VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2017 Bc. Tomáš Gerlich

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

DETEKCE ÚTOKŮ CÍLENÝCH NA ODEPŘENÍ SLUŽEB
DETECTION OF DENIAL OF SERVICE ATTACKS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. Tomáš Gerlich**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. Zdeněk Martinásek, Ph.D.**
SUPERVISOR

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. Tomáš Gerlich ***ID:*** 154726
Ročník: 2 ***Akademický rok:*** 2016/17

NÁZEV TÉMATU:

Detekce útoků cílených na odepření služeb

POKYNY PRO VYPRACOVÁNÍ:

Cílem diplomové práce je nalézt a podrobně porovnat nástroje vhodné pro detekci anomálií síťového provozu na základě behaviorální analýzy a detekce signatur, které způsobují útoky DoS (Denial of Service). Z analýzy současného stavu zprovozněte na experimentálním pracovišti detekce nejznámějších DDoS útoků (vyberte nejméně 5 typů útoků). Cílem práce je výsledky testování vhodně zpracovat (spolehlivost detekce na různé typy (D)DoS tj. SynFlood, UDPFlood, HTTP Flood, pomalé útoky a různá síla) a vybrat nejefektivnější metodu detekce. Na experimentálním pracovišti implementujte pravidlo, které po úspěšné detekci útok mitiguje (např. přesměrováním provozu do předem definovaného jednoduchého filtru). Dosažené výsledky přehledně zpracujte.

DOPORUČENÁ LITERATURA:

[1] MIRKOVIC, Jelena; REIHER, Peter. A taxonomy of DDoS attack and DDoS defense mechanisms. ACM SIGCOMM Computer Communication Review, 2004, 34.2: 39-53

[2] NYCHIS, George, et al. An empirical evaluation of entropy-based traffic anomaly detection. In: Proceedings of the 8th ACM SIGCOMM conference on Internet measurement. ACM, 2008. p. 151-156.

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

Vedoucí práce: Ing. Zdeněk Martinásek, Ph.D.
Konzultant:

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

ABSTRAKT

Diplomová práce je zaměřena na detekci útoků pro odepření služeb (DoS). Tyto distribuované DoS útoky představují hrozbu pro všechny uživatele na Internetu, protom dochází k nasazování detekčních a preventivních systému proti těmto útokům. Teoretická část popisuje DoS útok a jeho nejvyužívanější varianty. Jsou zde zmíněny i varianty pro detekci DoS útoků. Také je zde popsáno jaké nástroje pro detekci DDoS útoků jsou nejvyužívanější. Praktická část se zabývá nasazením softwarových nástrojů pro možnost detekce DoS útoků a vytvoření síťového provozu pro otestování detekčních schopností těchto nástrojů.

KLÍČOVÁ SLOVA

DoS, Detekce DDoS, Suricata, IDS, IPS

ABSTRACT

Master's thesis is focused on intrusion detection for denied of service attacks. These distributed DoS attacks are threat for all users on the Internet, so there is deployment of intrusion detection and intrusion prevention systems against these attacks.The theoretical part describes the DoS attacks and its variants used most frequently. It also mentioned variants for detecting DoS attacks. There is also described, which tools are used to detect DDoS attacks most frequently. The practical part deals with the deployment of software tools for detecting DDoS attacks, and create traffic to test detection abilities of these tools.

KEYWORDS

DoS, Detection of DDoS, Suricata, IDS, IPS

GERLICHÍ, Tomáš *Detekce útoků cílených na odepření služeb*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2017. 60 s. Vedoucí práce byl Ing. Zdeněk Martinásekí, Ph.D.

PROHLÁŠENÍ

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

Brno ....

podpis autora(-ky)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Zdeňku Martináskovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno ....

podpis autora(-ky)

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 diplomové 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(-ky)

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

OBSAH

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

ÚVOD

V dnešní době jsou síťové a webové služby nepostradatelnou součástí každodenního života. Této skutečnosti využívají útočníci k narušení činnosti nebo zamezení přístupu k těmto službám nejen firmám a korporacím. K těmto útokům dochází na mnoho velkých i malých subjektů připojeným k síti Internet. Pro zajištění přístupu k těmto službám musí být nasazen systém určený pro včasnou detekci DDoS útoků, které informují o nutnosti podstoupit protiopatření pro zajištění dostupnosti poskytovaných služeb.

Cílem práce je pomocí IDS/IPS programů detekovat útoky na odepření služby a také zmírnění dopadu útoku na dostupné zdroje. Vybraný detekční systém je použit i pro filtrování provozu při útoku na odepření služeb.

V první kapitole jsou objasněny útoky pro odepření služby a jejich nejvyužívanější typy. Také zde nalezneme zprávy popisující aktuální situaci ohledně těchto útoků. Dále se zde popisuje detekce DoS útoků a rozdíly mezi známými typy detekčních technik jejich výhody i nevýhody.

Druhá kapitola popisuje detekční software a rozdíly mezi vybranými detekčními systémy a různými možnostmi jejich nasazení v reálné síti.

Ve třetí kapitole je popsáno experimentální pracoviště, seznámení se servery sloužícími k detekci, filtraci provozu a generátorem datového provozu síťový tester SpirentAvalanche 3100B.

Čtvrtá kapitola se zabývá testováním detekčních schopností vybraných IDS systémů při různých silách provedení zvolených útoků pro odepření služby.

Pátá kapitola je zaměřena na testování filtračních serverů s IPS systémem a zjištěné výsledky z měření pro popsané typy útoků.

Poslední část seznamuje s možným použitím IDS systému k rozšíření filtračních schopností směrovačů Mikrotik.

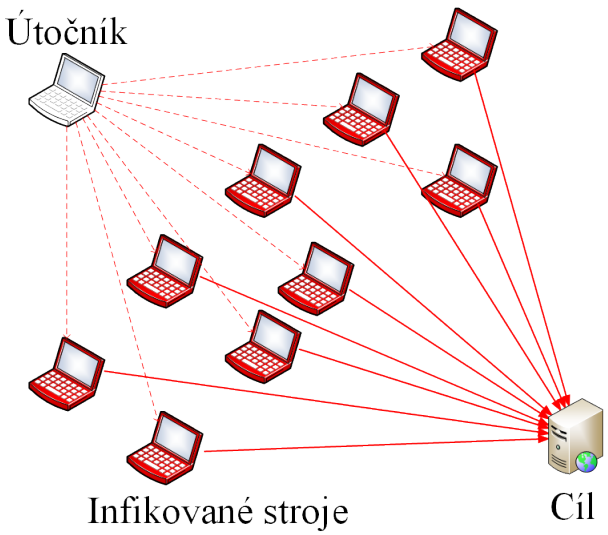
1 DOS ÚTOKY

Útoky pro odepření služby (denied of services) jsou hrozbou pro všechny prostředky připojeny k síti Internet. Jejich účelem je zabránit legitimnímu přístupu k určité službě. Přestože je tato problematika již dlouhou dobu zkoumána, jsou tyto typy útoků velmi využívané pro ochromení cíleného zařízení. Největší výhodou je schopnost i s malými prostředky napáchat velké škody a není nutné mít obsáhlé znalosti o této problematice, jelikož nástroje jsou volně dostupné na Internetu.

Tyto útoky jsou prováděny z různých důvodů. Může jít o poškození konkurenční společnosti pro její oslabení na trhu. Často jsou také využívány jako forma demonstrace na vládních i soukromých webových serverech. V poslední řadě jsou velmi často využívány jako teroristická hrozba.

Dva přístupy pro provedení DoS útoku první odeslání datového toku, který spotřebuje dostupné zdroje a tím dojde k odepření služby ostatním uživatelům nebo druhý způsob, kdy je odeslán poškozený paket pro zmatení aplikace a tím dojde k zamrznutí či restartu buď cílené služby případně celého zařízení. Při těchto útocích je využíváno bezpečnostních děr pro zefektivnění daného útoku.

Při DDoS (distribuovaném DoS) je veden útok z více zdrojů využívá se tzv. zombie, infikovaných zařízení, k následnému využití pro samotný útok. S rostoucím počtem zombie roste i síla daného distribuovaného útoku. Na Obr. 1.1 je znázorněno základní schéma průběhu DDoS útoku. Útočník pouze odesílá zombie informaci o cíli k útoku případně dalších parametrů k provedení útoku. V práci [9] je rozebráno dělení útoků podle stupně automatizace, druhu komunikace, dynamiky datového toku, počtu zombie, druhu cíle případně dopadu na cíl.

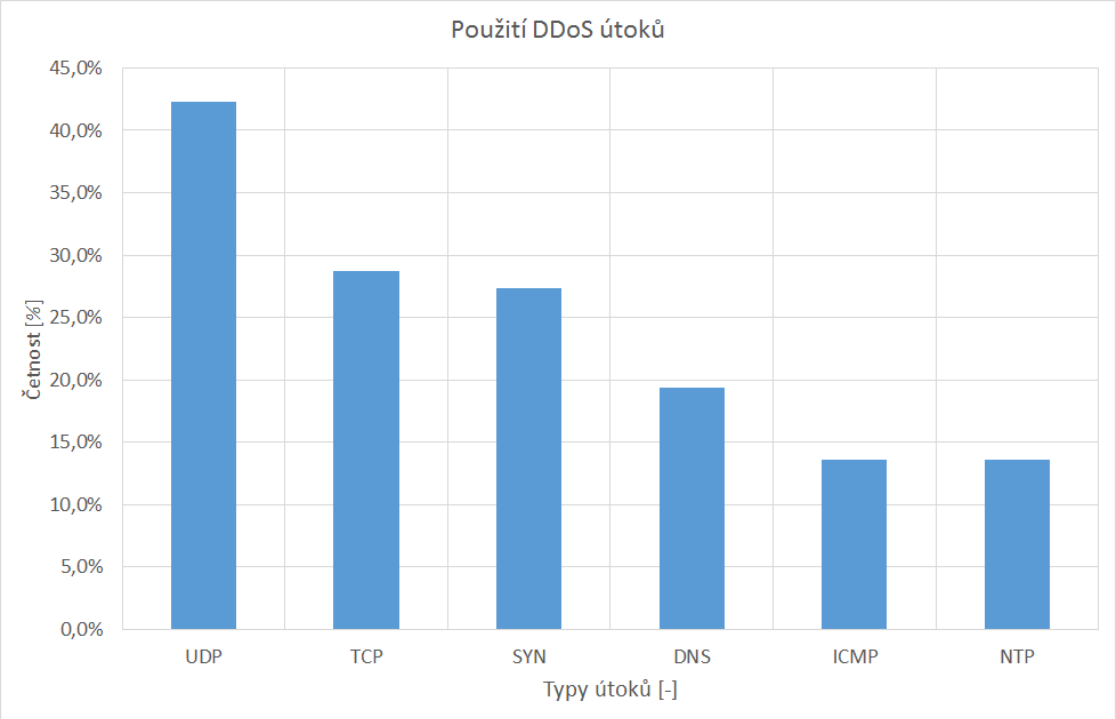


Obr. 1.1: Schéma DDoS útoku

Organizace zabývající se ochranou proti DDoS útoků vydává čtvrtletní zprávy o útocích vedených na zdroje, které chrání před DDoS útoky. Ve zprávě [22] je z Q4 2015 je zjištěno, že přes 80% útoků vedených na síťovou vrstvu byly kratší než 30 minut. Kolem 99% útoků netrvala déle než 6 hodin. Nejvíce provedeným útokem bylo použití TCP, UDP flood a SYN flood. V této době se začal také využívat tzv.více vektorový útok [18] přičemž dochází buď ke změně typu útoku např. DNS flood na UDP flood popřípadě k paralelnímu útoku více typů.

U útoků vedených na aplikační vrstvu však 34% útoků probíhalo mezi 30 minutami až jednou hodinou. Přes 98% všech útoků mělo délku trvání maximálně 24 hodin a 44% systémů, které se obětmi DDoS útoků byly napadeny více než jednou.

Ve zprávě [21] o Q1 2016 je zachyceno zobrazení útoků o síle využívající přes 100 Gb/s až ke 200 Gb/s. Nejen přenosová rychlost, ale i počet paketů způsobuje problémy síťovým prvků, takže graf s počtem paketů je nejčastěji kolem 50 Mp/s, avšak počet paketů dosahoval až k 120 Mp/s. Tyto typy útoků se snaží o zjištění výkonosti směrovačů a přepínačů. Přes 81% útoků trvalo pod 30 minut a 99 % všech útoků trvalo méně než 6 hodin. Využití typů útoků bylo stejné jako v předchozím čtvrtletí s rozdílem zvýšení použití UDP Flood útoků. Nejčastější typy útoků jsou zobrazeny na Obr. 1.2. Došlo také ke zvýšení více vektorových útoků přes 33% všech útoků oproti 24% v předchozí zprávě.



Obr. 1.2: Druhy DDoS útoků

Útoky na aplikační vrstvu nejčastěji trvaly v rozmezí 30 minut až jednou hodinou. U téměř poloviny serverů pod útokem nějakého typu DDoS útoku došlo k jejich napadnutí víceněž jedenkrát.

Zpráva [20] se týká Q2 2016 poukazuje na skutečnost, že jsou preferovány útoky na síťovou vrstvu než na vrstvu aplikační, tyto útoky jsou mnohem více komplexní, nejvyšší zaznamenaný útok byl 470 Gb/s. Délka útoku byla kratší než 30 minut v 76% případů, pro útoky na síťové vrstvě. U více vektorových útoků dochází ke zvyšování jejich použití na 36% ze všech útoků. Tyto sofistikované útoky jsou vedeny zkušenými pachateli. Většina útoků na síťovou vrstvu je vedena jedním druhem útoku neprofesionálními útočníky. Útoky na aplikační vrstvu se stali kratšími, 59% útoků probíhalo pod 30 minut. Pouze 25% útoků trvalo déle než hodinu, v předchozí zprávě 49% útoků trvalo déle než hodinu. Došlo ke snížení opakovaných útoků na 43% z 49,9% zjištěné za předchozí čtvrtletí. V tab. 1.1 je zobrazeno procentuální využití jednotlivých typů útoků.

Tab. 1.1: Používané DDoS útoku

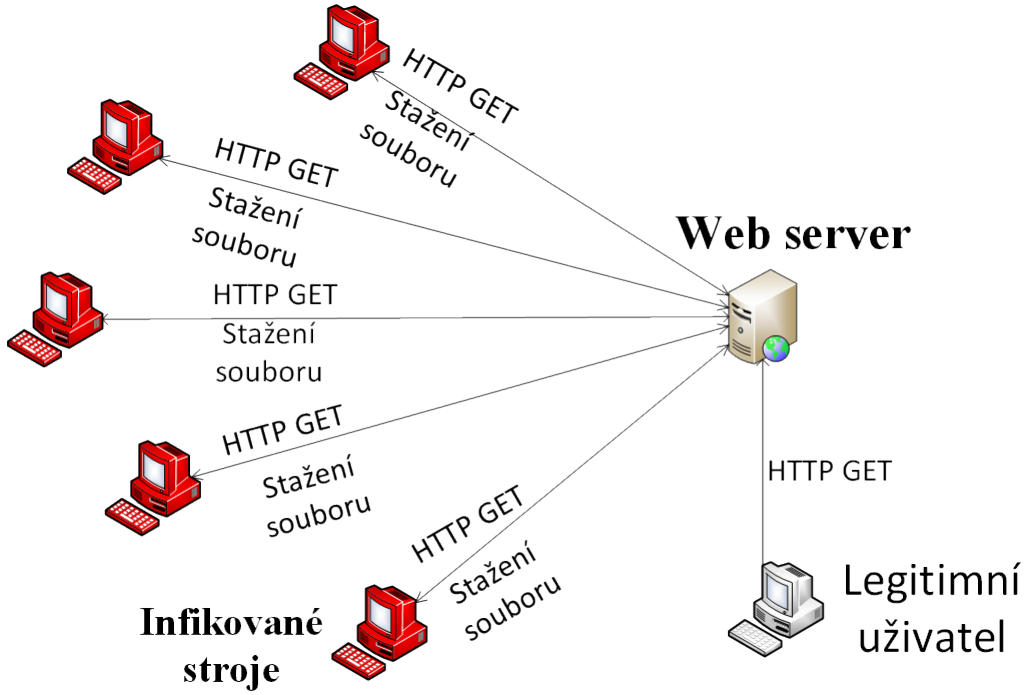
Typ	Četnost použití [%]		
	Q4 2015	Q1 2016	Q2 2016
TCP	44,8	28,7	37
UDP	33,2	42,3	32,9
SYN	18,6	27,3	31,5
ICMP	6,4	13,6	17,2

Jelikož TCP útoky jsou mezi nejvyužívanějšími typy útoků byly vybrány HTTP flood a také XMasTree jako zástupci útoku využívající TCP spojení.

V následující části budou popsány nejčastěji využívané DDoS útoky mezi kterými najdeme SYN flood, UDP flood, HTTP flood, DNS amplification, ICMP Echo Request a útok XMasTree.

HTTP FLOOD

V dnešní době dochází ke zvyšování útoků na aplikační vrstvě, nejčastěji je využit HTTP flood, kdy je aplikován příkaz GET či POST pro opakované stahování obsahu webového serveru Obr.1.3. Cílem útoku je zahlcení webového serveru požadavky od zombie a tak nedochází k obsluze požadavků od legitimních uživatelů. Tyto útoky podle [8] dostávají na oblibě, kdy se zombie vydávají za validní webové prohlížeče. Tento typ útoku je velmi těžké odlišit od legitimního provozu. Pakety mají validní hlavičku i spojení bylo sestaveno správně, takže musí server obsloužit všechny přijaté dotazy od útočníka.



Obr. 1.3: Průběh HTTP flood útok

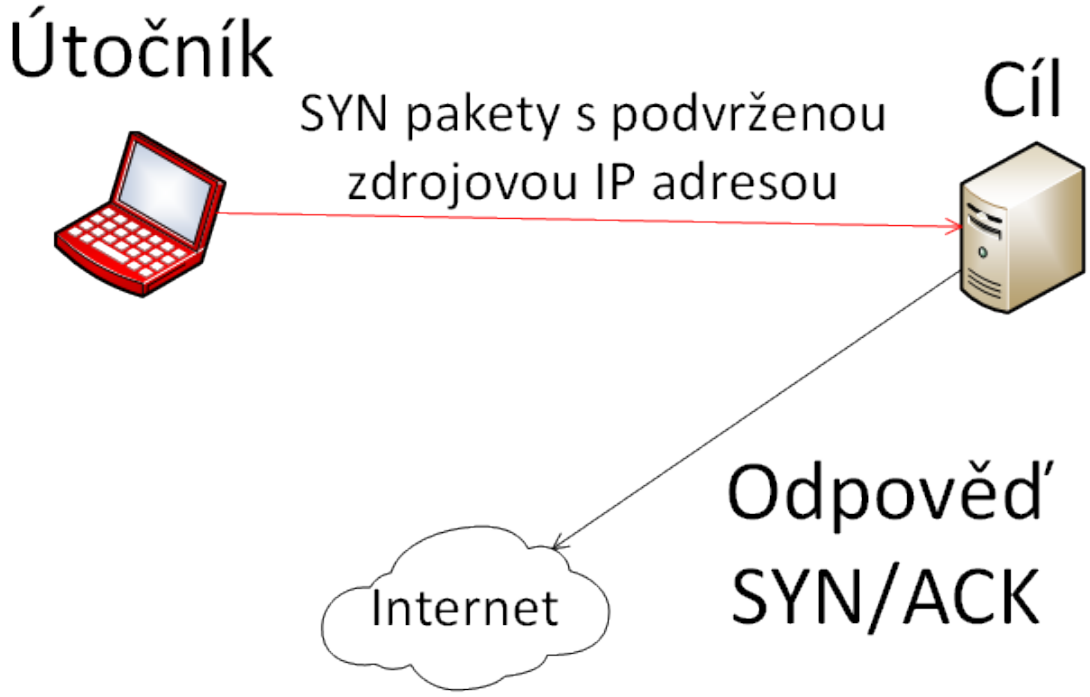
Pro omezení HTTP flood útoku je možné limitovat datový provoz pro uživatele, kteří nadměrně vyžadují stažení stejnou stránky vícekrát za určitou časovou jednotku, bohužel tímto mechanizmem dojde i k omezení legitimního provozu a umožňuje i velkou míru chybné detekce.

V článku [15] jsou navrženy dvě metody založené na strojovém učení, které lze aplikovat k detekci tohoto typu útoku. První metoda dokáže detekovat 78% útoků a jen 1% legitimního provozu je chybně identifikováno. Druhá metoda je založena na korelaci doby prohlížení webové stránky a množstvím informací na stránce. Tato metoda je schopna detekovat 100% útoku a 9% legitimního provozu je identifikováno chybně.

SYN FLOOD

Tento typ útoku využívá nedokončené navázání komunikace pomocí three–way handshake na obr.1.6,vytváří se pouze polootevřeného spojení TCP [2], cíl čeká na potvrzení, které nepříjde obr.1.4. Pokud dojde k vytvoření mnoha těchto spojení, nezbývají poté prostředky pro legitimní provoz.

U mnoha běžných aplikací jako webový server není nastaveno omezení pro vytvořeno spojení s neznámým klientem. Pro úspěšný útok je předpoklad vyhrazení zdrojů pro každý příchozí TCP SYN paket a je možné vytvořit pouze určitý počet.



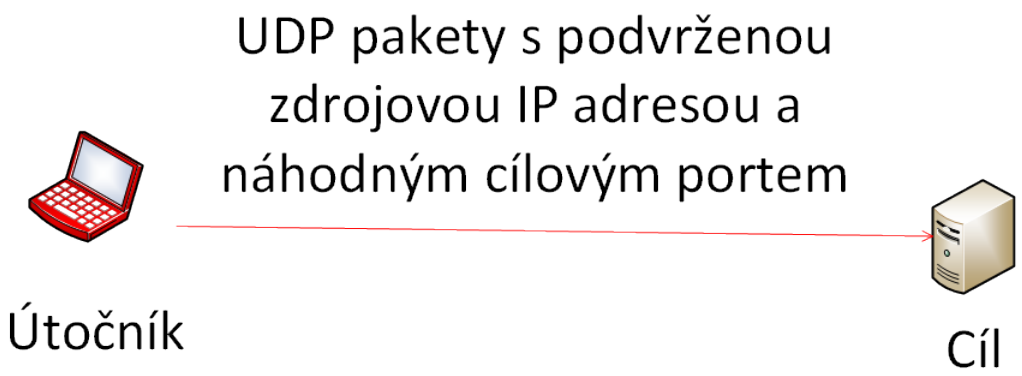
Obr. 1.4: Schéma SYN flood útoku

Tento útok je znám po mnoho let a bylo vytvořeno a nasazeno mnoho technik [23] pro detekci a potlačení SYN flood útoku, jak pro koncové stanice tak i mezilehlé síťové prvky. Některá opatření navrhují zvýšení velikosti fronty pro spojení, snížení časovače pro polootevřená spojení nebo přepis starých spojení ve frontě novými spojeními.

V práci [4] byly tyto metody popsány a nasazeny na experimentálním pracovišti. Nejlepší způsob je využití SYN cookies případně firewallu. SYN cookies nevytváří polootevřené spojení, ale uloží informace do sekvenčního čísla SYN–ACK paketu, pokud není zdrojová IP adresa ze SYN paket podvržena, po obdržení sekvenčního čísla z ACK paketu dojde k navázání spojení.

UDP FLOOD

Použití UDP protokolu pro DoS útok je zasláno velké množství paketů na náhodné cílové porty cílového stroje obr. 1.5. Dochází ke spotřebě nejen šířky pásma síťového spoje. Kvůli náhodným cílovým portům cílový systém musí skoumat která aplikace naslouchá na daném portu případně zasílá odpověď o nedostupnosti cíle. Tyto činnosti zahlcují cílové zařízení a nedostává se prostředků pro nová spojení.



Obr. 1.5: Průběh UDP flood útoku

ICMP FLOOD

Na cílové zařízení je generováno velké množství ICMP paketů s žádostí o ping (echo request) . Utočník se snačí vyčerpat prostředky cílového systému, který nebude schopen obsluhovat jiná spojení. Případně podvrhnul zdrojovou adresu, aby cíl zasílal odpověď na ping (echo response) na skutečný cíl útoku tomuto typu útoku se říká Smurf Attack [12].

XMasTree

V případě tohoto útoku dochází k vytvoření paketu s nastavenými příznaky v TCP hlavičce, buď všemi nebo v neplatných kombinacích. Velmi častá kombinace je používána PSH, FIN, URG. Vznikají tak kombinace, která se v legitimním provozu nevyskytuje. XMasTree se proto dá snadno detekovat pomocí IDS systému případně firewallu. Tyto pakety jsou vždy velmi podezřelé a značí pravděpodobnost zkoumání dané sítě útočníkem. Pokud mají systémy špatně ošetřenou ochranu vůči tomuto útoku může dojít až k havárii celého systému.

Po popisu jednotlivých typů útoků bude vysvětleno, jakým způsobem dochází k detekci těchto útoků, které metody existují, jejich výhody a nevýhody.

Detekce DoS

Největším problémem u DoS útoků je jejich detekce, protože je těžké je odlišit od legitimního provozu. Mnoho existujících detekčních mechanizmů má limitovanou úspěšnost, kvůli útokům využívajících legitimně vytvořených požadavků o spojení a detekce v reálném čase je náročná z důvodu vysokého objemu dat proudících v síti. Detekční medoty jsou rozděleny do dvou skupin:

- detekce signatur
- detekce anomálií síťového provozu

Bude popsán jejich princip a odlišná využitelnost v praktickém nasazení.

Detekce signatur

Přístup pro detekci signatur kontroluje procházející provoz a porovnává jeho vzory se známými útoky. Detekce signatur je více používána, jelikož je jednoduší na implementaci a konfiguraci.

Tato metoda poskytuje velmi dobré detekční schopnosti pro známé útoky, avšak nedokáže detekovat nové druhy napadení. Nutností u tohoto přístupu je pravidelná aktualizace databáze signatur pro detekci i nových typů hrozeb. Snort a Suricata využívají detekce DDoS útoků pomocí detekce signatur. Podobný přístup se využívá i antivirových programů pro detekci napadení.

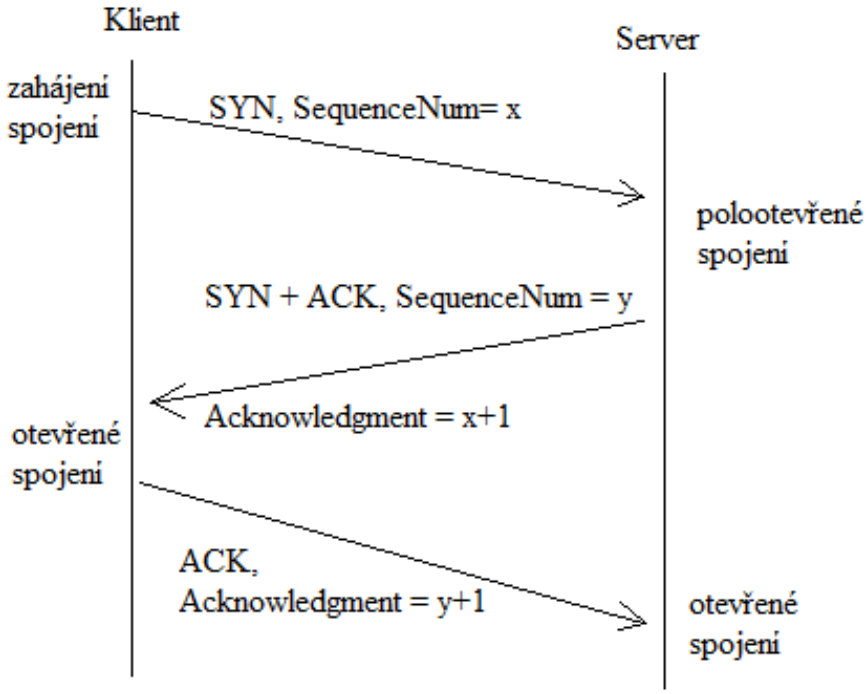
Tento způsob detekce může způsobovat zpomalení systému v důsledku neustále se zvyšujícího množství záznamů v databázi signatur, práce [7] popisuje zrychlení vyhledávání pomocí rozhodovacích stromů.

Detekce anomálií

Tento způsob zkoumá provoz v síti a vyhledává jakékoli odlišnosti od běžného provozu. Pokud je aktuální chování rozdílné od předpokládaného než určený limit dojde k vytvoření upozornění. Změny ve využívání zdrojů vlivem útoků lze detekovat více způsoby statistickou analýzou [10, 5], která se spoléhá na matematické rozdělení síťového provozu. Dalším použitelným způsobem je strojové učení [6], které vytvoří model chování sítě podle zadaných vstupních dat. Tyto vstupní data popisují normální provoz v síti. Porovnáním získaného modelu s aktuálním stavem lze detekovat anomálie síťového provozu

Největší výhodou tohoto přístupu je možnost detekce dříve neznámých útoků, bohužel počet nepravdivých upozornění v těchto systémech je vyšší než u systémů založených na detekci signatur. Tento mechanizmus může využívat specifikace určené standardem nebo sadou pravidel. Patřilo by sem navázání TCP spojení pomocí

three–way handshake. Detekční mechanizmus může využít standard k detekci částečně otevřených spojení a zahodit je. Three-way handshake je znázorněn na Obr. 1.6. Výhodou je absence falešných hlášení. Nevýhodou je, že útok může být proveden komplexněji a projít nedetekován.



Obr. 1.6: Three–way Handshake

Detekční mechanizmy, které používají modely založené na normálním síťovém provozu, využívají limitní hodnoty, pokud dojde k překročení těchto limitů, systém zaznamená vznik anomálie.

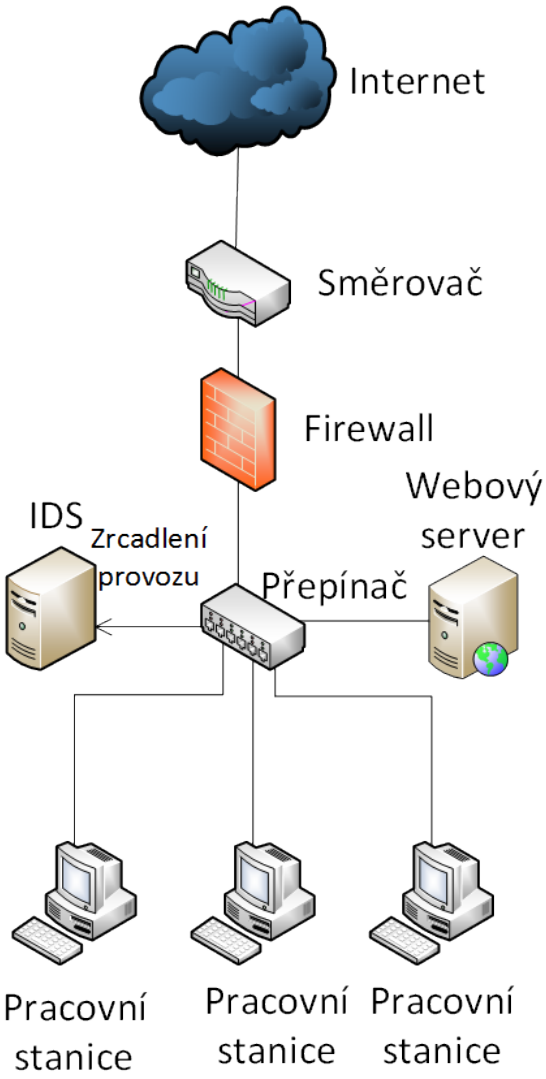
Nevýhody tohoto přístupu je nutnost správného nastavení limitů. Příliš vysoké hodnoty sníží detekční citlivost a při nízkých hodnotách bude docházet a falešným poplachům. Další nevýhodou je nutnost aktualizace modelů, v období, kdy neprobíhá útok. Tento přístup je náchylný pro dlouhotrvající útoky s pomalu rostoucí silou.

V příští kapitole jsou rozepsány detekční a preventivní systémy před DDoS útoky, jejich možnosti nasazení v sítích, rozdíly mezi jednotlivými systémy a důvod jejich nasazení.

2 SYSTÉM DETEKCE PRŮNIKU A SYSTÉM PREVENCE PRŮNIKU

V počítačových sítích je zabezpečení řešeno pomocí firewallu ve vstupním bodě do sítě, ale firewall musí povolit provoz na určitých portech do a ven ze sítě. Pravidla firewallu nemusí dostačovat k detekci, zda daný provoz je legitimní nebo se jedná o útok.

Přístup k webovému serveru je umožněn přes TCP port 80, útočník tak může využít tento port pro provedení útoku. IDS (Intrusion detection system) [13] může rozhodnout, zda provoz patří k legitimnímu provozu nebo se jedná o útok na webový server. Případ tohoto zapojení na obr. 2.1.



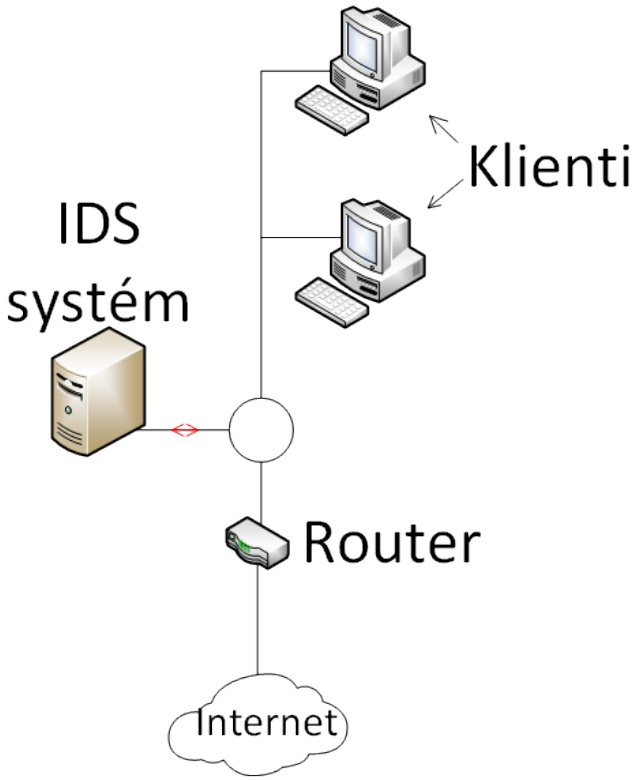
Obr. 2.1: Zapojení IDS systému v síti LAN

IDS [3] po detekování podezřelého provozu vytvoří varování, které je zapsáno do logovacího souboru a dojde k upozornění správce systému. IPS (Intrusion prevention system) [3] při detekci útoku provede protiopatření samostatně, bez nutnosti zásahu, spojení bude zamítnuto případně zahozeno.

IDS respektive IPS jsou síťové ochrany nasazeny v počítačových sítích po celém světě. Tyto systémy provádějí inspekci paketů, stavovou analýzu, znovu sestavení TCP segmentu, ověření funkce protokolu a porovnání signatur.

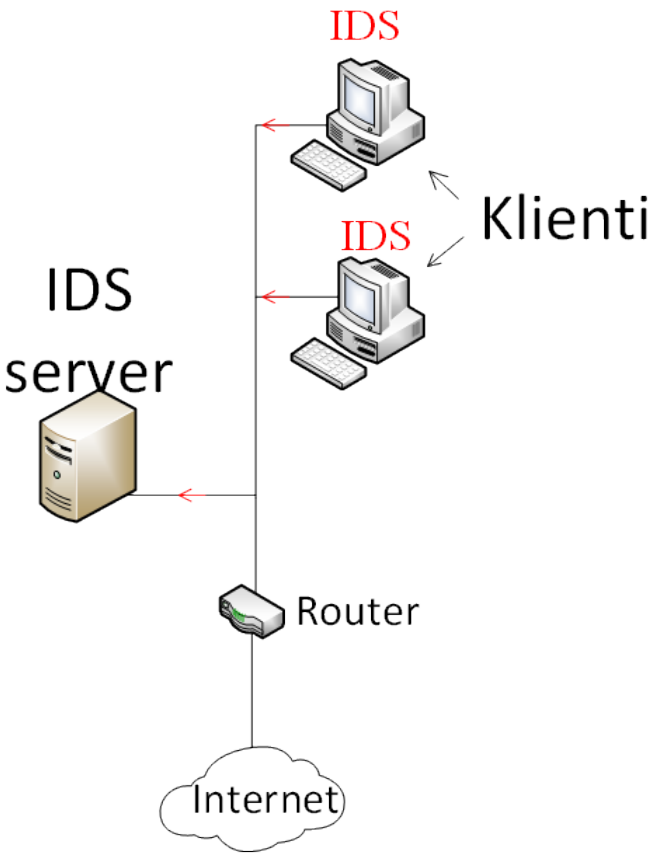
Rozdíl v těchto technologiích je v jejich provedení ochrany v nasazené síti. IDS vytváří pouze upozornění, pokud je zaznamenána hrozba. Dochází pouze k detekci možných narušení a oznámení administrátorům. IDS má vlastní sady pravidel, které dohlíží na události a procesy příbuzné k operačnímu systému, na kterém běží (HIDS, Hosted based Intrusion Detection System), také mohou analyzovat aktivitu na síti a porovnávat ji se známými útoky a vzory (NIDS, Network based Intrusion detection system), tento typ je znázorněn na Obr. 2.2. Síťový provoz je zaznamenáván, ale IDS není umístěn mezi síťové prvky, funguje jako sonda. HIDS nasazení je na Obr. 2.3 zobrazeno i se serverem, který shromažďuje informace z jednotlivých IDS systémů.

Výpočetní kapacita systému by měla odpovídat průměrné zátěži sítě, odezva systému může dosahovat až k několika minutám. IDS zaznamenává provoz sítě, proto musí obsahovat velkou vyrovnávací paměť, kvůli zaznamenání síťové zátěže.



Obr. 2.2: Schéma NIDS

IPS jsou nasazeny mezi síťovými prvky, mohou tak vykonat určitou akci na podezřelou aktivitu. Dochází tak nejen k detekci, ale také možnosti ukončení spojení. Poskytuje ochranu pro síť, zdroje a data. IPS je nasazeno přímo mezi síťové prvky výkon tohoto systému musí zvládat špičkové zatížení sítě. Zpoždění IPS musí být v řádu mikrosekund vzhledem k nutnosti rychlé odezvy.



Obr. 2.3: Schéma HIDS

IDS může minimálně identifikovat normální provoz jako nežádoucí, avšak IPS systém se této chyby dopustit nesmí. IDS pouze oznámí možné napadení, IPS však tento nežádoucí provoz zakáže. Nedetekování útoku může nastat, pokud bude IDS systém zahlcen nad jeho kapacitu, může dojít k zahazování paketů nutných pro detekci útoku. U IPS při zahlcení může docházet k zablokování nebo zahození spojení a tím předejde úspěšnému útoku.

Nyní budou popsány někteří zástupci open source IDS a IPS systémů, jejich výhody a možnosti.

Snort

Snort [25] je velmi známý jednovláknový IDS systém. Má vysoký podíl na trhu kvůli jeho vysoké stabilitě a dobré detekci škodlivých paketů. Je založen na detekci signatur a je dostupný zdarma do licencí GPL, funguje pod operačními systémy Windows i GNU/Linux. Bohužel velká šířka pásma síťového provozu způsobuje u jednovláknových IDS systému limit v jejich zpracování [1].

Snort se skládá z paketového snifferu, který odposlouchává komunikaci, preprocesoru kontrolujícího chování paketů. Další část je detekční engine porovnávající pakety s pravidly a poslední část upozorní uživatele nebo zaznamená upozornění do databáze. Snort může být nasazen přímo na cílovém stroji (HIDS) nebo v síti (NIDS). Pracuje jak v IDS režimu tak i jako IPS systém.

Suricata

Suricata [26] je architekturou podobná Snortu, ale je implementována jako více vláknová to umožňuje zvýšit schopnost přijmutí paketu, nedojde k ignorování paketu kvůli omezené kapacitě [11].

Suricata byla navržena, tak aby nahradila nevýhody Snortu ve více jádrových systémech. Suricata má kompatibilní pravidla s Snortem, je schopna akcelerace pomocí GPU a jiné funkce. Umí pracovat jako IPS i jako IDS systém. Může být nasazena na hostu jako HIDS, nainstalována přímo na hlídaném serveru či je nasazena přímo v síti jako NIDS.

Bro

Bro [17] je open-source NIDS, který pasivně monitoruje síť a vyhledává podezřelé aktivity. Ke spuštění Bro v síti je nutný stroj s operačním systémem Linux, je možné ho provozovat i na systému Solaris nebo FreeBSD. Bro cílí na vysokorychlostní sítě. Je určen pro sítě požadující flexibilitu, a vysokou přizpůsobitelnost detekčního systému.

Využívá složitější signatury než jiné IDS systémy. Má schopnost provádět více vrstvou analýzu, monitorování chování a zaznamenávání síťové aktivity. Umí detekovat jak specifické útoky definované signaturami, tak i neobvyklou aktivitu.

OSSEC

OSSEC [19] je open source HIDS, je označen jako multiplatformní, běží na většině dnešních operačních systémů Windows, Linux, FreeBSD, OpenBSD Solaris a

MacOS.Má výkonný korelační a analytický engine. Integrovanou analýzu logů a upozornění v reálném čase s aktivní odezvou.

OSSEC může být nainstalován samostatně k monitorování jediného klienta nebo nasazen na více klientů, kdy jeden bude mít roli serveru, a ostatní budou agenty. Je také obohacen o preventivní mechanizmy, které umožnují reagovat na specifické události a provést aktivní opatření. Systém má předefinovaná určitá opatření, ale administrátor může přidávat vlastní.

Samhain

Samhain [24] je HIDS systém s architekturou postavenou na modelu klient a server. Klienti jsou nainstalování na koncových stanicích a kontrolují nejen síťový provoz, ale v případě průniku do systému kontrolují, které soubory byly vytvořeny případně změněny. Server slouží jako uložiště logu, obsahuje konfigurační soubory a ověřuje připojené klienty. Klienti při spuštění stahují konfiguraci ze serveru, provedou kontrolu a hlásí serveru anomálie.

Tab. 2.1: Vlastnosti detekčních systémů

Název	Režim IPS	Využití více-vláken	Centralizace	Popora NIDS	První verze
Snort	ANO	NE	NE	ANO	1998
Suricata	ANO	ANO	NE	ANO	2009
Bro	NE	NE	NE	ANO	2003
OSSEC	NE	NE	ANO	NE	2003
Samhain	NE	NE	ANO	NE	2001

Jako vhodný systém pro detekci DDoS útoků na experimentálním pracovišti byl vybrán nástroj Suricata z důvodu kvalitní dokumentace, podpoře programů třetích stran a možnostem nasazení jako IDS i IPS systém.

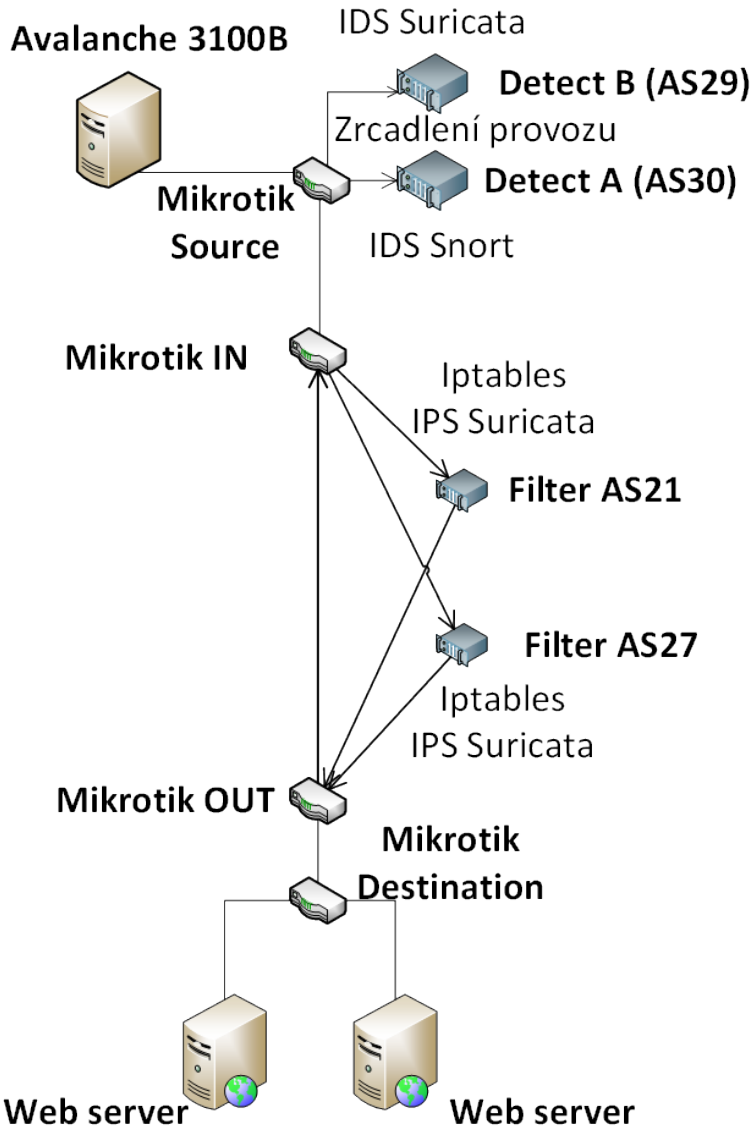
Druhým použitým systémem pro detekci útoků byl vybrán systém Snort. Snort je léty ověřený systém s dostupnou a kvalitní dokumentací nejen od vývojářů, ale také rozsáhlé komunity. Umožňuje snadnou a přehlednou konfiguraci s možnostmi rozšíření o přídavné moduly.

V následující části práce bude popsána topologie experimentálního pracoviště, typy nasazených serverů a provedena detekce útoků pomocí vybraných IDS systémů.

3 EXPERIMANTÁLNÍ PRACOVIŠTĚ

Praktická část této práce byla provedena na experimentálním pracovišti s využitím infrastruktury fakulty eletrotechniky a komunikačních techlonogií VUT Brno. Experimentální pracoviště, zobrazeno na Obr. 3.1, je rozděleno podle účelu na servery k filtraci provozu dále na detekci a ohlášení podezřelého provozu. Jako cíle provozu jsou v pracovišti využity dva webové servery. Pro generování provozu je použit síťový tester Spirent Avalanche 3100B. Na propojení jednotlivých serverů jsou využity přepínače Mikrotik, slouží i pro měření propustnosti datového provozu.

Přepínač Mikrotik před filtračními prvky rozhoduje o cestě paketu podle cílové IP adresy tím je dosaženo posílání provozu přes konkrétní filtrační server.



Obr. 3.1: Topologie experimentálního pracoviště

Na používaných serverech experimentálního pracoviště je využívána distribuce Debian operačnícho systém Linux. V tab. 3.1 jsou uvedeny hardwarové parametry jednotlivých serverů použitých v experimentálním pracovišti.

Tab. 3.1: Hardwarové parametry serverů

	Webserver	Filter AS27	Filter AS21	Detekční server
Procesor	AMD Opteron 148, 2194 MHz	Intel Xeon CPU 3.40GHz	Intel Xeon L5520 @ 2.26GHz	Intel Xeon E5420 @ 2.50GHz
Vlákna	1	4	16	8
Pevný disk	149 GB	135 GB	408 GB	68 GB
Nic#1	Broadcom NetXtreme BCM5704 Gigabit Ethernet	Compaq NC7782 Gigabit Server	Broadcom NetXtreme II BCM5709 Gigabit Ethernet	Broadcom NetXtreme II BCM5708 Gigabit Ethernet
Nic#2	Broadcom NetXtreme BCM5704 Gigabit Ethernet	Intel PRO/1000 MT Dual Port Server	Intel Ethernet Converged Network Adapter X540-T2	HP NC364T PCI Express Quad Port Gigabit Server

Všechny filtrační servery jsou opatřeny stejným hardwarovým vybavením, pouze filtr **AS21** je z hlediska hardwarového vybavení výkonější než ostatní servery zapojené v pracovišti. **AS21** obsahuje dva 4 jádrové procesory Intel Xeon L5520. Celkově server disponuje 12 vlákny. K dispozici je 49 GB operační paměti. Ostatní obsahují dva jednojádrové procesory Intel Xeon CPU 3.40GHz každy procesor má dvě vlákna na fyzické jádro. Operační paměť u těchto serverů je 4 GB. Dalším rozdílem mezi filtračními servery jsou verze operačního systému.

Filtrační server **AS27** používá systém Debian verze `3.16.36-1+deb8u1`, oproti tomu filtrační server **AS21** má instalován Debian verze `4.9.18-1 bpo8+1`

Každý server obsahuje dvě síťové karty, první je pro management daného serveru, druhá je využita pro filtraci provozu. Síťové karty pro filtraci mají více ethernetových rozhran jsou využity jako vstupní a výstupní rozhraní pro datový provoz.

Pro zajištění průchodu datového provozu byl nad filtračními rozhraními vytvořen bridge sloužící pro přeposílání provozu k cíli. U **AS21** byl bridge vytvořen pomocí balíčku iproute2 zatím co u **AS27** byl bridge vytvořen s využitím staršího balíčku bridge-utils [16]. Každý bridge dostal IP adresu nutnou ke správnému směrování

provozu ze vstupního přepínače Mikrotik. Odlišné vytvoření bridge může bude znamenat rozdíl ve provedených meřeních.

Filtrační server slouží k filtrování datového provozu, který dále směřuje k cílovému webovému serveru, tento provoz je pouze jednosměrný, opačný směr je veden jinou cestou není nutné, aby byl zatěžován filtr komunikací od webového serveru. Jako filtrační nástroj byl použit program iptables a dále i suricata v režimu IPS.

Iptables

Iptables nastavuje seznam pravidel pro paketový filtr v sytému Linux. Seznamy pravidel jsou uloženy v tzv chainech které popisují zachazení s pakety. Pravidla popisují, jaká akce je s paketem provedena při schodě s pravidlem. V první části testů jsou všechny chainy bez pravidel a veškerý provoz je povolen. Použitá pravidla v dalších částech této práce pro filtraci provozu jsou popsána v příloze A.1.

Detekční server byl použit jako detekce podezřelého provozu, datový provoz byl pomocí port mirroringu kopírován na vstupní rozhraní detekčního serveru. Příchozí datový provoz byl kontrolován pomocí IDS systému suricata a snort. V operačním systému byl povolen veškerý provoz, aby se předešlo zahazování paketů. Vstupní rozhraní bylo provozováno v promiskuitním režimu ke zpracování všech příchozích paketů IDS systémem.

Suricata

Suricata byla nainstalovaná podle dokumentace dostupné na oficiálních webových strankách projektu [26]. Pravidla pro detekci útoků a podezřelého provozu byla nastavena při instalaci nastroje Suricata. Pravidla jsou uložena v adresáři */etc/suricata/rules* výpis výchozích pravidel je na obr. 3.2. Při konfiguraci byly nastaveny patřičné rozsahy IP adres a případné určení v jakém režimu IPS bude Suricata pracovat. Zjištěné útoky jsou zapsány v logovacím souboru uloženém na lokálním disku serveru na kterém Suricata běží. Pro zasílání upozornění v rámci sítě by je bylo možné posílat pomocí syslog služby k dalšímu zpracování.

```
root@debian:~# ls /etc/suricata/rules
app-layer-events.rules          emerging-dns.rules              emerging-rpc.rules              http-events.rules
botcc.portgrouped.rules         emerging-dos.rules              emerging-scada.rules            LICENSE
botcc.rules                     emerging-exploit.rules          emerging-scan.rules             modbus-events.rules
BSD-License.txt                 emerging-ftp.rules              emerging-shellcode.rules        rbn-malvertisers.rules
ciarmy.rules                    emerging-games.rules            emerging-smtp.rules             rbn.rules
classification.config           emerging-chat.rules             emerging-snmp.rules             reference.config
compromised-ips.txt             emerging-icmp_info.rules        emerging-sql.rules              sid-msg.map
compromised.rules               emerging-icmp.rules             emerging-telnet.rules           smtp-events.rules
decoder-events.rules            emerging-imap.rules             emerging-tftp.rules             stream-events.rules
dnp3-events.rules               emerging-inappropriate.rules    emerging-trojan.rules           suricata-1.3-etpro-etnamed.yaml
dns-events.rules                emerging-info.rules             emerging-user_agents.rules      suricata-1.3-open.txt
drop.rules                      emerging-malware.rules          emerging-voip.rules             suricata-1.3-open.yaml
dshield.rules                   emerging-misc.rules             emerging-web_client.rules       tls-events.rules
emerging-activex.rules          emerging-mobile_malware.rules   emerging-web_server.rules       tor.rules
emerging-attack_response.rules  emerging-netbios.rules          emerging-web_specific_apps.rules unicode.map
emerging.conf                   emerging-policy.rules           emerging-worm.rules
emerging-current_events.rules   emerging-pop3.rules             gen-msg.map
emerging-deleted.rules          emerging-p2p.rules              gpl-2.0.txt
```

Obr. 3.2: Seznam pravidel použitých IDS Suricata

Snort

Snort byl nainstalován na druhém detekčním serveru s pomocí dokumentace poskytnuté na webových stránkách projektu [25]. Kofigurace byla provedena v souladu s dokumentací poskytovanou na webu Snortu. Pravidla použitá k detekci útoků jsou stažena opět ze stránek projektu Snort, tyto pravidla jsou pro registrované uživatele zdarma, jsou však k dispozici i obsáhlejší pro platící zákazníky. Pravidla jsou uložena v */etc/snort/rules* seznam pravidel ne vypsán ja obr. 3.3. Byl nastaven jako IDS systém pro kontrolu příchozího provozu na detekční server **detectA**.

```
toor@detect-a:~$ ls /etc/snort/rules
app-detect.rules          file-image.rules            nntp.rules              protocol-rpc.rules        smtp.rules
attack-responses.rules    file-java.rules             oracle.rules            protocol-scada.rules      snmp.rules
backdoor.rules            file-multimedia.rules       os-linux.rules          protocol-services.rules   specific-threats.rules
bad-traffic.rules         file-office.rules           os-mobile.rules         protocol-snmp.rules       spyware-put.rules
black_list.rules          file-other.rules            os-other.rules          protocol-telnet.rules     sql.rules
blacklist.rules           file-pdf.rules              os-solaris.rules        protocol-tftp.rules       telnet.rules
botnet-cnc.rules          finger.rules                os-windows.rules        protocol-voip.rules       tftp.rules
browser-chrome.rules      ftp.rules                   other-ids.rules         pua-adware.rules          virus.rules
browser-firefox.rules     icmp-info.rules             p2p.rules               pua-other.rules           voip.rules
browser-ie.rules          icmp.rules                  phishing-spam.rules     pua-p2p.rules             VRT-License.txt
browser-other.rules       imap.rules                  policy-multimedia.rules pua-toolbars.rules        web-activex.rules
browser-plugins.rules     indicator-compromise.rules  policy-other.rules      rpc.rules                 web-attacks.rules
browser-webkit.rules      indicator-obfuscation.rules policy.rules            rservices.rules           web-cgi.rules
chat.rules                indicator-scan.rules        policy-social.rules     scada.rules               web-client.rules
content-replace.rules     indicator-shellcode.rules   policy-spam.rules       scan.rules                web-coldfusion.rules
ddos.rules                info.rules                  pop2.rules              server-apache.rules       web-frontpage.rules
deleted.rules             local.rules                 pop3.rules              server-iis.rules          web-iis.rules
dns.rules                 malware-backdoor.rules      protocol-dns.rules      server-mail.rules         web-misc.rules
dos.rules                 malware-cnc.rules           protocol-finger.rules   server-mssql.rules        web-php.rules
experimental.rules        malware-other.rules         protocol-ftp.rules      server-mysql.rules        white_list.rules
exploit-kit.rules         malware-tools.rules         protocol-icmp.rules     server-oracle.rules       x11.rules
exploit.rules             misc.rules                  protocol-imap.rules     server-other.rules
file-executable.rules     multimedia.rules            protocol-nntp.rules     server-samba.rules
file-flash.rules          mysql.rules                 protocol-other.rules    server-webapp.rules
file-identify.rules       netbios.rules               protocol-pop.rules      shellcode.rules
```

Obr. 3.3: Obsah adresáře s pravidly pro IDS Snort

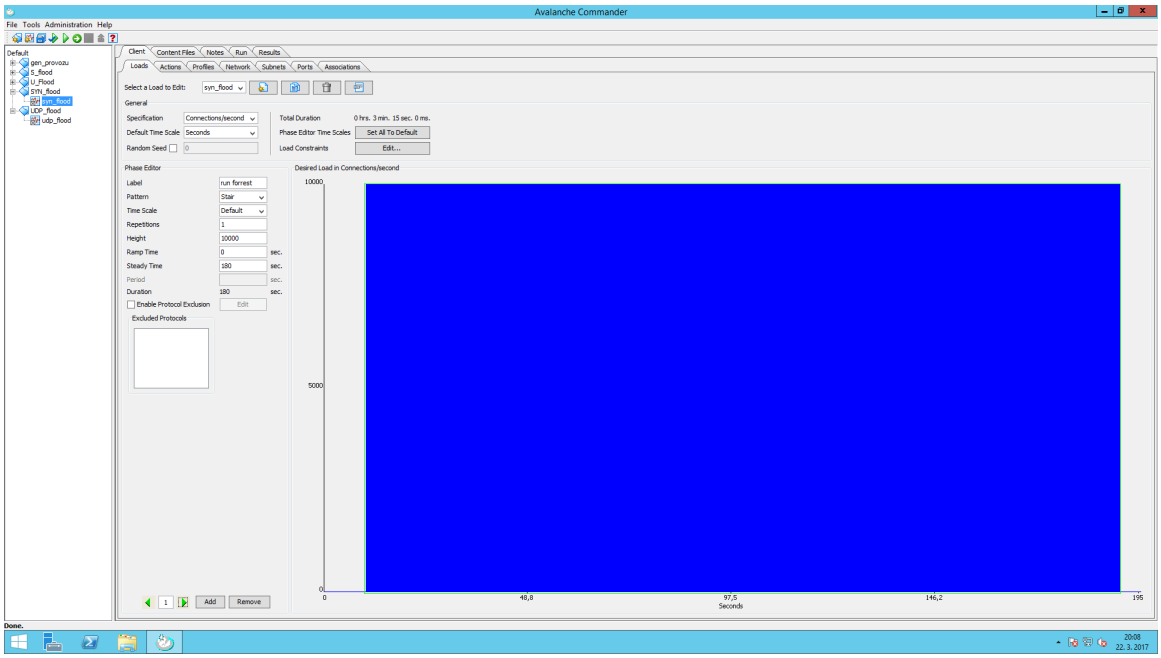
Pro cíl DDoS útoků byly využity dva webové servery, každý byl vybaven několika IP adresami pro jednoduší směrování provozu přes filtrační servery. Na tyto cílové servery byl nainstalován webový server Apache s možností přístupu přes všechna rozhraní.

Spirent Avalanche 3100B

Síťový tester Avalanche 3100B je zařízení poskytující testy pro síťovou infrastrukturu, webové služby a další. Je schopno simulovat nejen stranu klienta, ale také i serverovou stranu a současně na odlišných portech. Je schopno specifikovat druhy spojení, identifikovat zpoždění v síti a ztrátovost paketů. Obsahuje informace o proběhlých simulacích s jejich zjištěnými závěry.

Práce se síťový testerem je realizována skrz uživatelské grafické rozhraní, která slouží ke specifikaci simulace a její monitorování. Na Obr. 3.4 je zobrazeno nastavení pro klientskou stranu a její zátěžový profil. Určení záteže se v poli Load Specification na výběr více možností v této práci budou využity Connections/second označuje počet tcp spojení za jednotku času. Jedno spojení může obsahovat i stovky transakcí vše záleží na nastavení profilu a odpovědi serverové strany. Transactions/second definuje a udržuje počet generovaných transakcí za sekundu. Jedna transakce popisuje přenesení jednoho objektu typicky webové stránky, specifická pro protokol HTTP. SimUsers/second udržuje zadaný počet virtuálních uživatelů za sekundu vykonávajících příkazy z Action listu. Při použití tohoto nastavení odpovídá nastavená hodnota počtu paketů za sekundu.

Dalším nastavením je maximální zátěž označuje velikost vybrané specifikace. Celková doba testu je dána součtem délek trvání jednotlivých časových úseků.



Obr. 3.4: Grafické rozhraní pro Avalanche 3100B

V záložce Action je prováděna definice požadovaných útoků, včetně IP adresy cíle nastavení cílových a zdrojových portů, případně MAC adres. Subnet je záložka použita pro specifikaci rozsahů IP adres použitých jako zdrojové v případě, že nejsou pro daný útok použity náhodné IP adresy. Blok Port slouží pro výběr portu, který bude použit na generování datového provozu. Associations je posledním blokem nutným před zahájením generování zátěže, dochází zde k provázání předchozích nastavení, výběr profilu zátěže, typu vykonávané akce, přiřazení portu použitého k generování provozu a volba rozsahu zdrojových IP adres.

Následující kapitola je věnována detekčním schopnostem IDS systémů při různých typech DDoS útoků.

4 ZÁTĚŽOVÉ TESTOVÁNÍ PŘI DDOS ÚTOCÍCH

Tato část práce je zaměřena na dosažené výsledky po provedení různých druhů útoků na webový server a detekování jednotlivých útoků. Při provádění útoků popsaných v této části nebyla prováděna filtrace provozu na žádném filtračním serveru. Detekce útoků je prováděna dvěma nezávislými servery na každém je použit jiný IDS systém s odlišnými pravidly použitými k detekci útoků. Na oba servery je zrcadlen provoz z síťového testeru Avalanche

Detekční server **detect A** je používán s IDS systémem Snort, na detekčním serveru **detect B** je pro detekci použit IDS systém Suricata. Pravidla pro detekci podezdřelého provozu byla zpočátku použita bez jákýchkoliv úprav. Následně byly buď původně zakázené pravidla povolena nebo přidána nová pro zlepšení detekce. Povolená pravidla pro oba IDS systémy jsou vypsána v příloze A.2.

Doba trvální vybraných útoků byla zvolena s ohledem na vydané zprávy [22, 21, 20]. Většina útoků byla krátkého trvání a proto byly použity kratší testy zpravidla do 5 minut.

Dále jsou rozepsány použité DDoS útoky, popsáno jejich působení na různé servery nasazené v experimentálním pracovišti.

SYN Flood

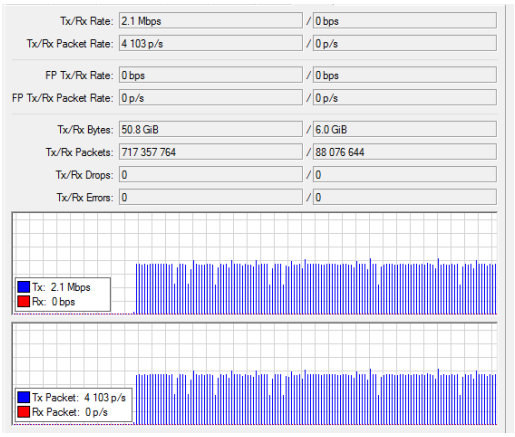
Syn flood útok byl generován testerem Avalanche byl veden proti webovému serveru s průchodem přes filtrační server. Při útoku byly posílány pakety o velikosti 60 byte . Při provádění tohoto útoku je použito podvržených adres, proto odpověď na SYN paket nemůže dojít ke zdroji útoku. V tab. 4.1 je zazmenanán datový provoz generovaný testerem Avalanche, označení filtračnho serveru, přes který byl provoz veden a zjištěný datový provoz po průchodu daným filtrem.

Tab. 4.1: Parametry útoků SYN flood

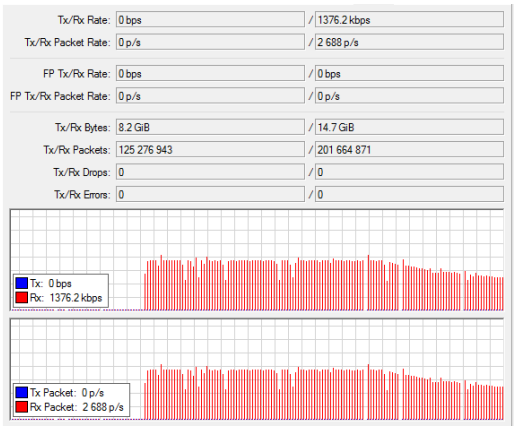
	Generováno Avalanchem			Po průchodu filtračním serverem	
počet uživatelů	Přenosová rychlost Mb/s	Počet paketů za sekundu	Filtrační server	Přenosová rychlost Mb/s	Počet paketů za sekundu
2000	1	2000	AS27	1	2000
3500	1,7	3500	AS27	0,8	1500
4100	2	4100	AS27	1,4	2700
4100	2	4100	AS21	2	4100
10000	5	10000	AS21	1,2	2500
164300	83,5	163140	AS27	1,3	2540
164300	83,7	163657	AS21	4,8	9432

Při provádění útoků s nižší silou nedocházelo k žádnému zatížení procesoru či ztrátě pateků. Při útoku s datovým provozem 2000 p/s nedocházelo k ztrátě paketů po celou dobu útoku, avšak po čtvrté minutě útoku došlo k vytěžování jednoho jádra procesoru na 94%, avšak po překročení počtu 3500 p/s, začalo docházet k nežádoucímu zahazovaní paketů a k náhlému vytížení jednoho jádra procesoru na 100%.

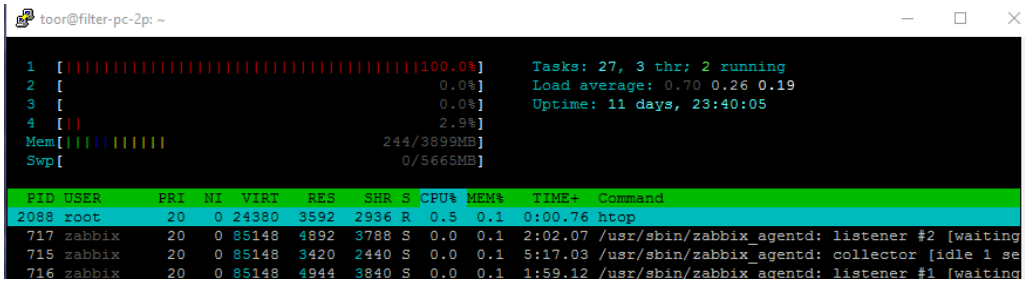
Při průchodu SYN flood útoku o síle 10 000 p/s filtračním serverem **AS27** je provoz odesílán přepínačem do filtru na Obr. 4.1a, vytížení procesoru tohoto serveru na Obr.4.1c a provoz, který je přijímán přepínačem po průchodu filtračním serverem na obr.4.1b.



(a) Příchozí SYN flood útok



(b) Výstupní datový provoz po průchodu



(c) Zatížení filtračního serveru

Obr. 4.1: SYN flood při průchodu filtračním serverem

Pokud doba mezi jednotlivými útoky byla kratší než 2 minuty, docházelo při druhém útoku okamžitě k zahazování datového provozu a plnému vytížení procesoru. Nejpravděpodobněji k tomuto jevu docházelo z důvodu zbývajících paketů z předchozího útoku ve vyrovnávací paměti.

Změnou filtračního serveru za server **AS21** nedocházelo k vytížení procesoru ani zahazování do počtu 4100 p/s. Při útoku 10 000 p/s, tento filtrační server propouštěl

útoku v celé síle po dobu 90 sekund a následně došlo k vytížení všech jáder procesoru a omezování propustnosti datového provozu až k hodnotě 2500 paketů za sekundu.

Detekční servery při SYN flood útoku kontrolovaly všechen datový provoz i přes neschopnost přesné detekce, nedocházelo k žádnému výraznému zatížení procesoru. Jediný scénář, kdy byly procesory detekčnch serverů značně zatížený byl při útoku s nejvyšší silou. Systém Snort vytěžoval přiřazené vlákno na 100%, u druhého serveru s IDS Suricata byl procesor vytěžován při útoku o této síle kolem 60%.

Suricata byla schopna tento typu útoku detekovat pouze na základě zdrojových IP adres, které byly v pravidlech zapsány jako nespolehlivý zdroj, ty byly přidělovány paketům testerem Avalanche náhodně. Docházelo tak k detekci nepravidelně a častěji při větších útocích, neboť tyto nedůvěryhodné IP adresy byly častěji přiřazovány paketům.

Snort dokázal identifikovat probíhající SYN flood útok při od útoku 4100 p/s popsaném v tab. 4.1. Na obr. 4.2 je zpráva generovaná programem Snort při detekci SYN flood útoku

```
05/14-14:28:35.722142  [**] [129:15:1] Reset outside window [**] [Classification: Potentially Bad Traffic] [Priority: 2]
 {TCP} 192.168.2.132:37067 -> 192.168.2.151:80
```

Obr. 4.2: Uporoznění na SYN flood programem Snort

Pro zajištění detekce bylo vloženo pravidlo, jak do pravidel pro systém Suricata, tak do pravidel systému Snort. Jelikož program Suricata vychází ze programu Snort, nebylo nutné pravidlo nijak modifikovat, či jej dále upravovat pro možnost použití.

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 80 (msg:"rychly prichod SYN
paketu na port 80, potencionalni DOS"; flags: S,12; threshold: type both,
track by_dst, count 5000, seconds 5; classtype:misc-activity; sid:5;)
```

Pravidlo kontroluje zda počet paketů s příznakem SYN přicházející na port 80 nepřesahuje hodnotu 5000 za dobu 5 sekund, tato hodnota je zvolena orientačně pro umožnění legitimní komunikace. V reálných podmínkách by tato hodnota musela být upravena podle vytížení daného serveru. V pravidlech je možné zvolit, zda se počítají pakety podle cíle paketu či jeho zdroje, v tomto případě je vhodné vybrat podle cílové IP adresy, jelikož jsou pakety posílány z náhodnými zdrojovými adresami.

UDP Flood

Při změně typu útoku na UDP flood byla situace obdobná jako u útoku SYN flood. Pakety měly větší velikost o proti paketům při SYN flood útoku, došlo tak ke zvýšení přenosové rychlosti. Pakety měly velikost 780 byte při UDP flood útoku. UDP flood byl směřován na webový server na cílový port 1200 port byl zvolen náhodně bylo nutné ho zadat jako jeden z parametrů útoku. Výběrem cílového portu mohlo dojít k znesnadnění detekce IDS systémy.

Tab. 4.2 je definována stejně jako tab. 4.1 pouze zachycuje změřené hodnoty provozu při UDP flood útoku.

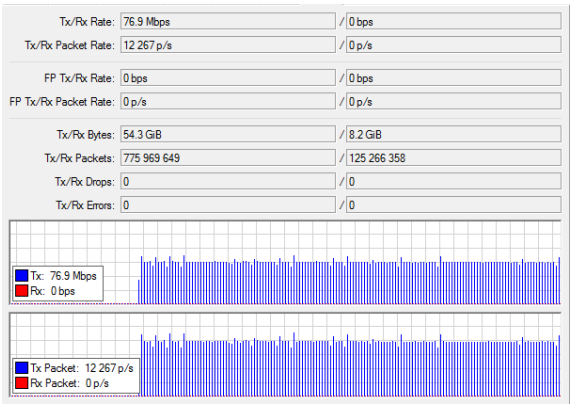
Tab. 4.2: Parametry útoků UDP flood

	Generováno Avalanchem			Po průchodu filtračním serverem	
počet uživatelů	Přenosová rych-lostMb/s	Počet paketů za sekundu	Filtrační server	Přenosová rychlost Mb/s	Počet paketů za sekundu
500	3	500	AS27	3	500
3500	17	3500	AS27	0,8	1500
4100	25,7	4100	AS27	25,7	4100
4100	25,7	4100	AS21	25,7	4100
14100	76,9	12267	AS27	19,3	3087
1410	69,2	11043	AS21	19,2	3000

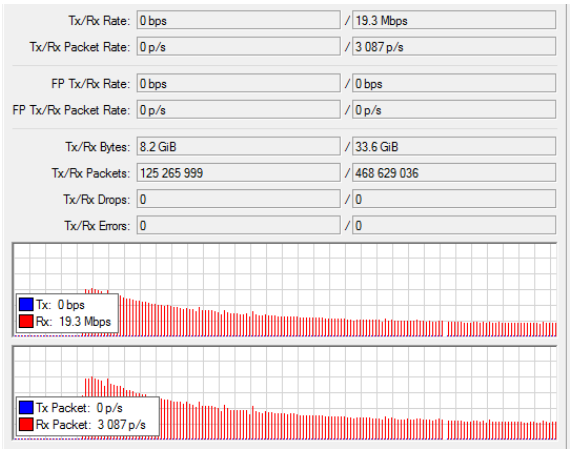
Probíhaly útoky při 500 p/s, tento provoz byl propuštěn až k cílovému webovému serveru bez zahazování na filtračním serveru. Zvýšením na 4100 p/s došlo ke plnému zatížení procesoru, avšak ne okamžitě po dobu 90 vteřin byl provoz propouštěn, až po této době byl procesor plně zatížen a provoz začal být omezován. Pokud docházelo k dalšímu navyšování počtu paketů docházelo i ke zkracování doby, po kterou byl provoz nezahazován. Tato situace nastala při průchodu datového provozu přes filtrační server **AS27**.

Při záměně průchodu UDP flood útoku pře filtrační server **AS21** bylo dosaženo při 4100 p/s propuštění celého datového provozu se zatížením procesoru kolem 20%. Stejně jako u předchozího filtračního serveru i zde při zvyšování počtu paketů docházelo k propouštění provozu pouze po určitou dobu a následně došlo k plnému vytížení procesoru. Při 11000 p/s byl provoz propouštěn bez omezení po dobu 90 vteřin a následně došlo opětovně k plnému vytížení procesoru a omezování datového provozu skrze filtrační server.

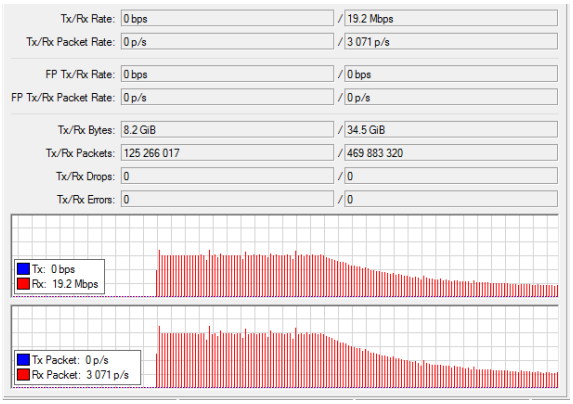
Na obr. 4.3a je vidět změna datové provozu před průchodem filtrovacím serverem. Jev který byl popsán výše a byl zjistěn u obou použitých filtrů jak **AS27** na Obr.4.3b tak u **AS21** na Obr. 4.3c oba jsou zachyceny při UDP flood útoku s silou útoku 14100 p/s.



(a) Provoz z testeru



(b) Filtrační server AS27



(c) Filtrační server AS21

Obr. 4.3: UDP flood 14100 SimUsers

Pokusy o eliminaci tohoto jevu byly všechny neúspěšné, dále docházelo k zahazování paketů i po následujících úpravách. Byla zvýšena maximální velikost front síťové karty pomocí programu ethtool. Tento program je schopen zobrazení a změny konfigurace síťové karty a parametrů ovladače této karty. Parametr ethX byl změněn za název rozhraní, které bylo mělo být upraveno. Tím dojde ke zvýšení velikosti vyrovnávací paměti pro příjem i odesílání.

```
ethtool -G ethX rx 4096 tx 4096
```

Byly měněny interní systémové hodnoty pro navýšení velikosti front pro síťovou komunikaci, avšak nedošlo ke zvýšení propustnosti. Zahazování paketů pokračovalo i přes tyto snahy o změny v systému.

```
sysctl -w net.core.rmem_max=8388608
sysctl -w net.core.wmem_max=8388608
sysctl -w net.core.rmem_default=65536
sysctl -w net.core.wmem_default=65536
sysctl -w net.ipv4.tcp_rmem='4096 87380 8388608'
sysctl -w net.ipv4.tcp_wmem='4096 65536 8388608'
sysctl -w net.ipv4.tcp_mem='8388608 8388608 8388608'
sysctl -w net.ipv4.route.flush=1
```

Byla zvětšena velikost pro vyrovnávací paměti pro příjem i pro vysílání využívané sokety, také byly zvětšeny i vyrovnávací paměti využívané operačním systémem pro všechny typy spojení. Ke stejné situaci docházelo u obou filtračních serverů, přestože byly rozdíly v hardwarové výbavě i obsahovaly jiné verze operačního systému.

Detekční server při UDP flood útoku nebyly schopni detekce, jakékoli velikosti popsané v tab. 4.2, ani při různých sílách jednotlivých útoku. Jednotlivé detekční servery při provádění daných útoků nebyly vůbec zatěžovány ani při nejvyšších silách UDP flood. Nejvyšší vytížení bylo dosaženo při útoku zobrazeném na obr. 4.3 kolem 10% pro IDS systém Snort, u programu Suricata na detekčním serveru detectB bylo celkové vytížení procesoru z důvodu vícevláknové aplikace špičkově pouze kolem 5%.

HTTP Flood

U typu útoku HTTP flood byly použity IP adresy definovány v profilu útoku pro možnost odpovědi na žádost HTTP GET generovanou testerem. Tento typ útoku byl specifikován transakcemi za sekundu.

Experimentálně bylo určeno jako hraniční hodnota, kterou je možné zatížit webový server, aniž by docházelo k plnému vytížení procesoru a tím by docházelo k nevyřízení požadavků, jako 4100 transakcí za sekundu. Pakety přenášející požadavek HTTP GET měli velikost 127 byte. V tab 4.3 jsou popsány, zjištěné hodnoty odpovídající různým silám útoku HTTP flood.

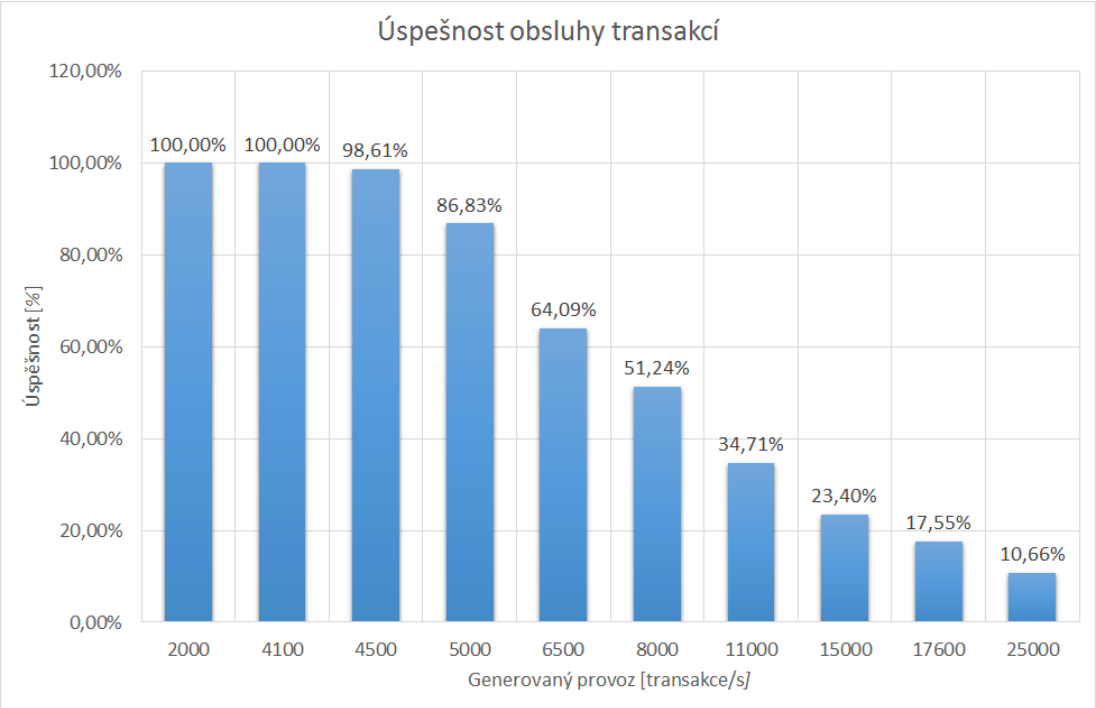
Tab. 4.3: Parametry útoků HTTP flood

	Generováno Avalanchem			Odpověď webserveru	
počet uživatelů	Přenosová rychlost Mb/s	Počet paketů za sekundu	Filtrační server	Přenosová rychlost Mb/s	Počet paketů za sekundu
2000	8,2	14000	AS27	185,8	20000
4100	16,9	28729	AS27	390,9	41038
4100	16,2	28739	AS21	381	41051
4500	18,7	33000	AS21	413,6	44552
5000	22,3	36600	AS21	396,7	43060
6500	23,8	33000	AS21	348,3	38295
8000	18,7	33000	AS21	413,6	44552
11000	38,9	64882	AS21	351,9	40740
15000	18,7	33000	AS21	413,6	44552
17600	58,3	90946	AS21	291,3	36496
25000	74,6	124267	AS27	259,5	35647
25000	76,5	119229	AS21	238,4	33386

Z tab.4.3 je zjištěno, že byly dosaženy stejné výsledky při použití filtračního serveru **AS21** i serveru **AS27**, výkonnostní rozdíl mezi oběma servery nebyl při HTTP flood útoku rozhodující. Nedocházelo k vytížení procesoru ani u jednoho ze serverů, při všech útocích libovolné síly ani nebylo pozorováno zahazování paketů, jako u předchozích typů DDoS útoků.

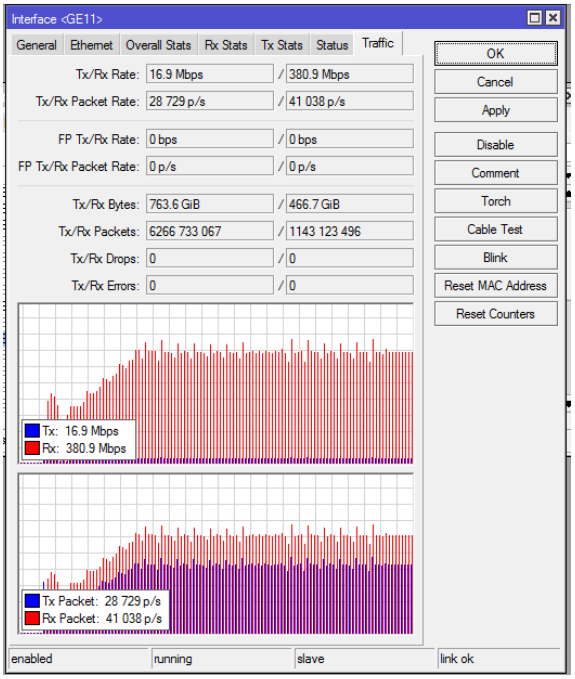
Útok o síle 25000 p/s byl nejsilnější, jaký bylo možné vygenerovat pomocí testeru Avalanche 3100B při používaném výstupním rozhraní.

Na obr. 4.4 je graf znázorňující kolik transakcí bylo uspěšně provedeno. Při zvýšení počtu transkací z 4100 na 4500 došlo o snížení vyřízených transakcí přibližně 1,5%. Dalším zvýšením tentokrát na 5000 transakcí za sekundu byla uspěšnost 86,8% tedy už pod 90

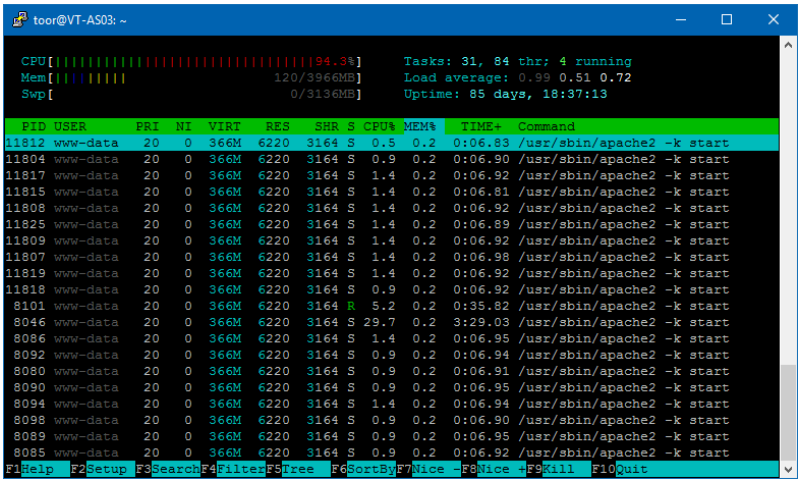


Obr. 4.4: Výkonnost webového serveru

Obr. 4.5a zobrazuje provoz přes rozhraní webového serveru. Zobrazují nejen datový provoz, který přicházel k webovému serveru, je zakreslen červenou barvou, ale i datový provoz tvořený odpověďmi na HTTP flood, ten je zaznamenán modře. Je v něm zapsán, jak počet paketů za sekundu, tak i přenosová rychlost v Mb/s. Druhý je Obr.4.5b a zachycuje vytižení procesoru webserveru v okamžiku generování provozu z Obr. 4.5a. Je vidět, jak při 4100 transakcích za sekundu nedochází k úplnému vytížení procesoru



(a) Síťový provoz



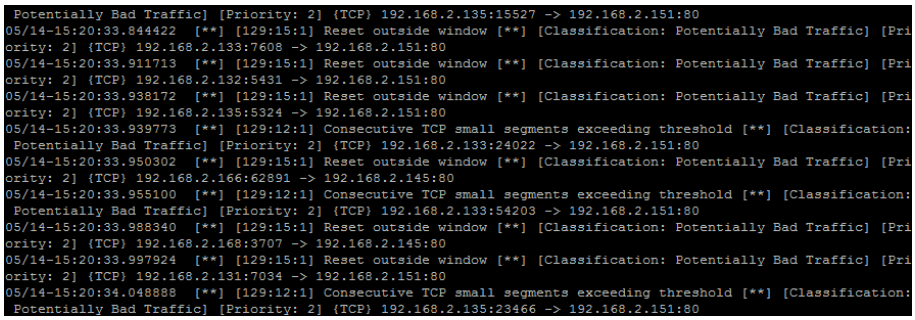
(b) CPU webserveru

Obr. 4.5: HTTP flood 4100 transakcí za sekundu

Detekční server zachytával veškerý provoz a byl nasledně kontrolován IDS systémem. Nástroj Suricata nebyl schopen detekoval žádný z útoků při použití pravidel bez jejich úpravy.

Po přidání pravidla použitého k detekci SYN flood útoku bylo možné detekovat i HTTP flood ve fázi vytváření TCP spojení při překročení 330 transakcí za sekundu, což přibližně odpovídá 1000 p/s, které jsou nutné k detekování provozu.

Na druhém detekční server s IDS systémem Snort detekoval úvodní čast HTTP flood útoku při vytváření TCP spojení mezi testerem a webovým serverem. Hlášení na Obr. 4.6 byla generována od hodnoty 1300 transakcí za sekundu, pokud datový provoz nedosahoval této hodnoty vůbec nedocházelo k jejímu generování.



Obr. 4.6: Detekce HTTP flood Snortem

Po přidání pravidla pro detekci SYN flood útoku docházelo k využívání tohoto pravidla a předchozí pravidlo z Obr. 4.6 přestalo být generováno. Pro aktivaci bylo nutné stejně jako v případě SYN flood útoku, aby datový provoz dosahoval minimáně hodnoty 1000 p/s.

Detekční servery zachytávající HTTP flood nebyly tímto útokem, téměř vytěžovány. U obou detekčních serverů, jak s programem Suricata, tak i programem Snort nedocházelo k žádnému vytížení procesorů, při nejvyšší sílé útoku docházelo pouze k 5% vytížení obou zmýněných systémů.

ICMP Flood

Při ICMP flood útoku byly zjištěny hodnoty uvedeny v tab. 4.4. První sloupec určuje Počet simulovaných uživatelů nastavených v profilu útoku. Druhý a třetí sloupec změřené hodnoty přenosové rychlosti a počtu paketů odpovídající útoku generovaného síťovým testerem Avalanche. Čtvrý sloupec označuje přes který filtrační server byl provoz směrován k webovému serveru. Velikost paketů použitých při ICMP flood útoku je 60 byte.

Tab. 4.4: Parametry útoků ICMP flood

Počet uživatelů	Přenosová rychlost Mb/s	Počet paketů za sekundu	Filtrační server
1000	0,51	992	AS27
1000	0,51	1001	AS21
25010	13	25270	AS27
25010	13,4	26345	AS21
164300	82,9	161000	AS27
164300	84,8	165780	AS21

U útoku ICMP flood nenastala situaci, která vznikala u UDP flood nebo SYN flood útoků. Nedocházelo k vytěžování procesoru při žádném scénáři ICMP flood útoku ani nevznikal problém se zahazováním provozu při vyšší síle útoku. Takto se chovali oba filtrační servery bez rozdílu, pro otestování bylo provedeno více útoků o více silách, avšak tento problém se nevyskytnul po celou dobu testování.

V počátečním nastavení nedocházelo při uvedených parametrech útoku k detekci ani programem Suricata na detekčním serveru **detect B** ani programem Snort nainstalovaným na serveru **detect A**. U programu Suricata byla původním nastavení pravidla pro ICMP komunikaci ignorována. Po povolení pravidel se podařilo detekovat uvedené ICMP flood útoků programem Suricata zachyceno na Obr. 4.7.

```
05/14/2017-16:13:03.986386  [**] [1:2100384:6] GPL ICMP_INFO PING [**] [Classification: Misc activity] [Priori
ty: 3] {ICMP} 192.168.2.131:8 -> 192.168.2.151:0
```

Obr. 4.7: Detekce Suricata

Pravidlo použito pro zachycení provozu ICMP flood útoku

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"GPL ICMP_INFO
PING"; icode:0; itype:8; classtype:misc-activity; sid:2100384; rev:6;)
```

Toto pravidlo je uloženo */etc/suricata/rules/emerging-icmp_info.rules*

Nástroj Snort u tohoto typu útoku nebyl schopen provést detekci útoku i při změnách síly útoku, důvodem byla zakázaná pravidla pro ICMP pakety. Povolením pravidel pro ICMP flood bylo možné dosáhnout detekce při všech experimentech s tímto typem útoku.

Nastavení pravidel bylo provedeno v souboru */etc/snort/rules/protocol-icmp.rules* na obr. 4.9. Obr. 4.8 ukazuje vygenerovanou zprávu IDS systémem Snort při detekci ICMP flood útoku.

```
05/14-16:13:03.603651  [**] [1:29456:2] PROTOCOL-ICMP Unusual PING detected [**] [Classification: Information
 Leak] [Priority: 2] {ICMP} 192.168.2.131 -> 192.168.2.151
```

Obr. 4.8: Detekce Snort

```
  GNU nano 2.2.6                    File: /etc/snort/rules/protocol-icmp.rules

# alert icmp 3.3.3.3/32 any -> $EXTERNAL_NET any (msg:"PROTOCOL-ICMP Stacheldraht server spoof"; icmp_id:666; itype:0; metadata:rulese$
# alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"PROTOCOL-ICMP tfn2k icmp possible communication"; icmp_id:0; itype:0; content:"A$
# alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"PROTOCOL-ICMP TFN Probe"; icmp_id:678; itype:8; content:"1234"; fast_pattern:onl$
# alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"PROTOCOL-ICMP Truncated ICMPv6 denial of service attempt"; content:"|60|"; depth$
# alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"PROTOCOL-ICMP Truncated ICMPv6 denial of service attempt"; content:"|60 58 58 58$
# alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"PROTOCOL-ICMP IPv6 0xfacebabe ICMP ping attempt"; itype:128; icode:0; icmp_id:64$
 alert icmp $HOME_NET any -> $EXTERNAL_NET any (msg:"PROTOCOL-ICMP Unusual Microsoft Windows 7 Ping detected"; icode:0; itype:8; dsize$
alert icmp $HOME_NET any -> $EXTERNAL_NET any (msg:"PROTOCOL-ICMP Unusual PING detected"; icode:0; itype:8; fragbits:!M; content:!"AB$
# alert icmp $HOME_NET any -> $EXTERNAL_NET any (msg:"PROTOCOL-ICMP Unusual Microsoft Windows Ping detected"; icode:0; itype:8; dsize:$
# alert icmp $HOME_NET any -> $EXTERNAL_NET any (msg:"PROTOCOL-ICMP Unusual L3retriever Ping detected"; icode:0; itype:8; dsize:>32; c$
# alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"PROTOCOL-ICMP FreeBSD rtsold dname_labeldec stack buffer overflow attempt"; ityp$
# alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"PROTOCOL-ICMP IPv6 multicast neighbor add attempt"; itype:135; icode:0; metadata$
# alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"PROTOCOL-ICMP Destination Unreachable Protocol Unreachable"; icode:2; itype:3; m$
# alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"PROTOCOL-ICMP destination unreachable port unreachable packet detected"; icode:3$
# alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"PROTOCOL-ICMP Microsoft Windows Ipv6pHandleRouterAdvertisement Route Information$
# alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"PROTOCOL-ICMP Microsoft Windows Ipv6pHandleRouterAdvertisement Prefix Informatio$
# alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"PROTOCOL-ICMP Squid Pinger IPv6 denial of service attempt"; icode:0; itype:>160;$
# alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"PROTOCOL-ICMP Squid Pinger IPv6 denial of service attempt"; icode:0; itype:11<>1$
```

Obr. 4.9: Konfigurace ICMP pravidel pro Snort

Detekční servery byly při ICMP flood útoku vytěžovány už od 25000 p/s, při útoku docházelo k úplnému vytížení přiřazeného jádra nejen programu Snort, ale také k vytížení jen jednoho vlákna u IDS systému Suricata. Jelikož je Snort jednovláknová aplikace je toto chování normální a očekávané v případě vysokého zatížení, ovšem Suricata používaná na druhém detekčním serveru je více vláknová a zatížení jednoho vlákna, tímto systémem nebylo zpozorováno při žádném z předchozích útoků provedených na experimentálním pracovišti.

XMasTree

Při generovaní XMasTree paketů testerem Avalanche byla zadána při výtváření útoku i statická IP adresa, která odpovídala danému adrese testeru. Daný DoS útok byl použit pro ověření zda Snort či Suricata zvládnou úspěšně detekovat i tento druh útoku. Během tohoto útoku byly posílány pakety o velikosti 60 byte. Tab 4.5 zaznamenává zjištěné hodnoty při průchodu XMasTree útoku přes filtrační servery.

Tab. 4.5: Parametry útoků XmasTree

Počet uživatelů	Přenosová rychlost Mb/s	Počet paketů za sekundu	Filtrační server
2000	1	2000	AS21
5000	2,5	5000	AS27
17000	8,7	17000	AS27
25000	12,8	25032	AS21
50000	25,6	50067	AS27
50000	25,6	50052	AS21

Filtrační servery vykazovaly při tomto útoku stejné chování jako v případě ICMP flood a HTTP flood útoků. Server **AS27** nezahazoval žádný síťový provoz, ani se nevyskytoval problém s vytěžováním procesoru daného serveru. Stejné chování bylo zjištěno u serveru **AS21** i ten propouštěl veškerý příchozí provoz a taktéž nevznikalo s tím spojené vytěžování procesoru.

Tento typ útoku je popisován jako lehce detekovatelný z důvodu výskytu nepoužívaných kombinací, ovšem ani jeden z použitých IDS systémů nebyl schopen provést úspěšnou detekci daného DoS útoku. Systém Suricata zachytávala datový provoz generovaný testerem a analyzoval ho nedokázal však detekovat tento útok. Po kontrole pravidel bylo zjištěno, že pravidla pro detekci tohoto útoku byla povolena a používána při detekování podezřelého provozu. Nejpravděpodobnější vysvětlení je odlišnost mezi detekčním pravidlem a pakety generované testerem. Program Snort i přes analýzu provozu, taktéž nebyl schopen detekovat tento útok při žádné síle. I přes rozsáhlou sadu detekčních pravidel nebyly nelezeny žádná implementovaná pravidla vhodná pro detekci XMasTree útoku.

Přestože detekční servery nedokázaly provést detekci útoku, musely provádět analýzu daného datového provozu, z toho důvodu docházelo k vysokému vytížení procesorů daných detekčních serverů. Detekční server s systémem Snort při zpracování datového provozu při 50000 p/s docházelo až k 35% vytížení CPU. IDS systém Suricata při stejném datovém provozu 50000 p/s využíval procesor kolem 35% stejně jako systém Snort na druhém serveru.

Následující kapitola je zaměřena na nasazení IPS systému a firewalu Iptablesl pro filtraci DDoS útoků.

5 VYUŽITÍ IPS A FIREWALL PRO FILTRACI DDOS

Z důvodu rozdílných verzí systémů nasazených na jednotlivých serverech bylo nutné zajistit, aby bylo možné provádět filtraci provozu přemostěním portů. O filtraci se stará nástroj iptables s pravidly identickými pro oba filtrační servery. U filtračního serveru **AS27** nebyl s tímto nasazením žádný problém a filtrování provozu fungovalo bez jakýchkoli potíží. Server **AS21** však používá novější verzi systému Debian a také novější verzi jádra Linux, kvůli tomu filtrace přes iptables nebyla ve výchozím stavu povolena a bylo nutné ji aktivovat. Příkaz přidal do jádra modul který umožňuje praci mezi bridgem a iptables . Druhý příkaz umožňuje programu iptables ovlivňovat provoz z bridge.

```
modprobe br_netfilter
echo "1» /proc/sys/net/bridge/bridge-nf-call-iptables
```

Pravidla použitá pro Iptables jsou uvedena v přilozeA.1. Pravidla povolují nová tcp spojení přes port 80 s příznakem SYN pro webové služby, TCP pakety na port 22 pro vzdálený přístup přes SSH a TCP pakety na cílový port 443 pro zabezpečenou webovou službu. Dále jsou povoleny ICMP zprávy a spojení která jsou již otevřená nebo patří k jinému otevřenému spojení.

Oba filtrační servery mají nainstalován IDS systém suricata. Tento systém byl nainstalován podle dokumentace na [26] pro IPS systém. Konfigurace je provedena podle dokumentace na [26] s využitím režimu IPS. Rozdílem oproti detekčnímu serveru je nástavení pro určení zacházení s pakety po průchodu IPS systémem Suricata, bylo nastaveno značení paketů, kdy v případě nedetekování podezřelého provozu jsou pakety IPS systémem označeny vráceny do chainu iptables pro ověření dalšími pravidly. Další rozdíl oproti IDS systému suricata je nastavení pravidel. Pravidla pro SYN flood a ICMP flood byla přepsána, tak aby nejen docházelo k generování zpráv, ale také k zahození paketu v případě že odpovídá danému pravidlu.

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 80 (msg:"rychly prichod SYN
paketu na port 80, potencionalni DOS"; flags: S,12; threshold: type both,
track by_dst, count 5000, seconds 5; classtype:misc-activity; sid:5;)
```

Jelikož detekční servery jsou instalovány před filtračními servery, tak nejsou ovlivněny filtračními metodami použitými na serverech **AS21** a **AS27**, proto nebudou v této části práce popisovány.

Pro ověření funkčnosti byly provedeny všechny útoky opakovaně ke zjištění změny při průchodu použitých druhý útoků na jednotlivých filtračních serverech.

UDP Flood

Pravidla nastavená v iptables byla nastavena k zahazování vetškeré komunikace přes UDP spojení proto docházelo k zatížení procesoru filtračního serveru v důsledku probíhajícího útoku. Přiložená tab. 5.1 zachycuje UDP flood útok při zahazování útoku firewallem Iptables na filtračním serveru **AS27**.

Tab. 5.1: UDP flood při filtraci iptables probíhájící na server AS27

Počet uživatelů	Přenosová rychlost Mb/s	Počet paketů za sekundu	Vytížení procesoru %	Doba před vytížením s
4100	25,7	4100	25	100
5000	31,3	5000	25	60
7000	43,9	7000	25	45
11000	69	11000	25	30

Procesor pracoval jen na 25% protože docházelo k vytížení pouze jednoho vlákna. V posledním sloupci je doba, po které začalo docházet k úplnému vytížení přiřazeného jádra.

Tab 5.2 zaznamenává údaje zjištěné při UDP flood útoku filtrovaném firewallem na filtračním serveru **AS21**.

Tab. 5.2: UDP flood při filtraci iptables probíhájící na server AS21

Počet uživatelů	Přenosová rychlost Mb/s	Počet paketů za sekundu	Vytížení procesoru %	Doba před vytížením s
4100	25,7	4100	4	
5000	31,3	5000	17	
7000	43,9	7000	100	160
11000	69,2	11043	100	90

Při měření útoku se sílou 5000 p/s bylo dosaženo vytížení procesoru kolem 17% na filtračním serveru **AS21**. Zvýšením generováného provozu síťového testeru došlo k úplnému vytížení procesoru. Tato situace se opakovala i při vypnutí IPS programu Suricata.

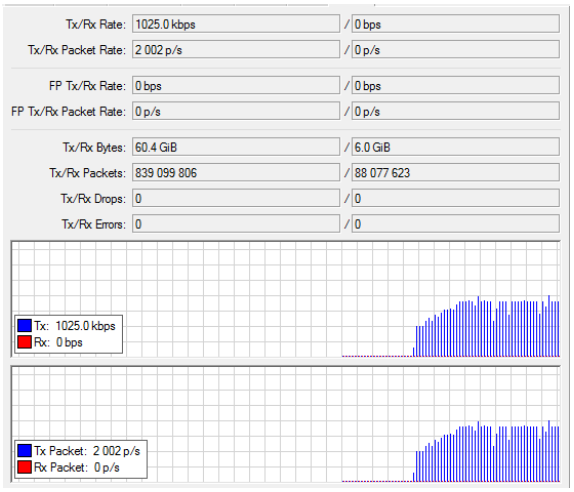
Porovnáním hodnot v tab. 5.1 a tab. 5.2 vyšly zajímavé hodnoty, zatímco čas potřebný k zahlcení serveru **AS21** při 11000 p/s byl 90 s, tak pro server **AS27** při 4100 p/s byla doba 100 s.

Rozdíl vzniklí popsaný výše byl doprovázen vytížením procesoru, který je rozdílný pro jednotlivé servery. Server **AS21** má 16 vláken, která byla všechny plně vytěžována. U **AS27**, který má pouze 4 vlákna a docházelo k vytížení pouze jednoho vlákna na 100%.

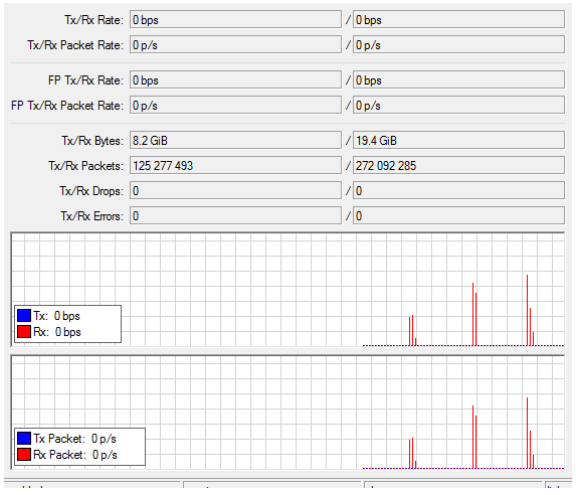
SYN flood

Pro manipulaci s SYN flood útokem generovaným testerem Avalanche bylo vloženo do systému Suricata nové pravidlo, které bylo použité už v detekčním serveru pro detekci SYN flood útoku, nyní poslouží pro jeho filtraci místo detekce.

Pokud byl datový provoz na hranici pravidla tzn. 1000 p/s procházel celý datový přenos bez omezení či nežádoucího vytěžování procesoru. Při zvýšení síly útoku na obr. 5.1a, však došlo k jeho úplnému zahození. Když systém Suricata začal kontroloval počet paketů odpovídající detekčnímu pravidlu, prošlo pouze menší množství paketů na obr. 5.1b.



(a) Příchozí SYN flood

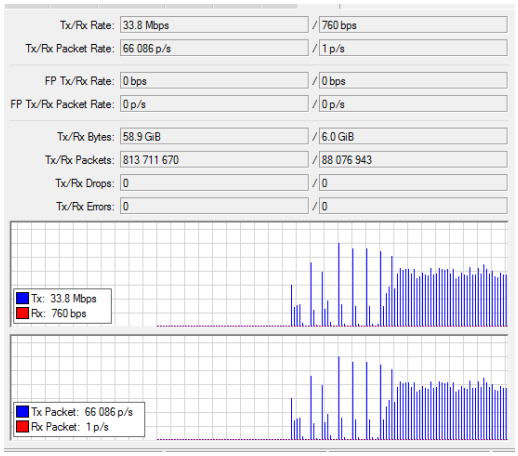


(b) Po průchodu filtračním serverem

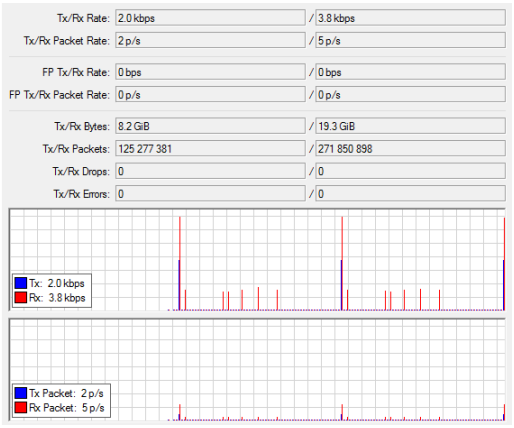
Obr. 5.1: SYN flood

HTTP flood

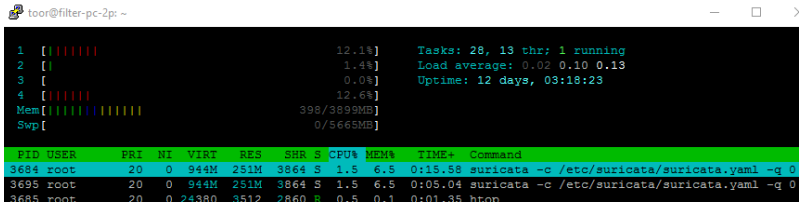
Použitím pravidla pro filtraci SYN flood útoku bylo možné pomocí IPS systému, také filtrovat provoz HTTP flood útoku. Největší zjištěné zatížení procesorů filtračních serverů bylo při síle útoku 25000 transakcí za sekundu. Na filtračním serveru **AS21** docházelo, při tomto zmíněném útoku, k vytížení okolo 10%, nenastávali žádné velké výkyvy v zátěži. U filtračního serveru **AS27** byl zjištěno zatížení procesoru okolo 5% na obr. 5.2c, však docházelo k nepravidelnému zvyšování zátěže nejspíš vlivem zpoždění v síti nebo ostatními aplikacemi na serveru. Na obr 5.2a je zachycen HTTP flood při 25 000 transakcí za sekundu přicházející do filtračního serveru a dále je na obr. 5.2b datový provoz, který z daného filtru směřuje k webovému serveru.



(a) Příchozí HTTP flood



(b) Po průchodu filtračním serverem



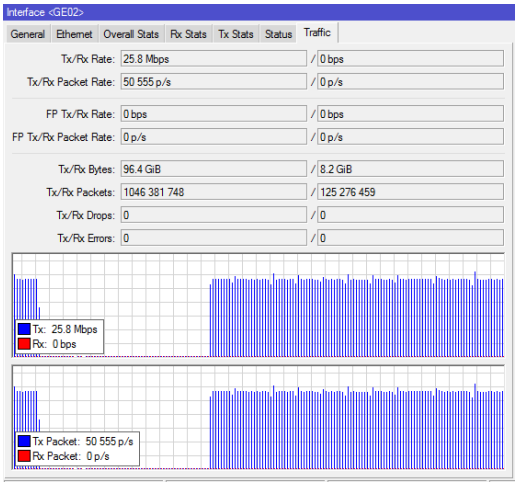
(c) Zatížení serveru během filtrace

Obr. 5.2: Filtrační server s IPS při HTTP flood

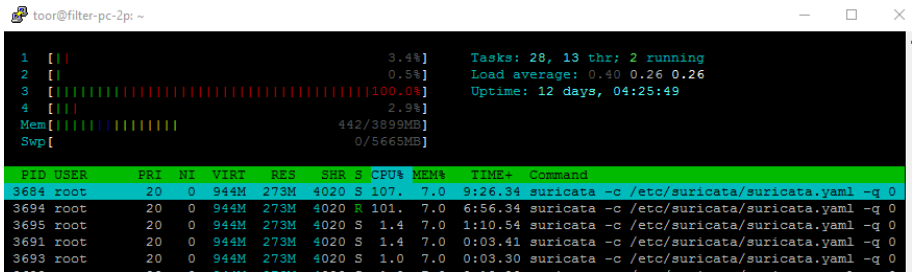
Vyřazením IPS systému a ponecháním pouze firewallu iptables byla testována výkonnost systému oproti systému bez použití iptables. U filtračního serveru **AS21** nebyla zjištěna žádná odchylka mezi nasažením iptables a jeho vypnutím. Nedocházelo k žádnému vytěžování procesoru při HTTP flood útoku ani zahazovaní provozu, byly zkoušeny i útoky o síle 25 000 transakcí za sekundu. Filtrační server **AS27** zpočátku nevykazoval žádné odchylky při využití iptables ani po jeho odstavení. Pouze u posledního útoku o síle 25 000 transakcí za sekundu začalo docházet k vytížení procesu a server začal zahazovat příchozí datový provoz.

ICMP flood

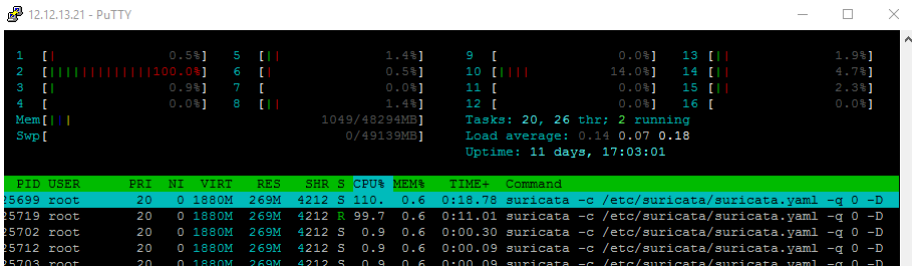
Použitím pravidla 4 pro filtraci ICMP echo request zpráv, bude docházet k odfiltrování probíhajícího ICMP flood útoku, ale také i legitimních dotazů, průchod těchto zpráv filtračním serverem nebude možný. Při filtraci dochází k vytížení procesoru už i při malém útoku. Na obr. 5.3a je příchozí ICMP flood o síle 2000 p/s. Při filtraci tohoto provozu filtračním serverem **AS21** dochází k zatížení jednoho vlákna na 100% viz obr. 5.3c. Pokud bude použit stejný příchozí icmp flood útok na filtrační server **AS27**, bude také plně zatížené jedno vlákno procesoru viz obr.5.3c.



(a) Příchozí ICMP flood



(b) Zatížení serveru AS21 během filtrace



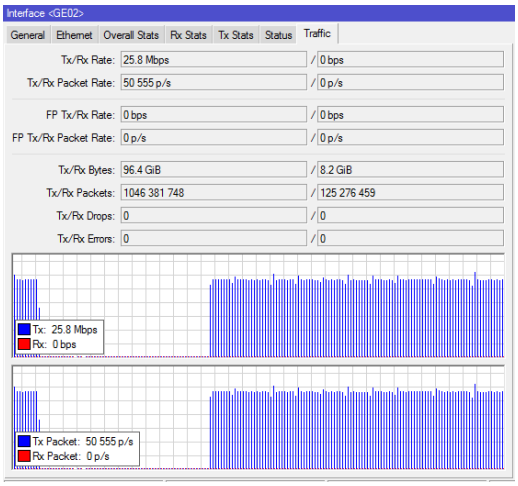
(c) Zatížení serveru AS21 během filtrace

Obr. 5.3: Filtrační servery při ICMP flood

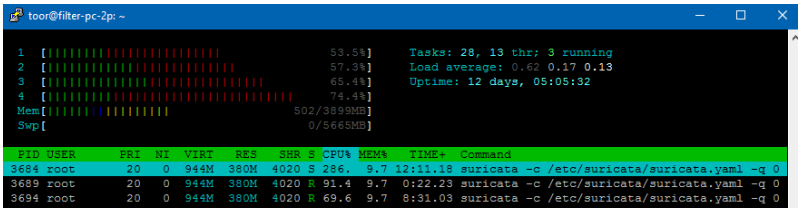
Při vyřazení IPS systému Suricata dochází k průchodu datového provozu firewallem, jelikož je v pravidlech iptables povolen průchod icmp zpráv. V tomto nastavení nedocházelo k žádnému vytížení procesoru ani omezení datového provozu. U tohoto typu útoku nezáleží zda prochází přes iptables.

XMasTree

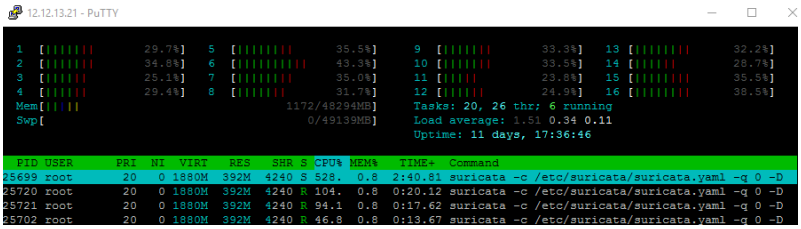
Tento typ útoku neodpovídá pravidlům IPS serverů a tak není tímto systémem odfiltrován a také pakety neodpovídají pravidlům v iptables, proto je veškerý provoz zahazován firewallem. Při filtraci generovaného XMasTree útoku zobrazeného na obr. 5.4a dochází k vysokému zatížení procesoru. Na obr. 5.4c je zobrazeno zatížení filtračního serveru **AS21**. Při stejném příchozím Xmas útoku dochází k velkému zatížení, také u filtračního serveru **AS27**, na Obr. 5.4b.



(a) Příchozí XMasTree attack



(b) Zatížení serveru AS27 během filtrace



(c) Zatížení serveru AS21 během filtrace

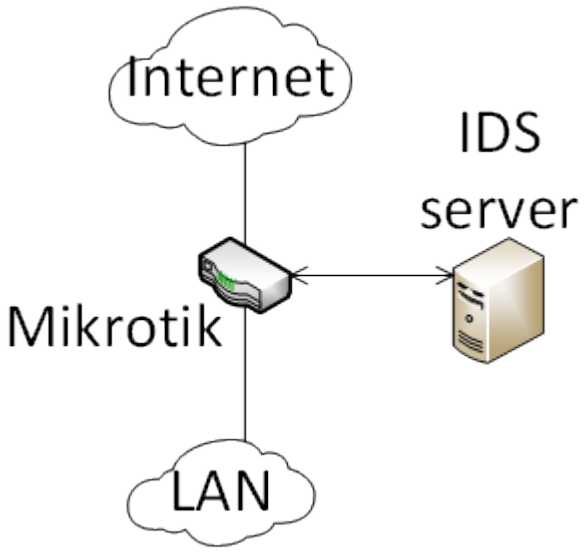
Obr. 5.4: Filtrační servery při XMasTree

Poslední kapitola se zabývá zmírněním dopadu útoků při využití varování z IDS systémů ve firewallu přepínačů Mikrotik.

6 VYUŽITÍ IDS SYSTÉMU PRO FILTRACI PROVOZU

Samotný IDS systém je schopen pouze detekce podezřelého provozu. Samostatně neumožňuje nijak zasáhnout do probíhající provozu ani v případě detekce podezřelé sítové aktivity. Při detekci podezřelého provozu dojde k upozornění, které obsahuje zdroj provozu, cíl kam jsou data posílány a název pravidla zodpovědného za vytvoření tohoto upozornění. Následným rozborem tohoto upozornění a dalším využitím ve firewallu lze provádět filtraci provozu a tím zmírnit dopad útoku.

Na Obr. 6.1 je jednoduché zapojení pro realizaci přenosu upozornění z IDS systému do směrovače.



Obr. 6.1: Topologie pro IDS a směrovač MikroTik

V příspěvku [14] byl využit IDS systém suricata, který byl použit k detekci provozu a následně byly generováné události separovány a použity pro blokování zasláním zprávy. Pro zasílání provozu k systému Suricata byl spuštěn paketový sniffer na přepínači Mikrotik.

Na samotném Mikrotiku byly vytvořeny ve firewallu pravidla pro zahazování provozu podle IP adres uložených v seznamu. Pro zápis IP adres do přepínače Mikrotik bylo využito rozhraní API integrováno v systému Mikrotik. Generované zprávy byly separovány na zdrojovou adresu a část z popisu zprávy uloženy do MySQL databáze, pro následné využití a zahazování daného provozu v přepínači Mikrotik. Byla vytvořena databáze s názvy sad pravidel, které mohou být použity pro zahazování provozu.

Následná databáze obsahuje události, které jsou zasílány do přepínače Mikrotik. Poslední databáze slouží pro propojení a funkčnost celého systému.

V posledním části je skript v PHP, který je využíván k opětovnému vytvoření seznamu blokovaných adres, pokud byl Mikrotik restartován. Druhá funkce slouží k hlídání nových záznamů v databázy a přidává nové záznamy do seznamu blokovaných IP adres.

Mírně odlišný přístup byl zvolen u realizace představené na [27]. Byl zde využít IDS program Snort, na nějž byl posílán datový provoz ze směrovače Mikrotik pomocí paketového snifferu. V daném příspěvku je popis celkového nastavení pro posílání provozu z Mikrotiku, tak i proces nutný ke korektnímu zachycení příchozího provozu programem Snort. Tento způsob popisuje IDS jako zařízení schopno přijímat a zpracovávat data i od více směrovačů.

Pro generování zpráv je využit Syslog, který je možné povolit v konfiguračním souboru Snortu. Na serveru s IDS systémem je prováděn PHP skript kontrolující Syslog. Jeho nastavení reaguje pouze při upozornění s prioritou 1 nebo v případě skenování portů. Skript provádí separeci IP adresy útočníka ze zachycené zprávy a poslá ji přes ssh do mikrotiku, kde dojde k blokaci zaslané IP adresy.

Ve směrovači Mikrotik je vykonáván skript, který po pěti minutách odstraní blokované IP adresy. Každou minutu je kontrolováno zde daná IP adresa má být odstraněna. Je vhodné pro umožnění zápisu pravidel do přepínače vytvořit nového uživatele s omezením na přístup pouze z IP adresy IDS systému.

7 ZÁVĚR

Při nasazení IDS systému Suricata bylo zjištěno, že sady pravidel použité pro detekci útoků, ve výchozím nastavení, nelze využít. Pravidla bylo nutné pro ICMP flood povolit v konfiguračním souboru. Pravidlo pro SYN flood bylo vytvořeno a vloženo k původním pravidlům. UDP flood neodpovídalo žádnému pravidlu pro jeho detekci, z toho důvodu bylo využito Iptables a provoz přes UDP zahazován na filtračním serveru. HTTP flood nebyl detekován a dokázal zahltit webový server, na který byl tento útok veden. Útok XMasTree se nepodařilo systému Suricata detekovat, i když měl pro tento typ útoku povoleny pravidla.

Testováním IDS programu Snort se podařilo detekovat více útoků než systému Suricata. Syn flood byl detekován bez nutnosti vložit nová pravidla. HTTP flood byl částečně detekován při každém vytváření TCP spojení mezi testerem Avalanche a webovým serverem pokud bylo nastaveno v profilu útoku minimálně 1300 transakcí za sekundu. ICMP flood se podařilo detekovat až po povolení pravidel, která byla ve výchozím nastavení zakázána. Snort nebyl schopen provést detekci dvou typů útoků XMasTree a UDP flood stejně jako systém Suricata.

Mezi použitými systémy bylo nepatrnější vytížení procesoru. Program Snort je jednovláknová aplikace a proto u větších útoků docházelo k vytížení jen jednoho jádra na 100%. Systém Suricata podporuje vícevláknové zpracování a díky této vlastnosti docházelo k plnému vytížení procesoru pouze v důsledku nejsilnějších útoků.

Při UDP flood a SYN flood útoku docházelo překročením 4000 p/s k omezení propustnosti a nežádoucí zahazování paketů na filtračním serveru, proto byly provedeny kroky ke snížení či úplnému odstranění této komplikace. Ovšem i po těchto změnách v systému, docházelo k nežádoucímu zahazování paketů datového provozu.

Po přidání pravidla pro detekci SYN flood útok došlo k zlepšení schopnosti detekovat útok SYN flood i útok HTTP flood.

Při testování filtračních schopností IPS systému Suricata společně i Iptables byla zjištěna možnost řízení propustnosti útoku HTTP flood a SYN flood. Ostatní útoky UDP flood a XMasTree útok byly kompletně odfiltrovány bez možnosti průchodu filtračním serverem. ICMP flood útok byl pravidly firewallu povolen, avšak IPS systém tetnto útok úplně odfiltroval.

Byly předloženy dvě možné metody přístupu ke zmírnění dopadu DDoS útoků při použití IDS systému Suricata a Snort ve spolupráci s firewallem využitým v přepínači Mikrotik.

LITERATURA

[1] ALBIN, Eugene a Neil C. ROWE. A Realistic Experimental Comparison of the Suricata and Snort Intrusion-Detection Systems. *2012 26th International Conference on Advanced Information Networking and Applications Workshops*. IEEE, 2012, , 122-127. DOI: 10.1109/WAINA.2012.29. ISBN 978-1-4673-0867-0. Dostupné také z: <http://ieeexplore.ieee.org/document/6185110/>

[2] AQIL, Azeem, Ahmed O. F. ATYA, Trent JAEGER, Srikanth V. KRISHNAMURTHY, Karl LEVITT, Patrick D. MCDANIEL, Jeff ROWE a Ananthram SWAMI. Detection of stealthy TCP-based DoS attacks. *MILCOM 2015 - 2015 IEEE Military Communications Conference*. IEEE, 2015, , 348-353. DOI: 10.1109/MILCOM.2015.7357467. ISBN 978-1-5090-0073-9. Dostupné také z: <http://ieeexplore.ieee.org/document/7357467/>

[3] ASHOOR, Asmaa Shaker a Sharad GORE. *Difference between Intrusion Detection System (IDS) and Intrusion Prevention System (IPS)*. , 497. DOI: 10.1007/978-3-642-22540-6_48. Dostupné také z: <http://link.springer.com/10.1007/978-3-642-22540-6_48>

[4] BOLEK, Daniel. *Zátěžové testování počítačových sítí*. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2014, 85 s. Bakalářská práce. Vedoucí práce Ing. Jan Hajný, Ph.D.

[5] FEINSTEIN, L., D. SCHNACKENBERG, R. BALUPARI a D. KINDRED. Statistical approaches to DDoS attack detection and response. *Proceedings DARPA Information Survivability Conference and Exposition*. IEEE Comput. Soc, 2003, , 303-314. DOI: 10.1109/DISCEX.2003.1194894. ISBN 0-7695-1897-4. Dostupné také z: <http://ieeexplore.ieee.org/document/1194894/>

[6] GARCÍA-TEODORO, P., J. DÍAZ-VERDEJO, G. MACIÁ-FERNÁNDEZ a E. VÁZQUEZ. Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers*. 2009, **28**(1-2), 18-28. DOI: 10.1016/j.cose.2008.08.003. ISSN 01674048. Dostupné také z: <http://linkinghub.elsevier.com/retrieve/pii/S0167404808000692>

[7] KRUEGEL, Christopher a Thomas TOTH. *Using Decision Trees to Improve Signature-Based Intrusion Detection*. , 173. DOI: 10.1007/978-3-540-45248-5_10. Dostupné také z: <http://link.springer.com/10.1007/978-3-540-45248-5_10>

[8] KRUEGEL, Christopher a Giovanni VIGNA. Anomaly detection of web-based attacks. *Proceedings of the 10th ACM conference on Computer and communication security - CCS '03*. New York, New York, USA: ACM Press, 2003, , 251-. DOI: 10.1145/948109.948144. ISBN 1581137389. Dostupné také z: <http://portal.acm.org/citation.cfm?doid=948109.948144>

[9] MIRKOVIC, Jelena a Peter REIHER. A taxonomy of DDoS attack and DDoS defense mechanisms. *ACM SIGCOMM Computer Communication Review*. 2004, **34**(2), 39-. DOI: 10.1145/997150.997156. ISSN 01464833. Dostupné také z: <http://portal.acm.org/citation.cfm?doid=997150.997156>

[10] NYCHIS, George, Vyas SEKAR, David G. ANDERSEN, Hyong KIM a Hui ZHANG. An empirical evaluation of entropy-based traffic anomaly detection. *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement conference - IMC '08*. New York, New York, USA: ACM Press, 2008, , 151-. DOI: 10.1145/1452520.1452539. ISBN 9781605583341. Dostupné také z: <http://portal.acm.org/citation.cfm?doid=1452520.1452539>

[11] PARK, Wonhyung a Seongjin AHN. Performance Comparison and Detection Analysis in Snort and Suricata Environment. *Wireless Personal Communications*. , -. DOI: 10.1007/s11277-016-3209-9. ISSN 0929-6212. Dostupné také z: <http://link.springer.com/10.1007/s11277-016-3209-9>

[12] PISKOZUB, A. *Denial of service and distributed denial of service attacks*. DOI: 10.1109/TCSET.2002.1015977. ISBN 10.1109/TCSET.2002.1015977. Dostupné také z: <http://ieeexplore.ieee.org/document/1015977/>

[13] SUNDARAM, Aurobindo. An introduction to intrusion detection. *Crossroads*. 1996, **2**(4), 3-7. DOI: 10.1016/j.cose.2008.08.003. ISSN 15284972. Dostupné také z: <http://portal.acm.org/citation.cfm?doid=332159.332161>

[14] TOMFISK. Suricata IDS/IPS integration with Mikrotik (now with OSSEC). In: *MikroTik RouterOS* [online]. 2016 [cit. 2017-05-16]. Dostupné z: <https://wiki.mikrotik.net/viewtopic.php?f=2&t=111727&sid=f2edac5dae911909f3984d5b812a3731>

[15] YATAGAI, Takeshi, Takamasa ISOHARA a Iwao SASASE. Detection of HTTP-GET flood Attack Based on Analysis of Page Access Behavior. *2007 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*. IEEE, 2007, , 232-235. DOI: 10.1109/PACRIM.2007.4313218. ISBN 1-4244-1190-4. Dostupné také z: <http://ieeexplore.ieee.org/document/4313218/>

[16] *Bridge-nf* [online]. [cit. 2017-05-12]. Dostupné z: <http://ebtables.netfilter.org/documentation/bridge-nf.html>

[17] *Bro network security monitor* [online]. [cit. 2017-05-04]. Dostupné z: <https://www.bro.org/>

[18] What is Multi Vector DDos attack and why are attackers preferring it ? *Coding Sec* [online]. 2016 [cit. 2017-05-02]. Dostupné z: <https://codingsec.net/2016/04/multi-vector-ddos-attack-attackers-shifting/>

[19] *OSSEC* [online]. [cit. 2017-05-04]. Dostupné z: <https://ossec.github.io/index.html>

[20] Global DDoS Threat Landscape Q2 2016. *Imperva Incapsula* [online]. 2016 [cit. 2017-05-02]. Dostupné z: <https://www.incapsula.com/ddos-report/ddos-report-q2-2016.html>

[21] Global DDoS Threat Landscape Q1 2016. *Imperva Incapsula* [online]. 2016 [cit. 2017-05-02]. Dostupné z: <https://www.incapsula.com/ddos-report/ddos-report-q1-2016.html>

[22] Global DDoS Threat Landscape Q4 2015. *Imperva Incapsula* [online]. 2015 [cit. 2017-05-02]. Dostupné z: <https://www.incapsula.com/ddos-report/ddos-report-q4-2015.html>

[23] THE INTERNET ENGINEERING TASK FORCE. *RFC 4987* [online]. 2007 [cit. 2017-05-06]. Dostupné z URL: <http://tools.ietf.org/html/rfc4987>.

[24] *Samhain* [online]. [cit. 2017-05-04]. Dostupné z:<http://www.la-samhna.de/samhain/index.htm>l

[25] *Snort* [online]. [cit. 2017-05-02]. Dostupné z: <https://www.snort.org/>

[26] *Suricata* [online]. [cit. 2017-05-02]. Dostupné z: <https://suricata-ids.org/>

[27] Mikrotik IPS IDS. *Mikrotik* [online]. 2014 [cit. 2017-05-16]. Dostupné z: <https://wiki.mikrotik.com/wiki/Mikrotik_IPS_IDS>

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

API	symbol Application Programing Interface
CPU	Central processing unit
DoS	Denied of Service
DDoS	Distributed Denied of Service
DNS	Domain Name System
GB	Giga byte
Gb/s	Giga bit za sekundu
GNU	GNU's Not Unix
GPL	General Public License
HIDS	Host Intrusion Detection System
HTTP	Hyper text transport protocol
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection system
IP	Internet Protocol
Mb/s	Mega bit za sekundu
Mp/s	milion paketů za sekundu
NIC	Network Interface Card
NIDS	Network Intrusion Detection System
PHP	Hypertext Preprocessor
p/s	pakety za sekundu
TCP	Transmission Control Protocoll
tzv.	takzvaný
UDP	User Datagram Protocol

SEZNAM PŘÍLOH

A PŘÍLOHA

A.1 Pravidla iptables

Seznam pravidel použitý pro filtraci provozu pomocí firewallu Iptables na filtračních serverech.

```
*filter
:INPUT ACCEPT [33:4623]
:FORWARD ACCEPT [33:4623]
:OUTPUT ACCEPT [675:47015]
-A INPUT -i eth0 -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT ! -i lo -d 127.0.0.0/8 -j REJECT
-A INPUT -m conntrack –ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -p tcp -m tcp –dport 80 -j ACCEPT
-A INPUT -p tcp -m tcp –dport 22 -j ACCEPT
-A INPUT -p tcp -m tcp –dport 443 -j ACCEPT
-A FORWARD -m conntrack –ctstate RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -m mark ! –mark 1/1 -j NFQUEUE
-A FORWARD -p icmp -j ACCEPT
-A FORWARD -p tcp -m tcp –dport 80 -j ACCEPT
-A FORWARD -p tcp -m tcp –dport 22 -j ACCEPT
-A FORWARD -p tcp -m tcp –dport 443 -j ACCEPT
-A INPUT -j REJECT
-A FORWARD -j REJECT
COMMIT
```

A.2 Obsah přiloženého CD

Přiložené CD obsahuje elektronickou verzi diplomové práce ve formátu PDF. Dále je přiložen seznam povolených pravidel pro IDS systémy Suricata a Snort.