální možný čas neaktivity toku, tedy určitý časový úsek, za který bude záznam toku smazán z tabulky toků v případě, že nedojde k žádné shodě paketu s tímto záznamem toku.

Cookie položka obsahuje netransparentní datové hodnoty určené kontrolérem. Tyto hodnoty může kontrolér použít k filtrování záznamů toku způsobeným určitými statistickými hodnotami toku, změnami v daném toku a požadavky na smazání toku.

Příznaky mění způsob, jakým je se záznamy toků zacházeno. Příznak „OFPFF_SEND_FLOW_REM“ např. spustí zasílání zpráv o smazání toku pro daný záznam toku.

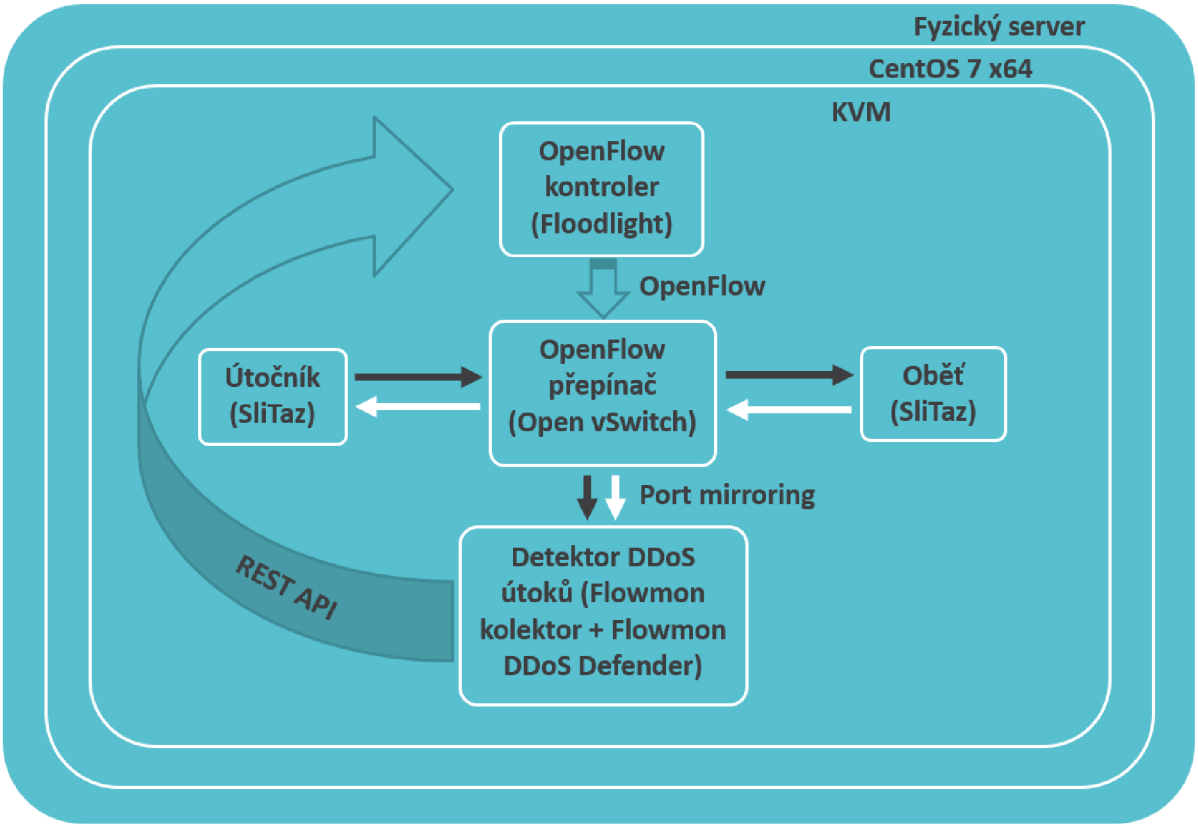
Jakmile paket dorazí na rozhraní OpenFlow přepínače, hledá se pro něj shoda mezi záznamy toků v tabulce nebo tabulkách toků. Jakmile dojde ke shodě, s paketem je dále naloženo tak, jak pro daný záznam toku definuje položka Instrukce. Pokud pro není nalezena žádná shoda se žádným záznamem toku v tabulce toků, další nakládání s paketem závisí na konfiguraci záznamu toku „table-miss“. Typicky to může být zahození paketu nebo přeposlání paketu směrem ke kontroléru přes OpenFlow kanál.

Druhou komponentou OpenFlow přepínače je tedy zabezpečený OpenFlow kanál umožňující komunikaci s externím kontrolérem. Hlavní úlohou kontroléru je spravovat obsah tabulky toků. Může vzdáleně přidávat, mazat a upravovat záznamy toku v tabulce toků. Tato komunikace je uskutečňována pomocí komponenty třetí, použitým standardem OpenFlow protokolu. [5]

Oproti protokolu BGP Flowspec je protokol OpenFlow málo rozšířený a složitý. Pro běžné síťové administrátory to může představovat překážku při jeho osvojení stejně tak jako změna konceptu správy sítě. Protokol OpenFlow je však daleko sofistikovanějším nástrojem na obranu před volumetrickými DDoS útoky. Poskytuje více možností pro tvorbu pravidel pro řízení datových toků a umožňuje centralizovanou správu více zařízení. Představuje více či méně vzdálenou budoucnost v oblasti počítačových sítí a moderních komunikačních technologií. V další části této práce bude tedy tato technologie dále rozváděna.

3 Návrh laboratoře pro detekci a obranu před volumetrickými DDoS útoky

Pro simulaci volumetrického DDoS útoku, jeho detekce a následnou obrannou reakci byla navržena laboratoř pro virtualizační prostředí QEMU operačního systému Linux obohaceného o modul jádra KVM kvm.ko. Tento modul umožňuje daleko výkonnější, akcelerovanou virtualizaci. Schéma zmíněné laboratoře je uvedeno na obr. 3.1.



Obr. 3.1: Blokové schéma laboratoře pro simulaci, detekci a obranu před volumetrickými DDoS útoky

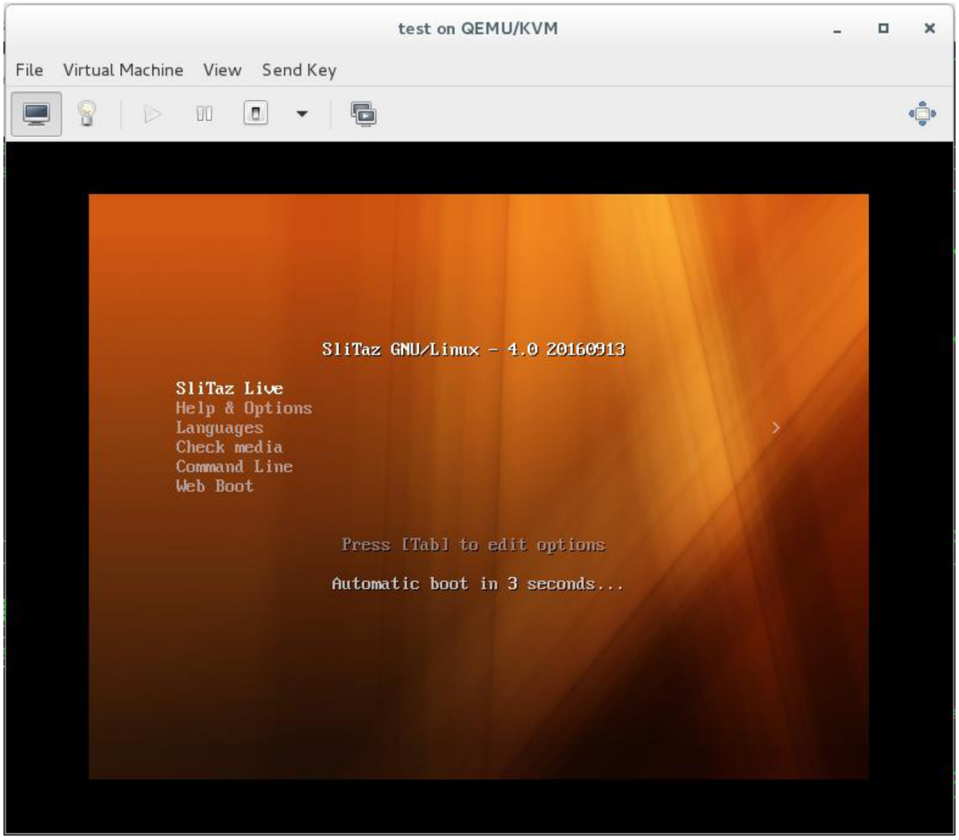
Ústředním prvkem laboratoře je SDN Open vSwitch přepínač, který realizuje síťové propojení útočníka a oběti. Mezi nimi probíhá simulovaná běžná komunikace pomocí nástroje hping3. Tato komunikace je neustále monitorována Flowmon kolektorem datových toků s rozšiřujícím modulem Flowmon DDoS Defender pro detekci volumetrických DDoS útoků. Aby byl tento monitoring možný, veškerý provoz proudící přes Open vSwitch je zrcadlen do monitorovacích portů Flowmon kolektoru, který z něj vytváří IPFIX standard a ukládá na svůj pevný disk. Rozšiřující modul Flowmon DDoS Defender pak automaticky analyzuje veškeré datové toky a v případě detekce DDoS útoku přes rozhraní REST API komunikuje s OpenFlow kontrolérem, konkrétně se jedná o implementaci Floodlight tohoto kontroléru. Tato komunikace je možná skrze integrační skript pro Flowmon DDoS Defender a Floodlight kontrolér. OpenFlow kontrolér dále díky zmíněnému skriptu získá informace o charakteristikách volumetrického DDoS útoku a pomocí OpenFlow protokolu nastaví v Open vSwitch přepínači takové pravidlo v ACL, aby účinně ubránil Oběť před volumetrickým DDoS útokem.

3.1 Open vSwitch přepínač

Open vSwitch je virtuální SDN přepínač s otevřeným zdrojovým kódem, který díky své programovatelnosti umožňuje automatizaci ve správě sítě. Tradiční funkce přepínače tedy obohacuje zejména o podporu protokolu OVSDB pro vzdálenou konfiguraci a správu síťových prvků. Rovněž podporuje komunikaci pomocí protokolu OpenFlow, který je podrobněji popsán v kapitole 2.2.1 včetně úlohy OpenFlow přepínače v kapitole 2.2. [11]

3.2 Útočník a Oběť

Útočníka a oběť simulovaly dva stejné stroje, speciálně optimalizované, s minimální hardwarovou náročností pro účely simulace DDoS útoků. Jednalo se o .iso bitový obraz upraveného systému GNU Linux distribuce SliTaz využívající jádro 2.6 na 32 bitové architektuře, viz obr. 3.3. Tyto virtualizované servery byly dostupné na portále technické podpory společnosti Flowmon Networks a.s jako live CD.



Obr. 3.3: Spouštěcí obrazovka SliTaz distribuce ve virtualizačním prostředí KVM

3.3 Floodlight OpenFlow kontrolér

Floodlight kontrolér je open-source projekt podporovaný společností Big Switch Networks založený na programovacím jazyce Java. Především podporuje komunikaci pomocí OpenFlow protokolu, díky němuž může fungovat jako centrální řídící bod pro správu všech prvků, které rovněž podporují tento protokol. Z pokročilých technologií Floodlight dále také nabízí podporu OpenStack platformy pro

správu cloudových distribuovaných sítí. Podrobnější informace o Floodlight kontroléru lze pak najít na oficiálních webových stránkách projektu. [12] REST API dokumentace Floodlight kontroléru, kterou jsme využili pro tvorbu integračních skriptů, je pak rovněž veřejně dostupná. [13]

3.4 Flowmon kolektor

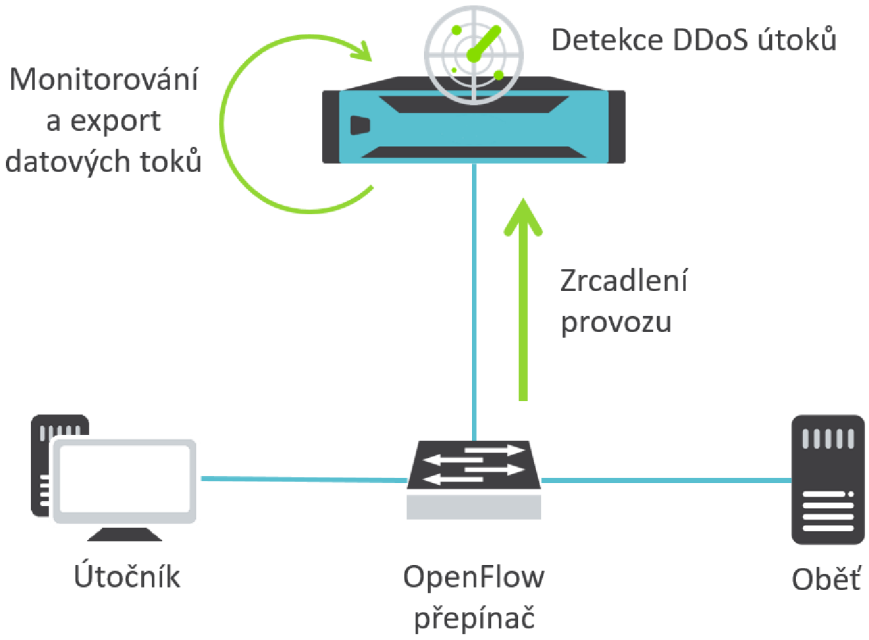
Flowmon kolektor je kolektor datových toků umožňující dlouhodobé uchovávání datových toků z mnoha zdrojů datových toků, detailní analýzu a zobrazení datových toků. Flowmon kolektor je možné nasadit jako hardwarový server nebo také jako server virtuální. Na webových stránkách výrobce je uvedena oficiální podpora pro VMware a Hyper-V virtualizační prostředí.

Hlavním měřítkem výkonnosti kolektoru je možný počet zpracovaných toků za vteřinu. Ta se u Flowmon kolektorů pohybuje od 75 000 až do 250 000 toků za sekundu u nejvýkonnějšího modelu. U virtuálních kolektorů je toto číslo 200 000 toků za sekundu. Výrobce však podotýká, že „maximálního výkonu lze dosáhnout při vyčlenění hardwarových prostředků odpovídajících specifikaci hardwarového kolektoru včetně výkonu diskového úložiště“. [14]

Rozdílem mezi hardwarovým a virtuálním Flowmon kolektorem je dále ten, že virtuální varianta navíc disponuje dvěma monitorovacími porty s exportéry, které umožňují export datových toků ve standardech NetFlow v5, NetFlow v9 a IPFIX. Flowmon kolektor lze tedy použít zároveň i jako zdroj datových toků, čehož s výhodou využijeme při sestavování laboratoře ve virtualizačním prostředí KVM.

Funkce Flowmon kolektoru mohou být rozšířeny o další softwarové moduly umožňující behaviorální analýzu sítě, monitorování výkonnosti aplikací, monitorování celých paketů a obranu před volumetrickými DDoS útoky. [15]

Pro monitoring a ukládání datových toků síťového provozu mezi obětí a útočníkem tedy využijeme právě Flowmon kolektoru díky jeho univerzálnímu využití. S tímto kolektorem totiž můžeme jednak vytvářet datové toky, jednak je ukládat a s rozšiřujícím modulem Flowmon DDoS Defender rovněž automaticky detekovat volumetrické DDoS útoky a dynamicky na ně reagovat. Flowmon kolektor s rozšiřujícím modulem DDoS Defender je umístěn v horní části souhrnného obr. 3.4.



Obr. 3.4: Funkce Flowmon kolektoru s rozšiřujícím modulem DDoS Defender v použité laboratoři

3.5 Flowmon DDoS Defender

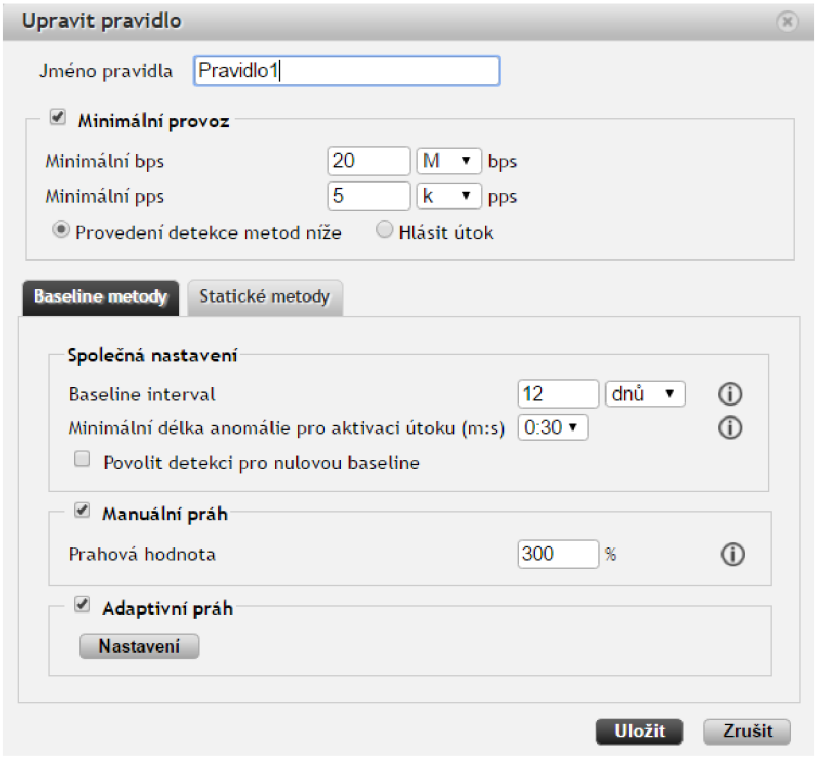
Tento rozšiřující softwarový modul pro Flowmon kolektor používá datové toky na něm uložené jako vstupní data pro detekci volumetrických DDoS útoků. Pro každý chráněný segment, kterým se rozumí IP adresa nebo rozsah IP adres, DDoS Defender provádí mnohafaktorovou detekci útoků.

Modul sleduje minimální množství příchozích paketů a bytů za sekundu do chráněných segmentů výše zmíněným definovaným způsobem. Dále sleduje dynamicky vytvářený práh příchozího provozu, kde uživatel může definovat, jak dlouho se klasifikátor bude učit charakter provozu pro výpočet tohoto prahu.

Mezi další sledovaný faktor pak patří „Manuální práh“, jehož překročení a následná detekce útoku závisí na nastavení procentuálního množství příchozích paketů z tohoto dynamicky učeného prahu.

Pro každý chráněný segment jsou rovněž sledovány adaptivní prahy detekující útok a prahy detekující podezření na tento útok. Tyto prahy jsou učeny pro ICMP, UDP a TCP provoz s různými kombinacemi TCP příznaků.

Posledním prahem je pak sledování uživatelsky definovaného poměru příchozích a odchozích paketů. Lze tedy definovat, kolikrát musí být množství příchozích paketů větší než množství odchozích z daného chráněného segmentu. Ukázka nastavení jednotlivých prahů detekce v modulu DDoS Defender je uvedena na obr. 3.5.



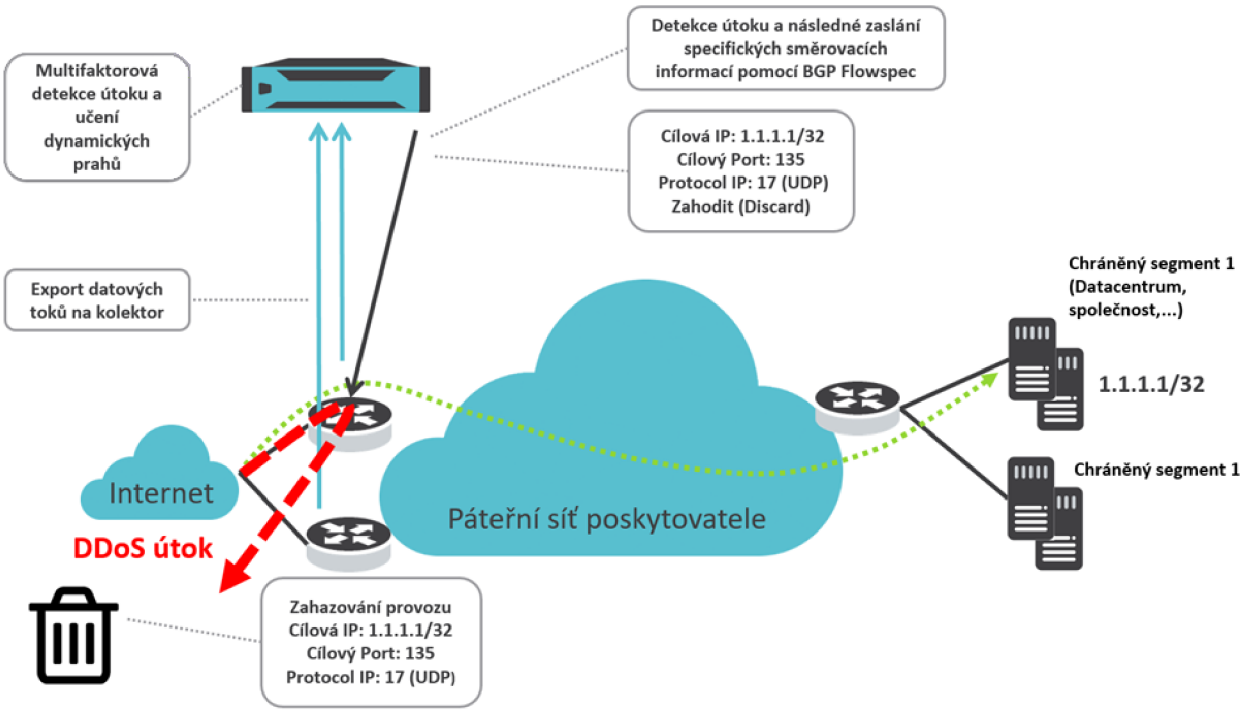
Obr. 3.5: Ukázka nastavení detekčních prahů modulu Flowmon DDoS Defender

Pokud je některý z prahů překročen, modul DDoS Defender detekuje útok a umožňuje automaticky provést několik druhů akcí za účelem ochrany chráněného segmentu či segmentů. První představuje jednoduché upozornění na probíhající detekovaný útok ve formě emailové zprávy, syslog zprávy, SNMP Trap zprávy nebo také umožňuje automatické spuštění uživatelského skriptu. Této možnosti dále využijeme pro nahrání integračního skriptu, jehož účelem bude předání charakteristik detekovaného volumetrického DDoS útoku OpenFlow kontroleru přes REST API komunikační kanál.

V případě útoku DDoS Defender získá informace o útoku, např. na jakém transportním protokolu probíhá, jakou má útok zdrojovou IP, a má možnost se s využitím SSH protokolu vzdáleně přihlásit na předem definovaný směrovač. Na tomto směrovači rovněž musí být již předem definovaný ACL, do kterého pak DDoS Defender zapíše pravidlo, aby daný chráněný segment uchránil před DDoS útokem. Díky tomu lze uskutečňovat koncept uživatelem definovaného směrování PBR nebo směrování provozu do místa jeho zahození, kdy se jedná o tzv. blackholing.

Další, automatizovaná, akce je možnost přesměrování DDoS provozu pomocí externí nebo interní BGP injekce. Toho lze typicky využít pro výše zmíněný blackholing nebo pro přesměrování veškerého provozu autonomního systému, odkud je iniciován DDoS útok, do čističky síťového provozu. Zde je pak provoz zbaven DDoS útoku a legitimní provoz směrován zpět do chráněnému segmentu.

Asi nejsofistikovanější možností je poslední možnost obrany, kterou modul DDoS Defender nabízí. Tou je využití BGP Flowspec protokolu, který byl blíže popsán v kapitole 2.1. V tomto scénáři pak odpadá část čističky provozu, neboť DDoS Defender je schopen předat informace o charakteristikách útoku směrovači s podporou technologie BGP Flowspec. Obr. 3.6 tuto situaci ilustruje. [16]



Obr. 3.6: Jedno z možných nasazení modulu Flowmon DDoS Defender s využitím technologie BGP Flowspec

4 Sestavení laboratoře pro detekci a obranu před volumetrickými DDoS útoky

Na dostatečně výkonném dedikovaném serveru s 8 procesorovými jádry (16 vlákny) a 16 GB RAM byl jako první krok nainstalován 64 bitový operační systém CentOS 7 na platformě Linux/UNIX. Dále pak byla potřeba nainstalovat balíčky, které umožní provozovat KVM virtualizaci:

yum install qemu-kvm qemu-img virt-manager libvirt libvirt-python libvirt-client virt-install virt-viewer # yum install "@X Window System" xorg-x11-xauth xorg-x11-fonts-* xorg-x11-utils -y bridge-utils.

Dalším důležitým krokem pro umožnění virtualizace bylo spuštění a povolení libvirtd démona:

systemctl start libvirtd
systemctl enable libvirtd.

Vzhledem k tomu, že se jednalo pouze o minimální instalaci pouze s nezbytnými komponenty tohoto operačního systému, byly po základní konfiguraci systému manuálně doinstalovány součásti umožňující uživatelsky přívětivější práci se systémem jako je grafické uživatelské rozhraní nebo program XRDP pro přístup ke vzdálené ploše.

4.1 Nasazení Open vSwitch přepínače

Následujícím krokem byla přes nástroj yum instalace a aktualizace následujících uvedených balíčků. Mj. byl aktualizován vývojářský nástroj „autoconf“, abychom zajistili budoucí správné kompilace a instalace softwaru na této platformě:

#yum -y install make gcc openssl-devel autoconf automake rpm-build redhat-rpm-config python-devel openssl-devel kernel-devel kernel-debug-devel libtool wget.

Dalšími kroky bylo stažení Open vSwitch přepínače ve verzi 2.5.1 s podporou OpenFlow protokolu, vytvoření RPM instalačního balíčku a samotná instalace nově vytvořeného balíčku:

mkdir -p ~/rpmbuild/SOURCES

wget http://openvswitch.org/releases/openvswitch-2.5.1.tar.gz

cp openvswitch-2.5.1.tar.gz ~/rpmbuild/SOURCES/

tar xfz openvswitch-2.5.1.tar.gz

sed 's/openvswitch-kmod, //g' openvswitch-2.5.1/rhel/openvswitch.spec > openvswitch-2.5.1/rhel/openvswitch_no_kmod.spec

rpmbuild -bb --nocheck ~/openvswitch-2.5.1/rhel/openvswitch_no_kmod.spec

ls -l ~/rpmbuild/RPMS/x86_64/

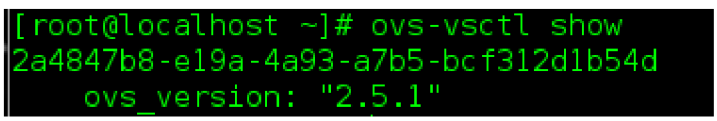
yum localinstall ~/rpmbuild/RPMS/x86_64/openvswitch-2.5.1-1.x86_64.rpm.

Na závěr byla spuštěna Open vSwitch služba. Zároveň také došlo k povolení spouštění této služby při každém startu systému. K tomu jsme použili následujících příkazů:

systemctl start openvswitch.service

chkconfig openvswitch on.

Pomocí příkazu „ovs-vsctl show”, který je uveden na obr. 4.1, jsme ověřili, že Open vSwitch byl úspěšně nainstalován a že služba openvswitch je v provozu.



Obr. 4.1: Ověření správné instalace Open vSwitch přepínače do KVM virtualizačního prostředí

4.2 Nasazení Floodlight OpenFlow kontroléru

Pro instalaci Floodlight OpenFlow kontroléru byly nejdříve nainstalovány a aktualizovány následující knihovny:

sudo apt-get install build-essential default-jdk ant python-dev eclipse.

Dále pak byl stažen Floodlight kontrolér, vytvořen java archiv a spuštěn samotný kontrolér:

git clone git://github.com/floodlight/floodlight.git

ant

java -jar target/floodlight.jar.

Na závěr jsme pak ve firewallu operačního systému fyzického serveru povolili TCP port 8080 a 6653. Povolením portu TCP/8080 bylo zpřístupněno grafické uživatelské rozhraní kontroléru a povolením portu TCP/6653 pak komunikace pro OpenFlow protokol mezi Open vSwitch přepínačem a Floodlight kontrolérem.

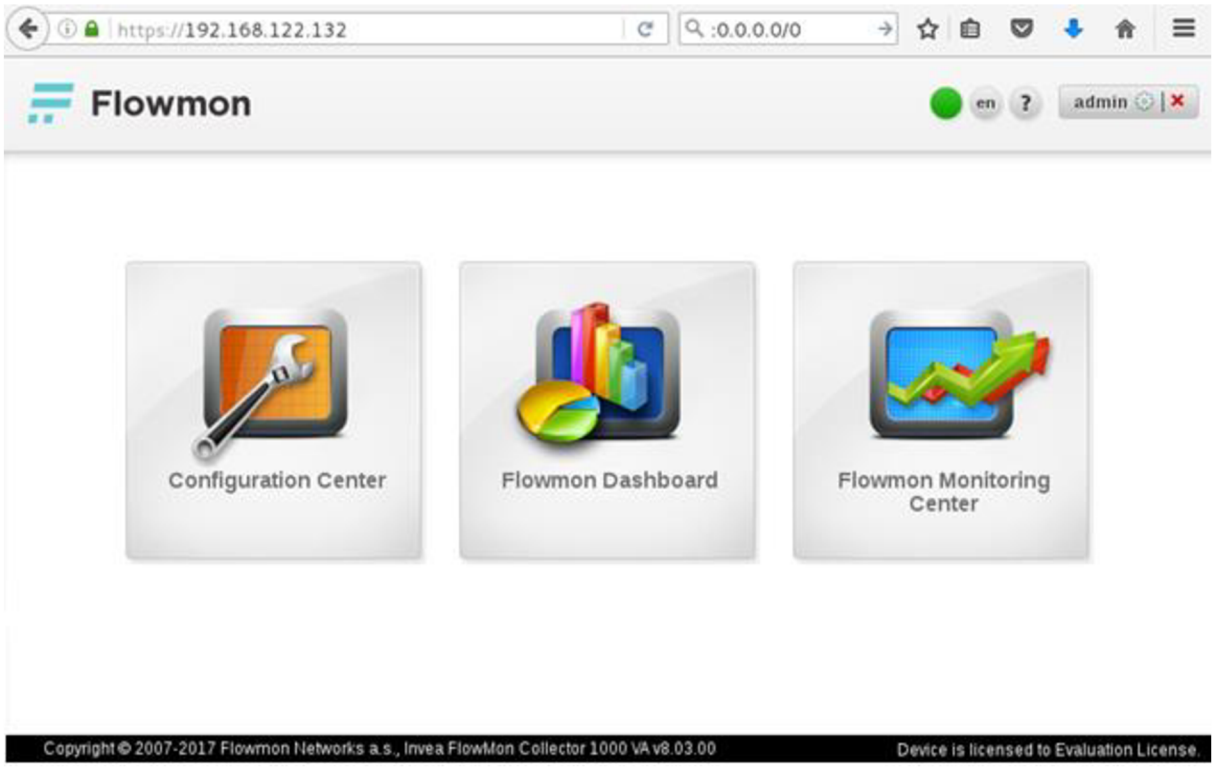
4.3 Nasazení Útočníka a Oběti

Nasazení útočníka a oběti do DDoS laboratoře představovalo jen několik málo jednoduchých kroků: přidání live CD těchto serverů do QEMU/KVM virtualizačního prostředí a po jejich úspěšném startu pak statická konfigurace IP adres na jejich administrativních síťových rozhraních.

4.4 Nasazení Flowmon kolektoru

Nasazení Flowmon kolektoru do prostředí QEMU/KVM proběhlo pomocí instalačního .iso obrazu. Aby se však systému podařilo korektně spustit, bylo zapotřebí ve správci virtuálních strojů Virtual Machine Manager změnit diskovou sběrnici na technologii SATA. V opačném případě pokus systému o start vždy skončil havárií – chybou kernel panic.

Po úspěšné instalaci a spuštění Flowmon kolektoru došlo pomocí virtuální konzole k nastavení dynamicky přidělované IP adresy administrativnímu rozhraní. Tím bylo zpřístupněno grafické uživatelské rozhraní kolektoru zobrazené na obr. 4.2, kde bylo provedeno nahrání testovací licence. Bez platné licence by totiž kolektor nebyl schopen vytvářet ani ukládat datové toky.

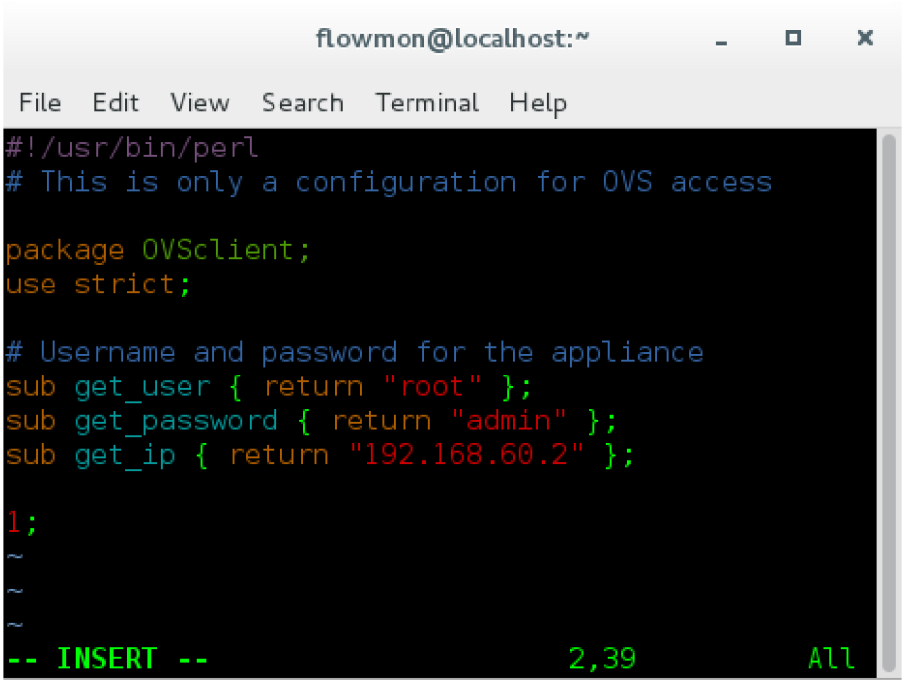


Obr. 4.2: Úvodní stránka grafického uživatelského rozhraní Flowmon kolektoru

4.5 Nasazení modulu DDoS Defender

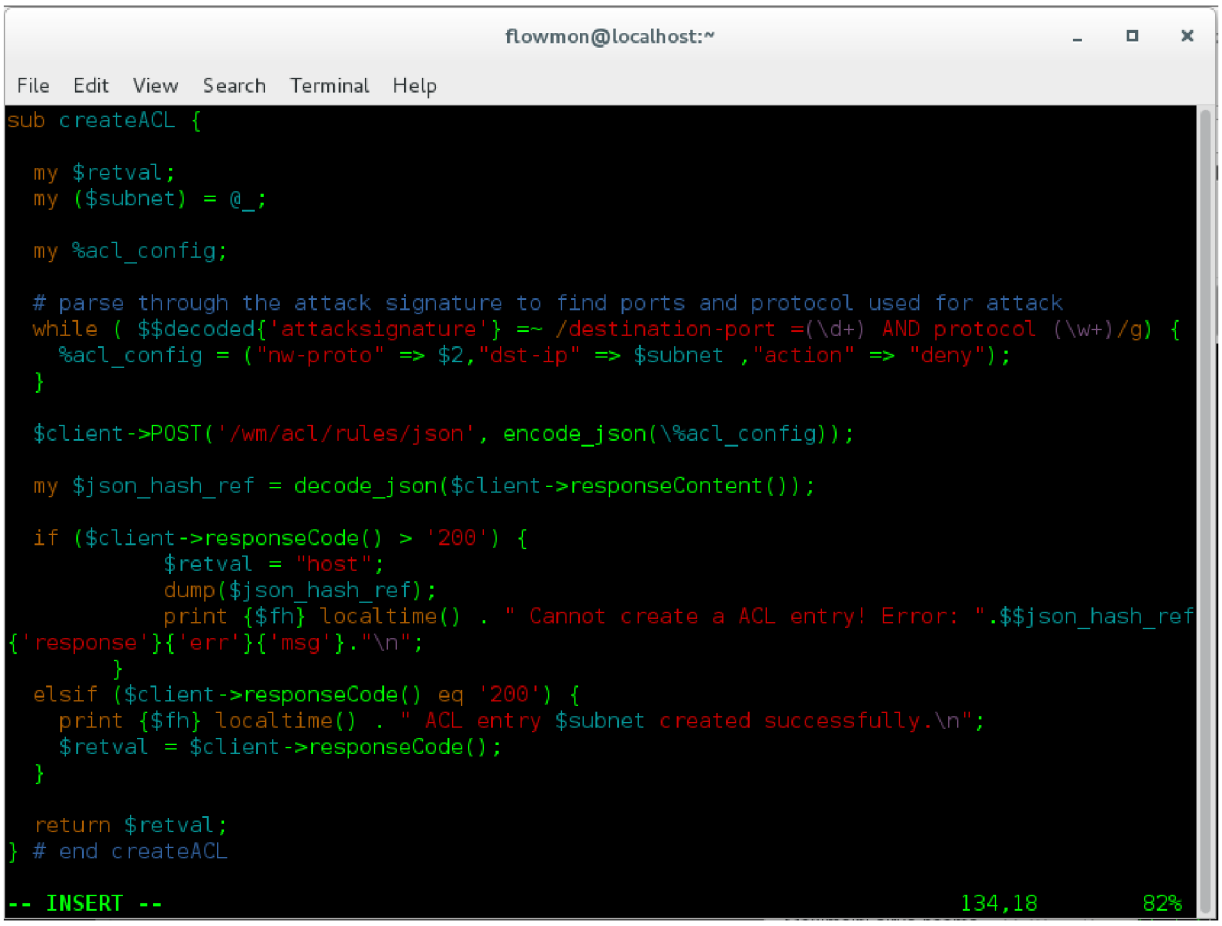
Nasazení rozšiřujícího modulu Flowmon DDoS Defender proběhlo jednoduchou instalací balíčku v grafickém uživatelském rozhraní modulu Flowmon Configuration Center. Nejprve však byla potřeba jej stáhnout. Zkušební licence umožňovala automatické stažení rozšiřujících modulů, na něž se licence vztahovala. Této možnosti jsme tedy pro instalaci modulu DDoS Defender využili. Žádné další instalační kroky nebyly vyžadovány.

Pro základní konfiguraci modulu DDoS Defender byly nejdříve na Flowmon kolektoru vytvořeny dva integrační skripty v jazyce Perl. První z nich zobrazený na obr. 4.3 má za úkol přihlášení se k operačnímu systému fyzického serveru laboratoře na IP adrese lokální sítě 192.168.60.2 pod uživatelským jménem „root“ a heslem „admin“.



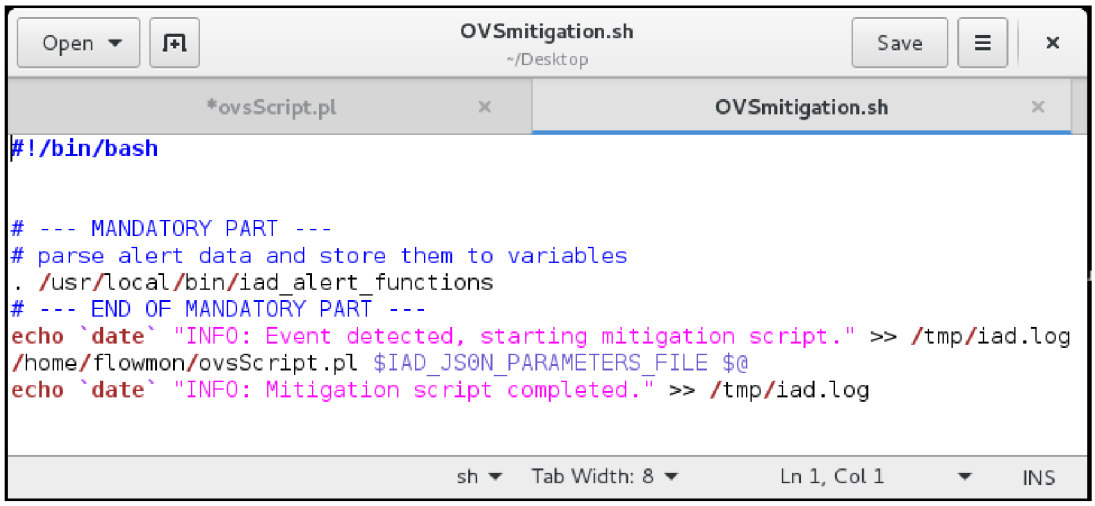
Obr. 4.3: Skript pro přihlášení k operačnímu systému fyzického serveru

Část druhého skriptu je uvedena na obr. 4.4. Tato část popisuje funkci skriptu, která vytváří ACL a jeho pravidlo na základě získané charakteristiky DDoS útoku uložené na kolektoru v podobě .json souboru. Toto pravidlo je na Open vSwitch přepínač doručeno pomocí REST API příkazu. Mezi funkce skriptu patří dále přihlášení se k Floodlight kontroléru a již zmíněné získání informací o charakteristikách DDoS útoku z modulu DDoS Defender.



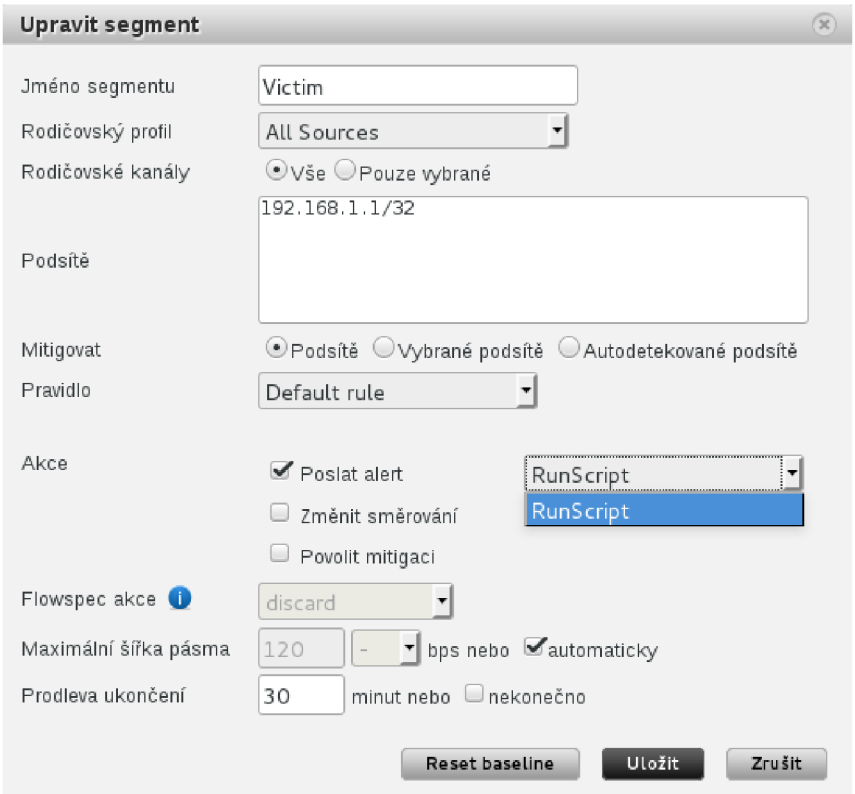
Obr. 4.4: Část obranného skriptu s funkcí pro vytvoření ACL pravidla pro Open vSwitch přepínač

Následujícím konfiguračním krokem bylo vytvoření Alertu v modulu DDoS Defender, který bude proveden v případě detekce DDoS útoku. Jedna z možností při definici Alertu je totiž spuštění uživatelského skriptu. Toho jsme s výhodou využili a vybrali poslední, třetí, skript, který má za úkol charakteristiku DDoS útoku uložit do proměnných a spustit skript pro vytvoření ACL pravidla. Tento skript je jako jediný napsán ve skriptovacím jazyce Bash, jeho celé znění je uvedeno na obr. 4.5.



Obr. 4.5: Bash skript pro spuštění obranného procesu

Posledním krokem pak byla definice chráněného segmentu v podobě podsítě 192.168.1.1/32 a propojením tohoto chráněného segmentu s výše vytvořeným Alertem. IP adresa 192.168.1.1/32 byla volena proto, že v kroku 4.3 byla přidělena administrativnímu síťovému rozhraní Oběti. Nastavení chráněného segmentu je zachycena na obr. 4.6.



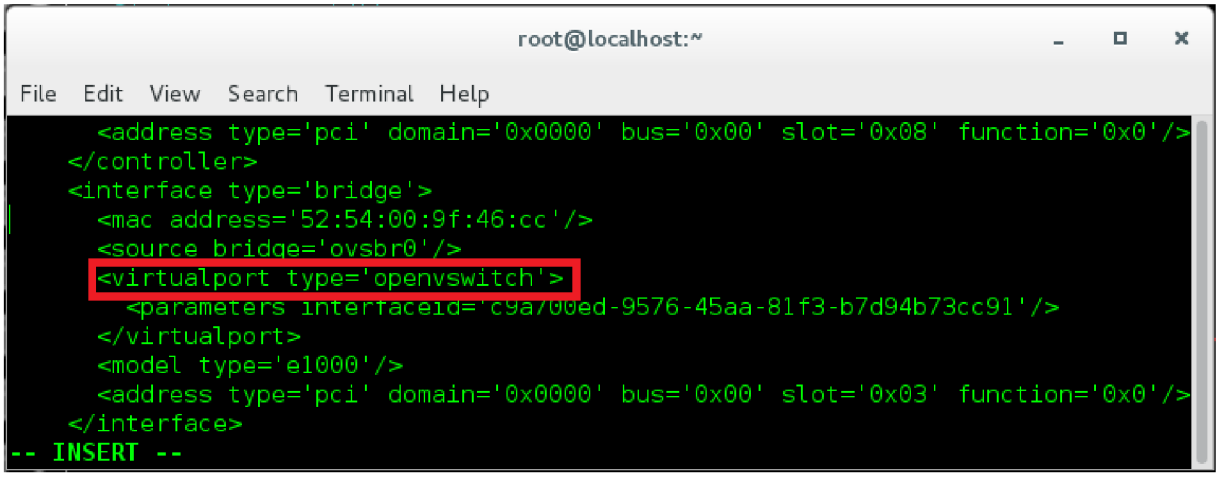
Obr. 4.6: Konfigurace chráněného segmentu v modulu Flowmon DDoS Defender

4.6 Síťová konfigurace laboratoře

Po úspěšné instalaci Open vSwitch přepínače byla všechna zařízení, útočník, oběť a analyzátor provozu Flowmon kolektor, síťově propojena. Za tímto účelem byl nejdříve vytvořen virtuální síťový most pro Open vSwitch přepínač pojmenovaný ovsbr0 pomocí příkazu „ovs-vsctl add-br ovsbr0“. Dále byla virtuálnímu síťovému mostu nakonfigurován IP adresa 192.168.1.208, abychom pomocí ní propojili Floodlight OpenFlow kontroler s Open vSwitch přepínačem. Toho bylo dosaženo použitím příkazu „ifconfig ovsbr0 192.168.1.208 netmask 255.255.255.0“. Na závěr byl tento virtuální síťový most pomocí příkazu „ovs-vsctl set bridge ovsbr0 protocols=OpenFlow13“ nakonfigurován tak, aby používal standard verze 1.3 OpenFlow protokolu.

V grafickém uživatelském rozhraní aplikaci Virtual Machine Manager sloužící pro správu virtualizovaných strojů bylo pak každému zařízení v simulované laboratoři upraveno nastavení síťové karty, aby využívala nově vytvořený síťový most „ovsbr0“. Rovněž byl změněn model síťové karty na „e1000“. V opačném případě totiž daný stroj zhavaroval s chybou kernel panic.

Jako další krok následovala úprava konfiguračního .xml souboru každého ze zařízení. Aplikace Virtual Machine Manager totiž není zcela uzpůsobena možnosti přidávání síťových karet virtualizovaných strojů do Open vSwitch přepínače. V programu virsh byla tedy potřeba na každém stroji manuálně nastavit typ síťové karty jako „openvswitch“. Tím se automaticky při startu zařízení síťová karta připojí do Open vSwitch přepínače. Ukázku jednoho z konfiguračního souborů zobrazuje obr. 4.7.



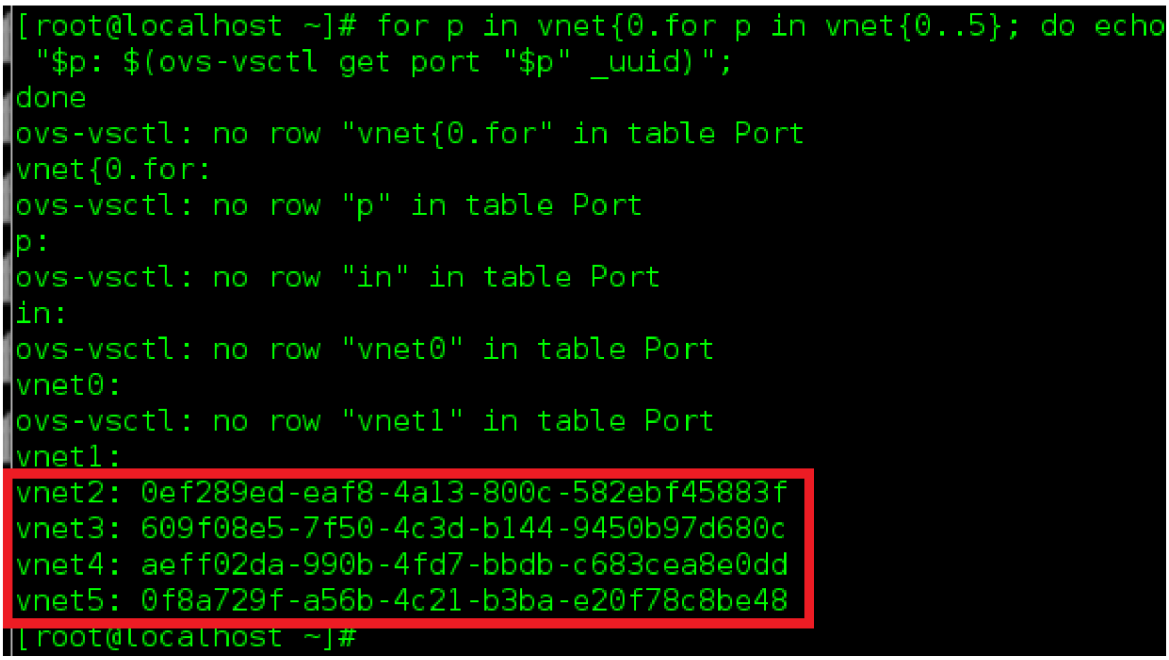
Obr. 4.7: Ukázka manuální konfigurace jednoho z virtualizovaných strojů

4.7 Konfigurace monitorování provozu v laboratoři

Aby bylo možné monitorovat provoz v laboratoři směřující z útočníka na oběť, nejprve byl vytvořen mirror port a přidán do virtuálního síťového mostu vytvořeného v předchozí kapitole:

#ovs-vsctl -- --id=@m create mirror name=MirrorPort0 -- add bridge ovsbr0 mirrors @m.

Po vytvoření mirror portu byla potřeba specifikovat, který provoz do něj bude kopírován. Nejdříve bylo tedy nutno zjistit, jaký je identifikátor UUID portu, jehož oboustranný provoz bude přiváděn do mirror portu. Zajímalo nás konkrétně UUID rozhraní vnet5, protože rozhraní vnet5 představuje síťové rozhraní Oběti. Na obr. 4.8 je uveden příslušný příkaz současně s výpisem UUID identifikátorů.



Obr. 4.8: Zobrazení UUID pro každé z rozhraní typu vnet

Jako další krok bylo nakonfigurováno zrcadlení provozu z rozhraní vnet5 v obou směrech. Toto zrcadlení bylo aplikováno do mirror portu vytvořeného výše:

#ovs-vsctl set mirror MirrorPort0 select_src_port=0f8a729f-a56b-4c21-b3ba-e20f78c8be48 select_dst_port=0f8a729f-a56b-4c21-b3ba-e20f78c8be48.

Posledním krokem při nastavení zrcadlení provozu byla specifikace, na které virtuální síťové rozhraní se nastavení mirror portu s názvem MirrorPort0 bude aplikovat. Bylo zvoleno vnet2 rozhraní, protože to představuje monitorovací port Flowmon kolektoru:

ovs-vsctl -- --id=@vnet2 get port vnet4 -- set mirror ddosmirror output-port=@vnet2.

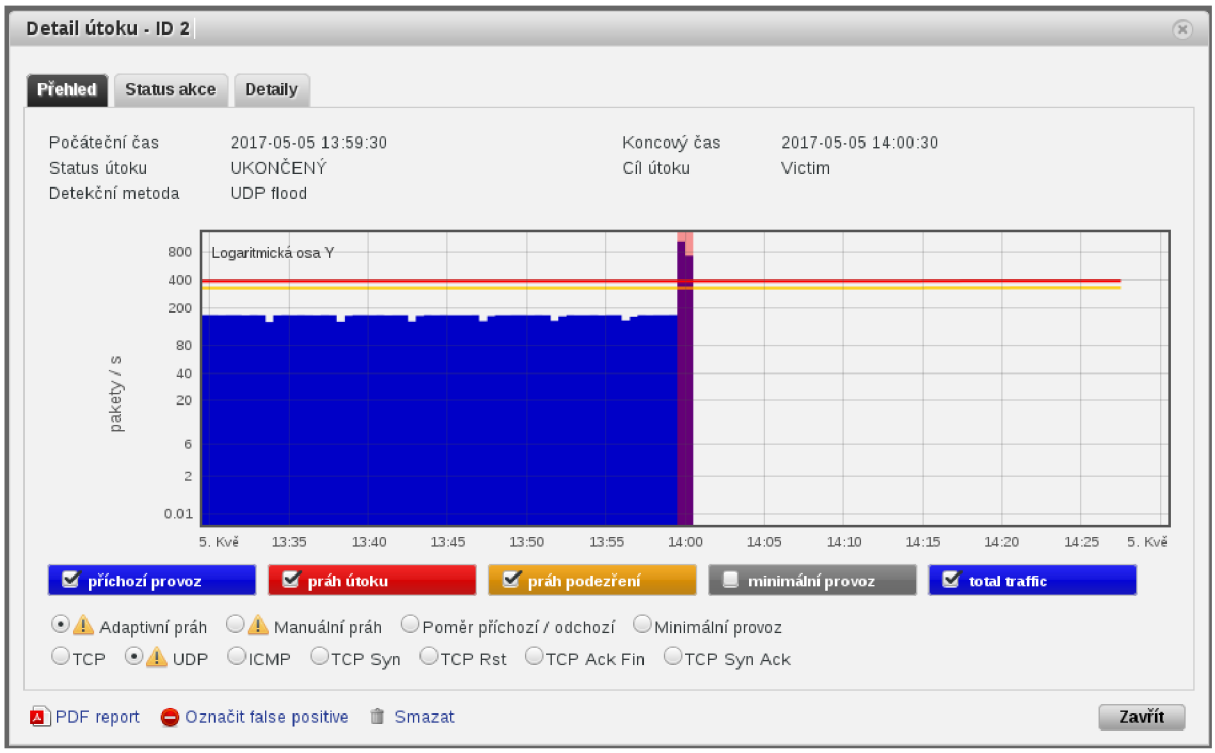
Tím jsme dosáhli toho, že celá oboustranná komunikace mezi Útočníkem a Obětí byla monitorována Flowmon kolektorem a vyhodnocována rozšiřujícím modulem Flowmon DDoS Defender.

5 Simulace volumetrického DDoS útoku typu UDP Flood a obrana před ním

Cílem této kapitoly je demonstrovat automatizovanou detekci a obranu před volumetrickým DDoS útokem typu UDP Flood v laboratoři, která byla blíže popsána a sestavena v kapitole 4.

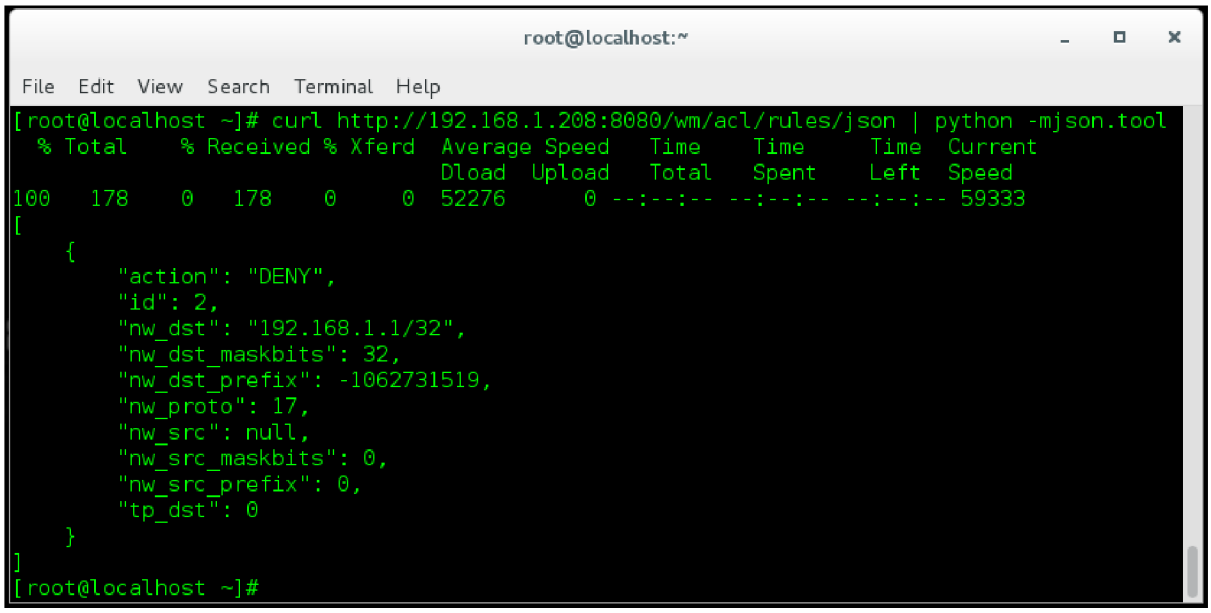
Ze všeho nejdříve byla potřeba přibližně simulovat běžný provoz mezi Útočníkem a Obětí. K tomu byl použit nástroj hping3, který byl nakonfigurován tak, aby generoval UDP a ICMP provoz v řádu Mb za sekundu. Tento provoz běžel několik hodin na pozadí, aby se modul Flowmon DDoS Defender naučil jeho charakter a vytvořil si prahy útoku pro každý ze zmiňovaných typů provozu.

Po tomto období byl pomocí stejného nástroje spuštěn volumetrický DDoS útok typu UDP Flood, perioda generování UDP paketů za sekundu byla snížena na 500µs. Během asi 40 vteřin modul DDoS Defender tento útok správně detekoval, viz obr. 5.1, a došlo ke spuštění obranných skriptů blíže popsaných v kapitole 4.5.



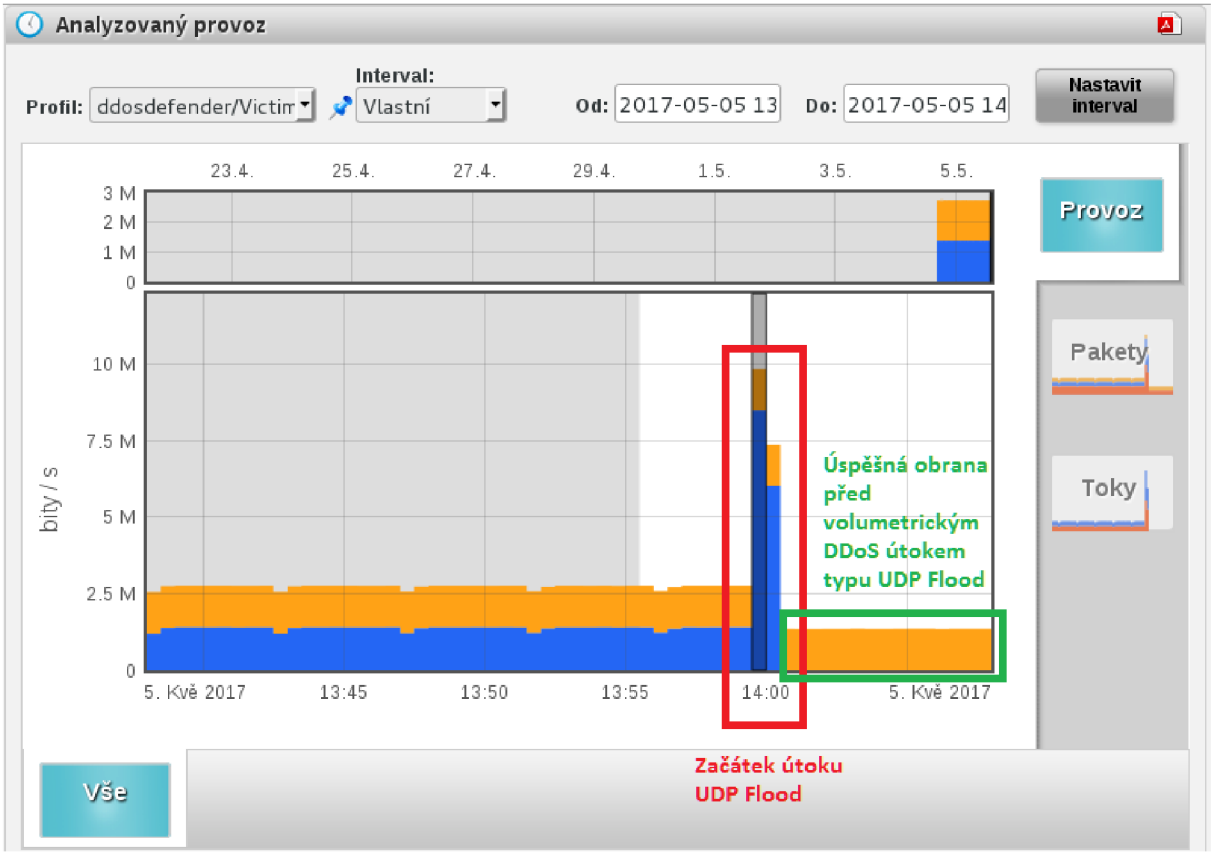
Obr. 5.1: Záznam detekovaného simulovaného DDoS útoku modulem Flowmon DDoS Defender

Jako důsledek těchto skriptů Flowmon DDoS Defender přes REST API rozhraní instruoval Floodlight kontrolér, aby nastavil nový ACL, který potlačí UDP Flood útok. Floodlight kontrolér pak pomocí OpenFlow protokolu toto ACL pravidlo zapsal na k němu připojený Open vSwitch přepínač, což lze vidět na obr. 5.2.



Obr. 5.2: ACL pravidlo na Open vSwitch přepínači vytvořené vlivem detekce UDP Flood útoku

Tím došlo k obraně chráněného segmentu před UDP Flood útokem. Ostatní provoz v laboratorních podmínkách reprezentovaný provozem na ICMP protokolu zůstal nedotčen. To dokazuje obr. 5.3, kde je UDP provoz reprezentován modrou barvou, ICMP pak barvou oranžovou.



Obr. 5.3: Analýza průběhu UDP Flood útoku

6 Závěr

V rámci diplomové práce bylo podrobně pojednáno o technologii NetFlow/IPFIX a dalších odvozených standardů pro monitorování počítačových sítí, se kterými se lze v praxi nejčastěji setkat. Rovněž je v práci popsáno Flowmon řešení primárně určené pro monitorování datových toků v počítačových sítích. V teoretické části je rozebrán jeden z nejmodernějších přístupů ke správě počítačových sítí. Přesněji technologie softwarově definovaných sítí SDN a s ní související protokol OpenFlow ve verzi 1.3 a vyšší. Dále jsou popsány některé možné metody monitorování a detekce volumetrických DDoS útoků v prostředí páteřních sítí.

Nejdříve navržena možnost monitorování a detekce pomocí SNMP protokolu. Tato metoda se pro účely detekce jevila jako velmi vhodná pro svou jednoduchost a výpočetní nenáročnost. Jako značná nevýhoda však byla skutečnost, že pro tyto účely SNMP protokol podával pouze velmi strohé informace o přenesených datech. Nešlo by tak tedy např. blíže identifikovat, odkud útok pochází a kam míří. Vytvoření automatizované efektivní obrany proti volumetrickým DDoS útokům by pak prakticky nebylo možné.

Druhá možnost zvažovala záchyt celého provozu, celých paketů. Tento přístup řešil problém s nedostatkem bližších informací o monitorovaném útoku, tak jak tomu bylo u monitorování a detekce v případě použití protokolu SNMP. V případě paketové analýzy lze totiž analyzovat veškeré informace o útoku včetně jeho datové části, která se však ukázala jako irelevantní pro účely monitoringu a detekce volumetrických DDoS útoků. Dalším nedostatkem tohoto přístupu byla vysoká výpočetní náročnost, a to především ve zkoumaném prostředí páteřních datových sítí.

Třetí možnost se tedy snažila najít a použít výhody obou dříve jmenovaných přístupů. Pro účely monitorování a detekce volumetrických DDoS útoků v prostředí páteřních datových sítích se tedy jako jednoznačně nejlepší ukázala technologie monitorování datových toků, protože představuje vyváženou kombinaci mezi výpočetní náročností a množstvím poskytovaných informací o monitorovaném provozu. Je tedy prakticky realizovatelná a použitelná pro podniknutí následujících efektivních obranných kroků při více či méně automatizované obraně před volumetrickými DDoS útoky.

Součástí výše zmíněných návrhů byla otázka praktického přivedení monitorovaného provozu na generátor a analyzátor datových toků. V úvahu připadala možnost použití zrcadlení provozu pomocí mirror portu nebo použití síťových tapů. Druhá jmenovaná byla pro prostředí páteřních datových sítí vyhodnocena jako jednoznačně lepší, neboť síťové tapy jsou schopny monitorovat šířku pásma celé linky a žádným způsobem nezatěžují a nezasahují do jiných prvků v síti. Proto bylo o síťových tapech také blíže pojednáno.

Praktická část diplomové práce pak čerpá z její části teoretické. Ze zjištěných poznatků byla zvolena a navržena konkrétní technika detekce a obrany před volumetrickými DDoS útoky pomocí konceptu monitorování datových toků a softwarově definovaných sítí. Byla tedy sestavena laboratoř ve virtualizačním prostředí KVM pro efektivní simulaci DDoS útoků. Obsahovala dvojí nasazení distribuce SliTaz. Jedna z nich zastávala roli oběti a druhá roli útočníka, který pomocí nástroje hping3 simuloval volumetrický útok typu UDP Flood. Dále byl do laboratoře nasazen Open vSwitch SDN přepínač verze 2.5.1 a nakonfigurován ve standardu protokolu OpenFlow verze 1.3. Tento přepínač realizoval síťové propojení mezi útočníkem a obětí a rovněž na něm bylo nakonfigurováno zrcadlení provozu do připojeného monitorovacího portu Flowmon kolektoru, který na bázi datových toků monitoroval provoz v síti.

Nad Flowmon kolektorem byl nainstalován jeho rozšiřující modul DDoS Defender pro detekci a obranu před volumetrickými DDoS útoky. Aby bylo dosaženo automatizované obrany před simulovaným DDoS útokem, součástí práce bylo vytvoření obranných skriptů. Při detekci DDoS útoku modulem DDoS Defender skripty získaly charakteristiky tohoto útoku a předaly je přes rozhraní REST API Floodlight SDN kontroléru. Ten pomocí OpenFlow protokolu vytvořil a zapsal do Open vSwitch SDN přepínače příslušné ACL pravidlo, které v konečném důsledku ochránilo oběť před simulovaným útokem typu UDP Flood. Legitimní provoz směřující k oběti zůstal touto akcí nezměněn.

Literatura

[1] CASE, J., M. Fedor, M. Schoffstal a J. Davin. A Simple Network Management Protocol (SNMP). *The Internet Engineering Task Force.*[online]. Květen 1990 [cit. 2016-10-01]. Dostupné z: https://tools.ietf.org/html/rfc1157

[2] FERENZ, Christian. GBIT COPPER TAPPING. *Cubro* [online]. 27.9. 2015 [cit. 2017-05-07]. Dostupné z: http://cubro.net/index.php/cubro-blog/item/gbit-copper-taps

[3] Cubro. Cubro products. *Cubro* [online]. [cit. 2017-05-07]. Dostupné z: http://www.cubro.net/index.php/cubro-products

[4] McPHERSON, D., R. Raszuk, B. Pithawala, A. Karch, S. Hares. Dissemination of Flow Specification Rules for IPv6. *The Internet Engineering Task Force.* [online]. 19.3.2016 [cit. 2016-10-04]. Dostupné z: *https://tools.ietf.org/html/draft-ietf-idr-flow-spec-v6-07*

[5] Open Networking Foundation. OpenFlow Switch Specification Version 1.5.1 (Protocol version 0×06). *Open Networking Foundation.*[online]. 26.3.2015 [cit. 2016-10-06]. Dostupné z: *https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.5.1.pdf*

[6] EVANS, Steve. The history of OpenFlow. *ComputerWeekly.* [online]. [cit. 2016-10-06]. Dostupné z: *http://www.computerweekly.com/feature/The-history-of-OpenFlow*

[7] OLIVER, Bill. Pica8: First to Adopt OpenFlow 1.4; Why Isn't Anyone Else?. *tom'sIT PRO.* [online]. 2.5.2014 [cit. 2016-10-08]. Dostupné z: *http://www.tomsitpro.com/articles/pica8-openflow-1.4-sdn-switches,1-1927.html*

[8] Open Networking Foundation. SDN/OpenFlow Products. *Open Networking Foundation.* [online]. 2016 [cit. 2016-10-08]. Dostupné z: *https://www.opennetworking.org/sdn-openflow-products*

[9] Open Networking Foundation. Software-Defined Networking (SDN) Definition. *Open Networking Foundation.* [online]. 2016 [cit. 2016-10-08]. Dostupné z: *https://www.opennetworking.org/sdn-resources/sdn-definition*

[10] McNICKLE, Michelle. Five SDN protocols other than OpenFlow. *SearchSDN.* [online]. 28.8.2014 [cit. 2016-10-08]. Dostupné z: http://searchsdn.techtarget.com/news/2240227714/Five-SDN-protocols-other-than-OpenFlow

[11] A Linux Foundation Collaborative Project. Open vSwitch. *A Linux Foundation Collaborative Project* [online]. [cit. 2017-05-07]. Dostupné z: http://openvswitch.org/

[12] Project Floodlight. Floodlight. *Project Floodlight* [online]. [cit. 2017-05-07]. Dostupné z: http://www.projectfloodlight.org/floodlight/

[13] IZARD, Ryan. Floodlight REST API. *Atlassian* [online]. 9.8.2016 [cit. 2017-05-07]. Dostupné z: https://floodlight.atlassian.net/wiki/display/floodlightcontroller/Floodlight+REST+API

[14] Flowmon Networks a.s. Seznam modelů Flowmon kolektorů. *Flowmon Networks a.s.* [online]. 1.7.2016 [cit. 2016-10-12]. Dostupné z: *https://www.flowmon.com/getattachment/df711231-b60b-4567-b303-8db6125483e4/Flowmon-Collector-Spec.aspx*

[15] Flowmon Networks a.s. FLOWMON NETFLOW/IPFIX KOLEKTOR. *Flowmon Networks a.s.* [online]. 2016 [cit. 2016-10-15]. Dostupné z: *https://www.flowmon.com/cs/products/flowmon/netflow-collector*

[16] Flowmon Networks a.s. Flowmon DDoS Defender 3.01.00. *Flowmon.* [online]. 1.12.2016 [cit. 2016-12-04]. Dostupné z: https://flowmon.invea.com/doc/iad_userguide_cz.pdf

Seznam použitých zkratek

ACL	Access Control List
BGP	Border Gateway Protocol
DDoS	Distributed Denial of Service
DSCP	Differentiated Services Code Point
HTTP	Hypertext Transfer Protocol
UDP	User Datagram Protocol
SDN	Software Defined Networks
SNMP	Simple Network Management Protocol
IANA	Internet Assigned Numbers Authority
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ISO	International Organization for Standardization
KVM	Kernel-based Virtual Machine
MAC	Media Access Control
NAT	Network Address Translation
NEL	Network Event Logging
NSEL	Network Secure Event Logging
NLRI	Network Layer Reachability Information
NMS	Network Management Station
OSI	Open Systems Interconnection
PBR	Policy Based Routing
RAM	Random Access Memory
RAID	Redundant Array of Independent Disks
RFC	Request for Comments
RJ – 45	Registered Jack 45
SATA	Serial AT Attachment
SSD	Solid State Drive

SW	Software
TCP	Transmission Control Protocol
TLS	Transport Layer Security
ToS	Type of Service
UUID	Universally Unique Identifier
VLAN	Virtual Local Area Network
QEMU	Quick Emulator

Seznam příloh

A. Disk DVD obsahující elektronickou verzi této práce s názvem Obrana_DDoS_SDN.pdf