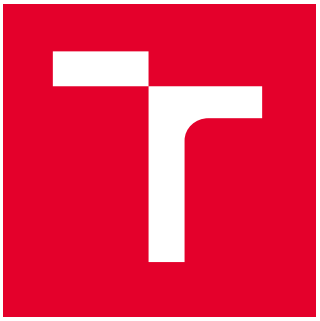


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

DDOS ÚTOKY ZAMĚŘENÉ NA DNS SERVERY

DDOS ATTACKS TARGETING DNS SERVERS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE — **Petr Zelinka**

AUTHOR

VEDOUCÍ PRÁCE — **Ing. Vlastimil Člupek, Ph.D.**

SUPERVISOR

BRNO 2025

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní program **Informační bezpečnost**

Ústav telekomunikací

Student: Petr Zelinka ***ID:*** 247637

Ročník: 3 ***Akademický rok:*** 2024/25

NÁZEV TÉMATU:

DDoS útoky zaměřené na DNS servery

POKYNY PRO VYPRACOVÁNÍ:

V bakalářské práci proveďte analýzu v praxi se vyskytujících útoků DDoS (Distributed Denial-of-Service) zaměřených na DNS (Domain Name System) a DNSSEC (Domain Name System Security Extensions) servery využívající protokoly UDP a TCP v IPv4 a IPv6 sítích. Na základě výsledků analýzy vyberte alespoň dva útoky a s využitím open-source knihoven je implementujte do aplikace Apache JMeter. V GUI útoku bude možné zvolit síťové rozhraní, ze kterého bude útok odcházet, cílovou MAC adresu, IPv4/IPv6 adresu a port, rozsah zdrojových MAC adres, zdrojových IPv4/IPv6 adres a zdrojových portů, sílu útoku a velikost (obsah) payloadu paketu. Otestujte funkčnost implementovaných útoků a proveďte výkonnostní testy, zaměřte se na velikost odesílaných dat a jakou rychlostí jsou data odesílána. Přehledně prezentujte dosažené výsledky.

DOPORUČENÁ LITERATURA:

[1] FANG, Lei, et al. A Comprehensive Analysis of DDoS attacks based on DNS. In: Journal of Physics: Conference Series. IOP Publishing, 2021. p. 012027.

[2] JMETER, Apache. Apache software foundation, https://jmeter.apache.org/.

Termín zadání: 10.2.2025 ***Termín odevzdání:*** 3.6.2025

Vedoucí práce: Ing. Vlastimil Člupek, Ph.D.
Konzultant: RNDr. Ing. Pavel Šeda, Ph.D.

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

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Bakalářská práce se zabývá problematikou DoS/DDoS útoků na DNS servery, konkrétně implementování a otestování těchto útoků. Hlavní zaměření je na záplavové útoky. Bylo vytvořeno několik konfiguračních souborů pro nástroj Trafgen, který generuje síťový provoz na DNS server. Konfigurační soubory jsou specificky vytvořené, aby dosáhly odepření služby. Následně byly útoky implementovány do programu Apache JMeter. Během útoku probíhá monitoring testu.

KLÍČOVÁ SLOVA

DoS, DDoS, odepření služby, DNS, IPv4, IPv6, trafgen, java, apache jmeter, jmeter, ack flood, fin flood, rst flood, dns query flood, Bind 9, mpstat, wireshark, testování, analýza, záplavové útoky

ABSTRACT

The Bachelor Thesis deals with the problem of DoS/DDoS attacks on DNS servers, specifically the implementation and testing of these attacks. The main focus is on flooding attacks. Several configuration files have been created for the Trafgen tool that generates network traffic to the DNS server. The configuration files are tailored to achieve denial of service. Subsequently, the attacks were implemented in Apache JMeter. The test is being monitored during the attack.

KEYWORDS

DoS, DDoS, denial of service, DNS, IPv4, IPv6, trafgen, java, apache jmeter, jmeter, ack flood, fin flood, rst flood, dns query flood, Bind 9, mpstat, wireshark, testing, analysis, flood attacks

Vysázeno pomocí balíčku `thesis` verze 4.11; `https://latex.fekt.vut.cz/`

ZELINKA, Petr. *DDoS útoky zaměřené na DNS servery*. Bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2025. Vedoucí práce: Ing. Vlastimil Člupek, Ph.D.

Prohlášení autora o původnosti díla

Jméno a příjmení autora:	Petr Zelinka
VUT ID autora:	247637
Typ práce:	Bakalářská práce
Akademický rok:	2024/25
Téma závěrečné práce:	DDoS útoky zaměřené na DNS servery

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

Brno ....

podpis autora*

*Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Vlastimilovi Člupkovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Obsah

Seznam obrázků

Seznam tabulek

Seznam výpisů

Úvod

DoS a DDoS útoky představují významnou hrozbu pro fungování komunikace na internetu. DNS, jakožto jedna ze základních komponent internetu, zajišťuje převod doménových jmen na IP adresy, a tím umožňuje komunikaci mezi zařízeními. Proto je častým terčem útočníků, jejichž cílem je přetížení serveru a znemožnění poskytování služeb.

Tato práce se zaměřuje na analýzu a implementaci vybraných DDoS útoků využívajících protokoly UDP a TCP v IPv4 a IPv6 sítích. Hlavní pozornost je věnována záplavovým útokům.

V teoretické části je představen teoretický základ problematiky, zahrnující popis relevantních protokolů a klasifikaci DDoS útoků. Na konci teoretické části jsou uvedeny jaké nástroje a programy jsou použity v praktické části.

Praktická část se věnuje samotným výsledkům práce. Zahrnuje nastavení prostředí a tvorbu konfiguračních souborů pro generátor síťového provozu Trafgen. Dále se věnuje implementaci vytvořených útoku do programu Apache JMeter a následné otestování účinnosti útoků.

Cíle práce

Cíl bakalářské práce je provést analýzu v praxi se vyskytujících útoků DDoS zaměřených na DNS servery využívající protokol UDP a TCP v IPv4 a IPv6 sítích. Na základě výsledků analýzy vybrat alespoň dva útoky a s využitím open-source knihoven je implementovat do programu Apache JMeter a následně otestovat. Při testování se zaměřit na velikost odesílaných dat a jakou rychlostí jsou data odesílána.

1 Teoretická část

V teoretické části bude popsán a vysvětlen základní model používaný při komunikaci mezi počítačovými sítěmi. Následně bude specifikováno několik protokolů, které jsou klíčové pro řešení praktické části. V dalších kapitolách se vymezí DDoS útok a konkrétně záplavové útoky. Nakonec je zmíněno jaké nástroje jsou v praktické části použity.

1.1 Referenční model TCP/IP

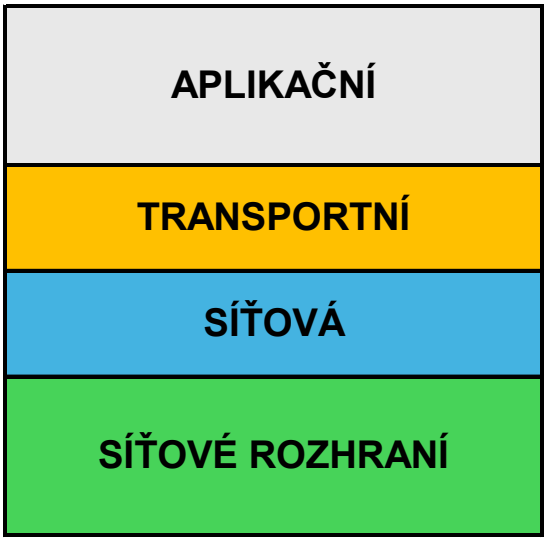
TCP/IP (Transmission Control Protocol/Internet Protocol) je název spojený ze dvou komunikačních protokolů, na kterých je postavená komunikace přes Internet. Ve skutečnosti to je soustava několika protokolů a předloha, jak by se počítačové sítě měly budovat a fungovat.

Tvůrce TCP/IP modelu byla organizace ARPAnet, která měla za účel budovat velké počítačové sítě a k tomu potřebovala nějaký systém, jak má taková komunikace probíhat, a proto v roce 1978 začal vznikat tento model. Po intezivním testování na počátku roku 1983 ARPAnet přešla celkově na tento nový model, za důsledek to mělo exponenciální růst sítě. I přesto, že v roce 1990 společnost ARPAnet zaniknula, se model stále udržuje, každopádně postupně se modifikuje, aby se držel trendu a požadavků se neustále měnícího Internetu. [1]

Jak už bylo nastíněno, hlavní úkol TCP/IP modelu je přenést data z jednoho zařízení na druhé. Hlavní podmínkou tohoto procesu je, aby data byla spolehlivá a přesná, aby příjemce obdržel stejné informace, které byly odeslány ze zdroje. Aby bylo zajištěno, že každá zpráva dorazí do svého konečného cíle přesně, rozděluje model TCP/IP posílaná data do paketů a na druhém konci je spojuje, což pomáhá zachovat integritu dat při přenosu z jednoho konce na druhý. TCP/IP model rozděluje data čtyřvrstvou procedurou, kde na straně odesílatele se tento proces nazývá zapouzdřování a na straně příjemce rozbalování (v angličtině se v této souvislosti používají termíny *encapsulation* a *de-encapsulation*). Na obr. 1.1 je znázorněno, jak je model TCP/IP rozdělen do vrstev. [1]

Aplikační vrstva

Aplikační vrstva v TCP/IP modelu je nejvyšší vrstva, která poskytuje rozhraní mezi uživatelskými aplikacemi a síťovými službami. Zajišťuje komunikaci a přenos dat mezi aplikacemi na různých zařízeních prostřednictvím sítě. Tato vrstva umožňuje přístup uživatelských aplikací k síťovým službám jako jsou například webové prohlížeče nebo e-mailové klienty. Převádí data do formátu, který může být odeslán



Obr. 1.1: Vrstvy referenčního modelu TCP/IP.

přes síť, a naopak. Klíčové protokoly, které se v této vrstvě nachází jsou např. HTTP (Hypertext Transfer Protocol), FTP (File Transfer Protocol), DNS (Domain Name System) a mnoho dalších. [1]

Transportní vrstva

Druhá vrstva v TCP/IP modelu je transportní, která zajišťuje spolehlivý přenos dat mezi aplikacemi na různých zařízeních v síti. Hlavní funkcí je poskytovat mechanismy pro přenos dat, jejich správu a zajištění integrity. Rozděluje data z vyšších vrstev na menší části [1] pro přenos sítí a zajišťuje jejich opětovné sestavení na straně příjemce. Zajišťuje vytvoření, udržování a ukončení spojení mezi dvěma zařízeními v případě použití protokolu TCP, ale může se použít i nespojitý přenos v podobě UDP. Umožňuje přenos dat mezi více aplikacemi současně, a to pomocí portů, které identifikují konkrétní služby. [1] Identifikace portů probíhá následovně:

- 0–1023 jsou dobře známé porty (např. 80 pro HTTP, 443 pro HTTPS, 53 pro DNS),
- 1024–49151 jsou registrované porty pro specifické aplikace,
- 49152–65535 jsou dynamické nebo privátní porty, přidělované dočasně.

Síťová vrstva

Třetí vrstva v TCP/IP modelu je síťová, také nazývaná Internetová vrstva, je zodpovědná za směrování dat mezi zařízeními v různých sítích. Zajišťuje logickou ad-

[1] Pokud je použitý protokol TCP, tak se jednotka nazývá segment, pokud je použitý UDP (User Datagram Protocol), tak se nazývá datagram.

resaci, přenos datových paketů napříč sítěmi a výběr nejefektivnější trasy. Síťová vrstva přiděluje každému zařízení v síti logickou adresu (IP adresu), která umožňuje jednoznačnou identifikaci zařízení. Na základě IP adres rozlišuje zdrojovou a cílovou adresu datového paketu. Směrováním se určuje optimální trasa, kterou se má paket dostat od zdroje k cíli, a to i přes různé sítě. Používá směrovací tabulky a protokoly k zajištění efektivního doručení. Přijímá data z transportní vrstvy a zapouzdřuje je do paketů, které obsahují hlavičku s adresami a dalšími informacemi pro přenos. Rozděluje velké pakety na menší části nazývané fragmenty, pokud přenosová cesta nepodporuje jejich původní velikost. Na straně příjemce zajišťuje opětovné složení fragmentů. Dva nejdůležitější protokoly, které se na této vrstvě nacházejí jsou IPv4 (Internet Protocol version 4) a IPv6 (Internet Protocol version 6). Další klíčové vlastnosti jsou, že je nezávislá na přenosovém médiu a IP protokol nezaručuje spolehlivost, pořadí ani integritu dat, to je zajištěno vyšší vrstvou. [1]

Vrstva síťového rozhraní

Vrstva síťového rozhraní v TCP/IP modelu je nejnižší vrstva, která zajišťuje fyzický přenos dat mezi zařízeními. Překládá data z logické podoby (pakety) do fyzické (bity) a zpět. V rámci soustavy TCP/IP není tato vrstva blíže specifikována, neboť je zcela závislá na použité přenosové technologii. Mezi nejznámější technologie patří standard pro kabelové sítě Ethernet IEEE 802.3, bezdrátový standard Wi-Fi IEEE 802.11 nebo bezdrátový standard Bluetooth IEEE 802.15. [1]

1.1.1 Protokol IPv4

IPv4 je jeden z nejdůležitějších protokolů používaných v síťové komunikaci. Poskytuje mechanismus pro adresování zařízení a směrování dat. IPv4 používá adresy o délce 32 bitů, což umožňuje přibližně 4,3 miliardy unikátních adres (2^{32}). Adresa se zapisuje v desítkovém tvaru ve formátu čtyř čísel oddělených tečkami (např. `192.168.1.1`).

Aktuální počet obyvatel na Zemi je 8,2 miliardy, navíc většina lidí používá více než jedno zařízení se síťovou kartou, tím pádem vzniká velký nedostatek volně dostupných IPv4 adres. Existují metody, které tento problém minimalizují jako např. NAT [2] a podsíťování, ale z dlouhodobého hlediska nebude možné tento problém řešit pouze prostřednictvím těchto metod. Z tohoto důvodu už probíhá několik let snaha o přechod na IPv6.

[2]NAT překládá soukromé IP adresy používané v lokálních sítích na veřejné IP adresy a naopak. Tím umožňuje, aby mnoho zařízení v lokální síti sdílelo jednu veřejnou IP adresu. Tento mechanismus efektivně šetří veřejné IPv4 adresy tím, že snižuje jejich potřebu, protože zařízení využívající soukromé IP adresy nezasahují přímo do veřejného adresního prostoru. [6]

Hlavička IPv4

Délka IPv4 hlavičky má délku minimálně 20 bajtů a maximálně 60 pokud se použijí volitelné položky záhlaví. Detailní struktura IPv4 hlavičky je znázorněna na obr. 1.2.

- verze IP – 4 bity, identifikuje verzi protokolu (4 pro IPv4),
- délka záhlaví – 4 bity, hodnota se musí uvádět, protože záhlaví může mít kvůli volitelným položkám proměnnou délku,
- typ služby – 8 bitů, položka by měla sloužit ke specifikaci požadované kvality přenosu IP datagramu,
- celková délka IP datagramu – 16 bitů, definuje délku datagramu včetně záhlaví a uživatelských dat,
- identifikace IP datagramu – 16 bitů, určeno k identifikaci k sobě patřících fragmentů pro sestavování,
- příznaky – 3 bity, používají se dva: DF (don't fragment) říká nefragmentovat a MF (more fragments) říká, že datagram byl fragmentován a že bude následovat další část,
- posunutí fragmentu od počátku – 13 bitů, udává pozici obsahu dat datagramu vzhledem k původního paketu,
- doba života (TTL) – 8 bitů, tato hodnota definuje maximální počet skoků směrovačema daného paketu než má být zahozen,
- protokol vyšší vrstvy – 8 bitů, obsahuje číselnou identifikaci protokolu vyšší vrstvy (TCP nebo UDP),
- kontrolní součet záhlaví datagramu – 16 bitů, v každém uzlu se přepočítává záhlaví datagramu a paket se zahodí pokud součet nesedí,
- IP adresa odesílatele/příjemce paketu – 32 bitů, jedná se o logickou adresu v rámci IP protokolu,
- volitelné položky záhlaví – nepovinné, až do délky 40 bajtů, nevyužívá se příliš často,
- přenášená data – délka až 65536 bajtů (v součtu se záhlavím), jsou to údaje, které IP vrstvě předal protokol vyšší vrstvy. [2]

1.1.2 Protokol IPv6

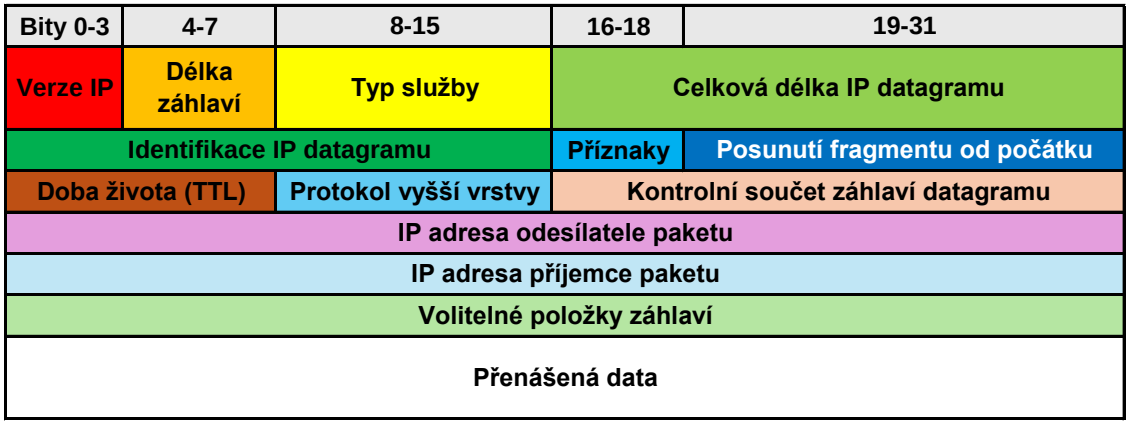
IPv6 (Internet Protocol version 6) je nejnovější verze internetového protokolu, která nahrazuje IPv4. Byl navržen, aby vyřešil omezení IPv4, zejména nedostatek adres, a přinesl další vylepšení v oblasti efektivity, bezpečnosti a škálovatelnosti. IPv6 používá adresy dlouhé 128 bitů, což umožňuje až 2^{128} adres. Adresa se zapisuje jako osm hexadecimálních číslic oddělených dvojtečkami (např. `2001:0db8:85a3:0000:0000:8a2e:0370:7334`).

Při vytváření Internetu byl záměr poskytovat přímočarou komunikaci mezi dvěmi koncovými stanicemi, to je v současnosti s nasazením NATu znesnadněno. Z tohoto důvodu byla vyvinuta IPv6. Nasazování IPv6 ve světě probíhá už několik let, ale celkový přechod je stále v nedohledu.

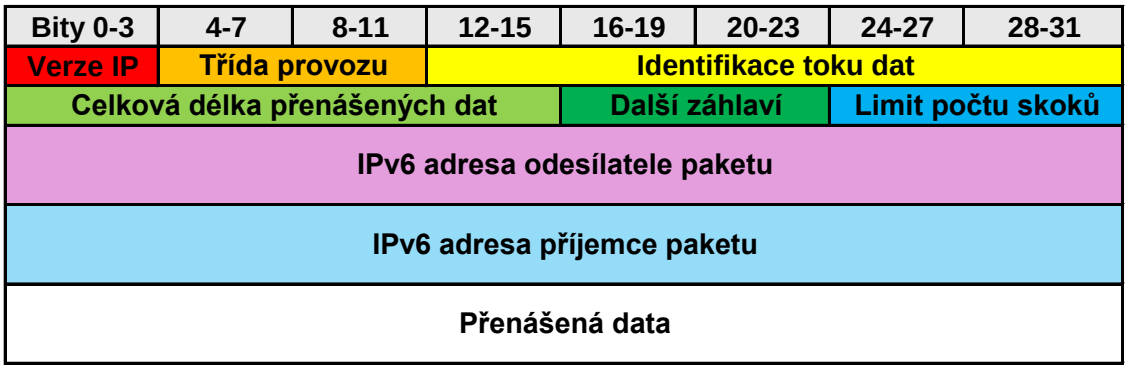
Hlavička IPv6

Délka IPv6 hlavičky má pevnou délku 40 bajtů, což zrychluje zpracování paketů. Celkově byla hlavička zjednodušena, několik polí, které IPv4 hlavička obsahuje se v IPv6 nenachází. Detailní struktura IPv6 hlavičky je znázorněna na obr. 1.3.

- verze IP – 4 bity, identifikuje verzi protokolu (6 pro IPv6),
- třída provozu – 8 bitů, toto pole umožňuje nastavit prioritu paketu,
- identifikace toku dat – 20 bitů, označení toku dat pro zjednodušení směrování,
- celková délka přenášených dat – 16 bitů, délka přenášených dat,
- další záhlaví – 8 bitů, typicky informace o protokolu vyšší vrstvy (TCP nebo UDP),
- limit počtu skoků (hop limit) – 8 bitů, ekvivalent položky TTL u IPv4,
- IPv6 adresa odesílatele/příjemce paketu – 128 bitů, jedná se o logickou adresu v rámci IP protokolu. [3]



Obr. 1.2: IPv4 hlavička.



Obr. 1.3: IPv6 hlavička.

1.1.3 Protokol UDP

Protokol UDP je první ze základních komunikačních protokolů. UDP je nespolehlivý a nespojovaný, což znamená, že neprovádí žádnou kontrolu doručení ani pořadí dat. Tato jednoduchost dělá UDP rychlým a nenáročným na zpracování, tudíž vhodný pro aplikace, kde je klíčová nízká latence a není potřeba spolehlivé doručení. Vlastnosti, principy a funkce jsou následující:

- nespolehlivost – nezaručuje, že data dorazí do cíle, a neposkytuje žádný mechanismus pro opakování ztracených paketů,
- minimální režie – hlavička UDP je jednoduchá a zabírá málo místa,
- rychlost – díky absenci potvrzení a dalších mechanizmů přenos probíhá rychle,
- integrita dat – provádí se kontrolní součet, ale oprava chyb neprobíhá.

Přenosová jednotka UDP se nazývá datagram. Hlavička je jednoduchá, protože UDP protokol jako takový neposkytuje mnoho funkcí. Funguje na principu, v angličtině označováno jako „best effort“, zjednodušeně řečeno tzn. vyexpeduj data za každou okolnost a neřeš jestli a v jakém stavu dorazí do cíle. Detailní hlavička je vyobrazena na obr. 1.4 a funkce jednotlivých polí jsou následující:

- zdrojový/cílový port – 16 bitů, identifikuje aplikaci resp. port na straně odesílatele a příjemce,
- celková délka – 16 bitů, určuje délku celého datagramu,
- kontrolní součet – 16 bitů, základní detekce chyb. [5]

1.1.4 Protokol TCP

Protokol TCP je druhý ze základních komunikačních protokolů. TCP zajišťuje spolehlivý přenos dat mezi aplikacemi běžícími na různých zařízeních v síti. Je pomalejší než UDP, a proto se využívá v případech, kdy je více potřeba spolehlivost, než rychlost. Vlastnosti, principy a funkce jsou následující:

- spojově orientovaná služba – pro začátek přenosu musí být navázáno spojení a pro konec přenosu musí být spojení ukončeno viz obr. 1.5 [3],
- spolehlivá služba – používá potvrzovací mechanizmy pomocí zpráv ACK,
- zajištění správného pořadí – je zajištěno, že pakety dorazí ve stejném pořadí v jakém byly odeslány,
- ovládání toku dat – umožňuje regulaci rychlosti posílaných dat, aby příjemce nebyl přetížen,
- detekce chyb – využívá kontrolního součtu pro detekování poškozených paketů.

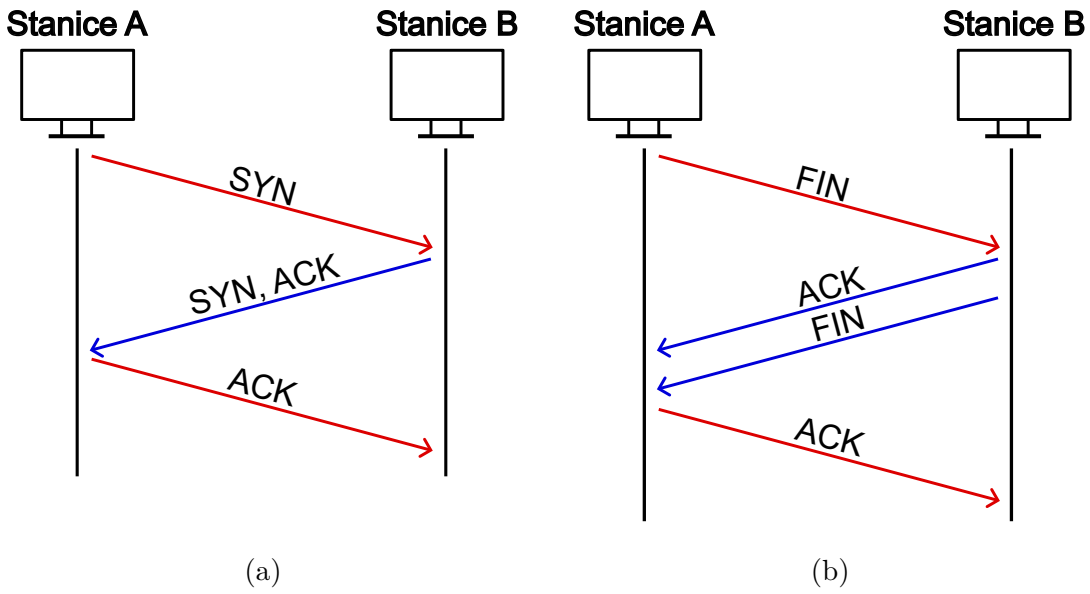
Přenosová jednotka TCP se nazývá segment. Hlavička je rozsáhlejší oproti UDP hlavičce, kvůli více funkcím, které TCP poskytuje. Detailní hlavička je vyobrazena na obr. 1.6 a funkce jednotlivých polí jsou následující:

- zdrojový/cílový port – 16 bitů, identifikuje aplikaci resp. port na straně odesílatele a příjemce,
- pořadové číslo odesílaného/potvrzovacího bajtu – 32 bitů, SEQ (sequence number) a ACK (acknowledgment number) označuje pořadí odesílaných dat a indikuje, která data byla přijata,
- délka záhlaví – 4 bity, číslo udávající délku celého záhlaví,
- rezerva – 6 bitů, nepoužívá se,
- příznakové bity – 6 bitů, pro řízení spojení:
  - URG – určuje naléhavá data,
  - ACK – potvrzuje platná data,
  - PSH – data ihned předat aplikaci,
  - RST – odmítnout spojení,
  - SYN – navázat spojení,
  - FIN – ukončit spojení.
- délka okna – 16 bitů, určuje velikost dat, která můžou být poslána, než je vyžadováno potvrzení,
- kontrolní součet – 16 bitů, mechanizmus k zajištění integrity dat,
- ukazatel naléhavých dat – 16 bitů, vyplněno pouze když je použit URG příznakový bit,
- volitelné položky záhlaví – pole nemusí být přítomné. [4]

[3] Navázání spojení se nazývá „třícestné podání rukou“ a ukončení spojení „čtyřcestné podání rukou“, anglicky three-way handshake a four-way handshake.

Bity 0-15	16-31
Zdrojový port	Cílový port
Celková délka	Kontrolní součet
Data aplikace	

Obr. 1.4: UDP hlavička.



(a) (b)

Obr. 1.5: Komunikace TCP protokolem a) navázání, b) ukončení.

Bity 0-15								16-31
Zdrojový port								Cílový port
Pořadové číslo odesílaného bajtu								
Pořadové číslo potvrzovaného bajtu								
Délka záhlaví	Rezerva	URG	ACK	PSH	RST	SYN	FIN	Délka okna
Kontrolní součet								Ukazatel naléhavých dat
Volitelné položky záhlaví								
Data aplikace								

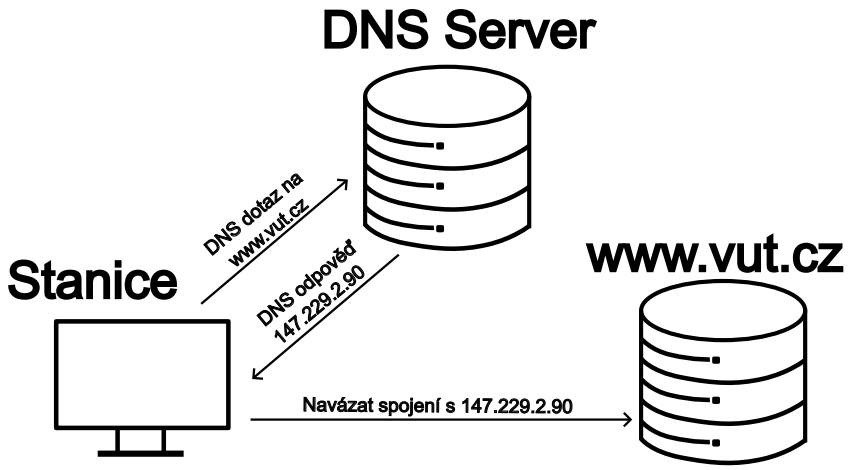
Obr. 1.6: TCP hlavička.

1.1.5 Služba DNS

DNS je aplikační protokol, který překládá názvy domén na IP adresy, které jsou nezbytné pro komunikaci mezi zařízeními v síti. DNS se dá přirovnat k telefonnímu seznamu, má uložený ve své paměti záznamy, které IP adresy patří k jaké doméně a naopak. Umožňuje uživatelům se připojit k webovým stránkám a službám pomocí snadno zapamatovatelných názvů místo číselných adres, které jsou pro člověka těžké na zapamatování.

Systém je založený na principu klient - server a navíc je decentralizovaný, takže DNS serverů je po světě několik. Ukázka zjednodušené komunikace klienta se serverem je znázorněna na obr. 1.7. DNS domény jsou strukturovány hierarchicky:

- root doména – nejvyšší úroveň, označuje se tečkou,
- doména nejvyšší úrovně – TLD (Top Level Domain), např. `.cz` nebo `.com`,
- doména druhého řádu – např. `vut` nebo `seznam`,
- doména třetího řádu – subdomény, např. `moodle`.vut.cz. [11]



Obr. 1.7: Zjednodušená komunikace s DNS serverem.

Systém DNS odlišuje několik typů záznamů, které jsou uloženy v záznamech RR (Resource Records). Nejčastěji používané záznamy jsou vypsány v tab. 1.1.

Zabezpečení DNS

DNSSEC (Domain Name System Security Extensions) je rozšíření, které zvyšuje bezpečnost služby doménových jmen. DNSSEC zvyšuje bezpečnost při používání DNS tím, že zabraňuje podvržení falešných, pozměněných či neúplných údajů o doménových jménech. Typickým případem může být příchod uživatele na falešnou webovou stránku, která se tváří jako jeho bankovní systém o po zadání přihlašovací údajů mu jsou odcizeny. Díky zavedení DNSSEC získá uživatel jistotu, že informace,

které z DNS získal, byly poskytnuty správným zdrojem, jsou úplné, důvěrné a jejich integrita nebyla při přenosu narušena.

DNSSEC zavádí do DNS asymetrickou kryptografii. Držitel domény vygeneruje dvojici soukromého a veřejného klíče. Soukromým klíčem elektronicky podepíše údaje, které o své doméně do DNS vkládá. Pomocí veřejného klíče je pak možné ověřit pravost tohoto podpisu. Aby byl tento klíč dostupný všem, publikuje jej držitel domény u nadřazené autority, kterou je pro všechny domény `.cz` registr domén `.cz`. I na úrovni registru domén `.cz` jsou data v DNS podepsána a veřejný klíč k tomuto podpisu je opět správcem registru předán nadřazené autoritě. Vytváří se tak řetězec, který zajistí důvěryhodnost údajů, pokud není v žádném svém článku porušen, a všechny elektronické podpisy souhlasí. [12]

Tab. 1.1: Typy záznamů DNS serveru.

Typ záznamu	Účel záznamu
A	Překlad na IPv4 adresu
AAAA	Překlad na IPv6 adresu
MX	Informace o poštovním serveru
NS	Jméno autoritativního DNS serveru
CNAME	Alias ke stejné doméně

1.2 Útoky DoS a DDoS

Denial of service (DoS) a Distributed Denial of service (DDoS), česky odepření služby a distribuované odepření služby, je typ kybernetického útoku, který má za cíl znefunkčnit a znepřístupnit legitimní službu ostatním uživatelům. Může se to týkat webových stránek, převodu peněz z bankovního účtu nebo i poslouchání rádiové stanice. Podstata útoku spočívá v zahlcení služby neúměrným množstvím zpráv resp. paketů. Tím dochází k přetížení infrastruktury nebo aplikačních zdrojů, což může vést k různým problémům, jako jsou:

- úplné selhání služby,
- nucené restartování systému,
- pozastavení provozu,
- omezení výkonu. [7]

1.2.1 Rozdíl mezi DoS a DDoS

DoS

DoS útok je prováděn z jednoho zařízení, přičemž útočník se snaží narušit funkčnost služby opakovaným odesíláním velkého množství požadavků. Tento typ útoku je relativně snadněji identifikovatelný a blokovatelný, protože zdroj útoku pochází z jednoho konkrétního zařízení. [7]

DDoS

DDoS útok je sofistikovanější a mnohem nebezpečnější podtyp DoS útoku. V tomto případě je útok prováděn ze široké sítě zařízení, často označované jako botnet [4]. Tato zařízení mohou být infikována škodlivým softwarem a ovládána útočníkem bez vědomí jejich majitelů. Díky této vlastnoti je velmi obtížné takový útok odrazit, protože požadavky přicházejí z mnoha různých zařízení. [7]

1.2.2 Ochrana proti DDoS útokům

Velmi podstatnou částí ochrany proti DDoS útokům je monitoring. Pokud existují monitorovací nástroje, které jsou správně nastaveny, detekce nestandardního chování na základě anomálií je většinou velmi rychlá. Detekují se anomálie, což jsou neobvyklé vzorce, jako jsou náhlé nárůsty požadavků. Druhý způsob je detekce signatur, to jsou vzory, součástí je dobrá znalost samotného DDoS útoku. Signatury

[4]Botnet je rozsáhlá síť kompromitovaných zařízení (tzv. zombies) ovládaných kybernetickými útočníky (bot herder). [8]

jsou sestavovány experty a jsou implementovány do bezpečnostních a dohledových síťových prvků.

Druhým krokem je mitigace pro zmírnění nebo úplného zabránění útoku. Je hodně způsobů, jak může probíhat mitigace:

- použití firewallu – nastavení pravidel, která blokují podezřelý provoz,
- rate limiting – omezuje počet požadavků, které mohou přijít z jedné IP adresy během určitého časového období, existuje riziko blokování legitimního provozu,
- geo-blokace – blokování provozu z určitých zemí nebo regionů,
- redundantní linky – přesun legitimních uživatelů na záložní linku,
- blacklist a whitelist – umístit legitimní uživatele na whitelist a podezřelé na blacklist,
- tarpit akce – snížení TCP délky okna na 0 a tím neumožnit přenos dat až po timeout. [13]

1.2.3 Klasifikace útoků DDoS na DNS

DNS DDoS útoky se můžou rozdělit do různých typů podle cesty útoku, cíle, způsobu útoku atd. Podle způsobu útoku se dělí na DNS query flood, DNS reply flood, DNS water torture útok a hybridní útok. [9] Dalším typem útoku, který nespadá přímo pod kategorii DNS DDoS útoků se nazývá „state-exhaustion attack“, jedná se o typ útoku zaměřený na vyčerpání zdrojů služby nebo samotného serveru. [10]

DNS query flood

Tento útok spočívá v posílání velkého množství DNS dotazů na server pro dosáhnutí odepření služby. Běžná metoda útoku může být buď s podvrženou IP adresou, nebo se použije botnet s legitimními IP adresami. Útočník posílá specificky vyrobené žádosti o překlad s podvrženým doménovým jménem, tím pádem DNS server nebude mít tento záznam uložený v mezipaměti a bude muset provádět rekurzivní hledání. S tím, jak objemné tyto útoky jsou se následně server přetíží, zahltí se mezipaměť nebo obojí. [9]

DNS reply flood

Komunikace na DNS serveru převážně používá ke komunikaci UDP protokol, který je nespojovaný. Tento útok spočívá v posílání DNS reply zpráv na DNS server. DNS server následně musí zpracovat všechny zprávy a tím server přetíží.

DNS reflection flood také spadá pod tento způsob útoku. Spočívá v tom, že útočník zfalšuje zdrojovou IP adresu za adresu oběti a následně využije otevřenosti DNS serverů. DNS servery mají veřejně známé IP adresy, při využití této nevýhody

se můžou posílat žádosti na DNS servery a odpovědi budou přicházet oběti. Při tomto útoku se využívá amplifikace. [9]

DNS water torture útok

Princip tohoto útoku spočívá v přidání PRSD (Pseudorandom Subdomain) k doménovému jménu, aby se obešla DNS mezipaměť a bylo to zasíláno relevantnímu autoritnímu serveru. Jelikož taková subdoména neexistuje, tak je nejprve dotaz zaslán na root server, obdrží se odpověď `NXDOMAIN`, neexistující doména, Dále je dotaz zaslán na TLD, taky se obdrží odpověď `NXDOMAIN`. Autoritní server musí zkontrolovat vždy jestli takový záznam je přítomen v jeho registrech, tím se přetíží DNS server. [9]

Objevuje se zde také PTR (Pointer) útok, který spočívá v zahlcení DNS požadavky na zpětné vyhledávání. Jsou zasílány dotazy ve tvaru `192.168.1.1.in-addr.arpa`, účel je též přetížení DNS serveru neexistujícími adresami.

Hybridní útok

Kombinuje předchozí způsoby mezi sebou.

State-exhaustion útok

State-exhaustion útok je typ DDoS útoku, jehož cílem je přetížit síťová zařízení, která udržují stavové informace o síťových spojeních. Mnoho síťových zařízení sledují stav každého síťového spojení – např. TCP spojení mezi klientem a serverem. Tyto stavy jsou uloženy ve speciální paměti (tzv. *connection tracking table* nebo *conntrack table*), která má ale omezenou kapacitu. Jakmile je paměť pro tyto stavy plná, tak nová spojení nemohou být přijata, legitimní uživatelé nemohou přistupovat ke službám, systém se zpomalí nebo selže.

Typy útoku, které spadají pod tuto kategorii záplavových útoku a jsou implementovány v praktické části jsou FIN, RST a ACK flood. Zneužívají vlastnosti protokolu TCP, protože server musí vyhodnotit příchozí pakety. V případě FIN záplavového útoku útočník zasílá pakety s příznakem FIN, tento příznak vyvolává ukončení spojení, každopádně žádné předchozí spojení z útočníkovy strany nebylo navázáno. Server pro vyhodnocení paketu musel alokovat potřebné prostředky, které dále nemůžou být použity na legitimní provoz. Na stejný způsob fungují i RST a ACK flood, které server musí zpracovat, i když nejsou součástí žádné předchozí komunikace.

1.3 Nástroje a balíčky použité pro testování a generování provozu

Existuje mnoho nástrojů nebo programů, se kterými se dají implementovat útoky v praktické části. Pro generování síťového provozu bude použit open-source nástroj `trafgen`. Jako DNS systém v testovacím prostředí `Bind 9`. Zátěž na serveru se monitoruje pomocí nástroje `mpstat` a sledování provozu na straně útočníka s `wireshark`.

1.3.1 Trafgen

Trafgen [14] je volně dostupný nástroj pro generování síťového provozu, který umožňuje odesílání přesně definovaných paketů s vysokou rychlostí. Je součástí balíčku netsniff-ng, což je sada nástrojů pro analýzu, manipulaci a generování síťového provozu.

Trafgen umožňuje rozšířenou konfiguraci díky tomu že používá svůj vlastní konfigurační jazyk pro vytváření paketů založený na makrech viz tab. 1.2, vytvořených v jazyce C. Díky této konfiguraci lze sestavit libovolný paket. Ve výchozím stavu spouští trafgen tolik procesů, kolik je k dispozici procesorů. Připojí každý proces k příslušnému procesoru a nastaví cyklickou vyrovnávací paměť poté, co zkompiluje seznam paketů k přenosu.

Vytvořený paket je možné odeslat pomocí příkazu:

```
$ sudo trafgen --cpp --dev enp0s3 --conf soubor.cfg
```

Argument `--cpp` značí, aby byl paket předám preprocesoru C, `--dev` označuje odchozí síťové zařízení a `--conf` vstupní konfigurační soubor s definovaným paketem.

Tab. 1.2: Příklady maker nástroje trafgen.

Definice	Význam
drnd(n)	Vygeneruj náhodné číslo o velikosti n bajtů
csumip(n, m)	Vypočítej kontrolní součet pro IP hlavičku od bajtu n po m
fill(m, n)	Naplň hodnotami m n-krát

1.3.2 Mpstat

Mpstat [17] je užitečný nástroj pro monitorování výkonu CPU na víceprocesorových systémech. Je součástí balíčku `sysstat`, který zahrnuje různé nástroje pro sledování systémových prostředků a výkonu v reálném čase nebo na základě historických dat. Příklad použití může být následovné:

```
$ mpstat -P ALL 1 > cpulog.txt
```

Argument `-P ALL 1` specifikuje, že se mají zobrazit statistiky pro všechna jádra CPU každou sekundu. Výstup se přesměruje do souboru `cpulog.txt` obsahující sloupce se zatížením. Všechny sloupce a jejich významy jsou uvedeny v tab. 1.3.

Tab. 1.3: Výstupní data mpstat.

Sloupec	Význam
%usr	Procento času stráveného vykonáváním uživatelských procesů
%nice	Procento času pro procesy s upravenou prioritou
%sys	Procento času vykonáváním systémových procesů
%iowait	Procento času CPU čekajícího na dokončení I/O operací
%irq	Procento času obsluhou hardwarových přerušení
%soft	Procento času obsluhou softwarových přerušení
%steal	Procento času CPU „ukradeného“ jinými virtuálními stroji
%idle	Procento času, kdy CPU neprovádí žádné operace

Sloupec, který má v testování DoS a DDoS útoků největší význam je `%idle`, kde při 100 % znamená, že procesor je v klidu a při při 0 % je zatížen maximálně.

1.3.3 Wireshark

Wireshark [16] je volně dostupný nástroj pro analýzu síťového provozu. Umožňuje zachytávat, zobrazovat a analyzovat data, která putují sítí v reálném čase nebo byla dříve zaznamenána. Ze zachycené komunikace nabízí různé statistické přehledy, jako je využití protokolů nebo IO grafy. Umožňuje filtrování zachycených dat pomocí zachytávacích filtrů a displejových filtrů. Příklady filtrů, které jsou primárně využívány v řešení práce se nachází v tab. 1.4.

Tab. 1.4: Příklad filtrů ve wireshark.

Filtr	Význam
dns	Filtruje podle paketů využívající protokol dns
ip.src	Filtruje podle zdrojové IP adresy
ip.dst	Filtruje podle cílové IP adresy

1.3.4 Bind 9

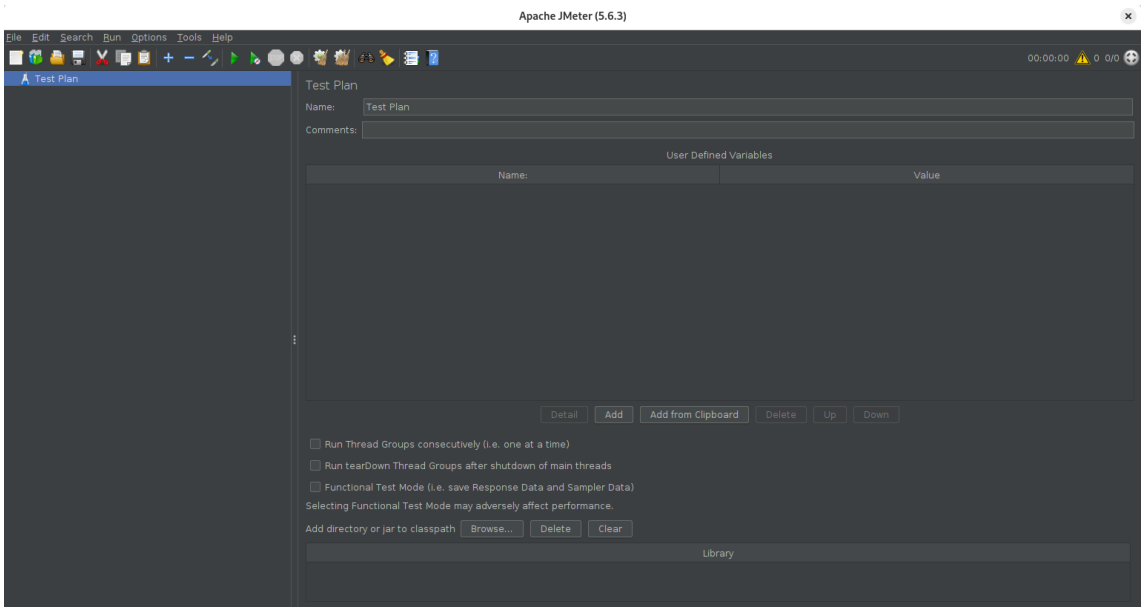
BIND 9 [18] je jedním z nejrozšířenějších a nejběžněji používaných DNS serverů. BIND je volně dostupný software, který poskytuje funkce pro správu doménových

jmen, jejich překlad na IP adresy a další související služby.

1.3.5 Apache JMeter

Apache JMeter [19] je open-source nástroj pro testování výkonnosti a měření zátěže aplikací, primárně zaměřený na webové aplikace, ale podporuje i další protokoly jako `FTP`, databáze (`JDBC`), `LDAP`, e-mail (`SMTP/POP3/IMAP`), `SOAP/REST API`, `MQTT` a další. Je vyvíjen Apache Software Foundation a je široce používán pro simulaci náročných scénářů, identifikaci úzkých míst a optimalizaci systémů. Je navržen tak, aby umožňoval simulaci realistických scénářů zátěže, například pomocí distribuovaného testování, kdy se testy spouštějí na více počítačích pro dosažení vyšší škálovatelnosti. Uživatelé mohou vytvářet komplexní testovací plány pomocí komponent jako Thread Group, která definuje počet uživatelů a jejich chování, nebo Samplers, které specifikují jednotlivé požadavky na systém, například `HTTP GET` nebo `POST`. Na úvodní obrazovce na obr. 1.8 v horní liště se nachází možnosti pro start testování pomocí zelené šipky, uložení testu disketou a další. V levé části se nachází testy a uprostřed GUI (Graphical User Interface) právě vybraného testu.

JMeter je možné používat v prostředích, kde se dá nainstalovat Java. Umožňuje uživatelům možnost rozšiřovat funkce pomocí vlastně vytvořených testů prostřednictvím tzv. pluginů. V rámci této práce jsou vytvořeny 4 pluginy, které jsou následně i otestovány.



Obr. 1.8: Úvodní obrazovka programu Apache JMeter.

2 Výsledky studentské práce

V praktické části bude nejprve popsána příprava prostředí pro realizaci záplavových útoků. Jsou uvedeny verze použitých nástrojů a operačních systémů.

V další kapitole se nachází vytvoření konfiguračních souborů pro nástroj `trafgen`. V následující kapitole bude představeno rozšíření programu Apache JMeter, konkrétně již vytvořeného modulu Ddos [20], o útoky vytvořených díky konfiguračním souborům. V poslední kapitole jsou prezentovány výsledky samotných útoků, včetně otestování a srovnání účinnosti.

2.1 Instalace a nastavení prostředí

Všechny scénáře jsou realizované ve virtuálních strojích v izolované síti. Hostovský počítač je vybazen procesorem Ryzen 5 2600, který disponuje 6 fyzickými jádry a 12 vlákny. Použité verze programů a OS jsou:

- Na straně útočníka:
  - trafgen – 0.6.8+
  - Wireshark – 4.4.2
  - Apache JMeter – 5.6.3
  - OS – Debian 12.8.0, 64bit, verze kernelu: 6.1.0-26, 4 virtuální procesory, 4 GB RAM
- Na straně oběti (DNS server):
  - Bind 9 – 9.18.28-0
  - Tshark – 4.2.2
  - OS – Ubuntu Server 24.04.1 LTS, 64bit, verze kernelu: 6.8.0-48, 2 virtuální procesory, 2 GB RAM

Kompilace trafgenu

Nejprve je potřeba doinstalovat nutné závislosti:

```
$ sudo apt-get install ccache libnet1-dev libnl-3-dev
 libnl-genl-3-dev libnl-route-3-dev libgeoip-dev bison
 libnetfilter-conntrack-dev libncurses5-dev liburcu-dev
 libnacl-dev libpcap-dev zlib1g-dev libcli-dev flex
```

Po doinstalování závislostí je potřeba stáhnout repozitář balíčku `netsniff-ng` a zkompilovat trafgen:

```
$ git clone https://github.com/borkmann/netsniff-ng
$ cd netsniff-ng/
```

```
$ ./configure
$ make
$ make trafgen
```

Instalace wireshark

Wireshark se nainstaluje příkazem:

```
$ sudo apt-get install wireshark
```

Instalace prostředí Apache JMeter

Pro používání JMeteru je potřeba mít nainstalovanou Javu minimálně verze 8. Na oficiálních webových stránkách https://jmeter.apache.org-download_jmeter.cgi se může stáhnout nejnovější verze binárních souborů. Po rozbalení a extrahování souborů je připraveno vše pro spuštění.

Konfigurace DNS serveru

Bind 9 se nainstaluje příkazem:

```
$ sudo apt-get install bind9 bind9utils bind9-doc
```

Dále se soubor `/etc/bind/named.conf.options` upraví tak, aby vypadal, jak na výpisu 2.1.

Výpis 2.1: Konfigurace Bind 9 serveru.

```
options {                                                    1
directory "/var/cache/bind";                                 2
listen-on { any; };                                          3
listen-on-v6 { any; };                                       4
allow-query { any; };                                        5
allow-query-on { any; };                                     6
};                                                           7
```

2.2 Vývoj modulu – Apache JMeter

Byly vytvořeno celkem 4 rozšíření modulu DNS záplavových útoků a to jak pro verzi s IPv4 tak i IPv6. Každé rozšíření zahrnuje GUI, kde má uživatel umožněno nastavit parametry související s vybraným útokem.

Jednotlivé útoky mají vytvořené konfigurační soubory nástroje trafgen, který slouží jako generátor provozu a jsou umístěné v adresáři s DDoS modulem. Pro

zachování jednoty jsou jednotlivé soubory pojmenované `AbstractAckFlood.cfg`, `AbstractDnsQueryFlood.cfg`, `AbstractFinFlood.cfg` a `AbstractRstFlood.cfg`. Stejně jsou pojmenované i konfigurační soubory pro verzi s IPv6 variantou, akorát navíc s příponou „Ipv6“.

2.2.1 Struktura konfiguračních souborů pro nástroj trafgen

Konfigurační soubory jsou specificky vytvořené, aby byly kompatibilní s DDoS modulem používaným v Apache JMeter. Soubory jsou před sestavením vloženy do projektu do adresáře `jmeter-ddos-plugin/src/main/resources/trafgen_cfg`. Parametry, které definují konfigurační soubory jsou nastavené jako proměnné viz výpis 2.2, tím pádem umožňují konkrétnější konfiguraci uživatelem skrz GUI.

- `ETH_P_IP` – proměnná označující EtherType, 0x0800 pro IPv4 a 0x86DD pro IPv6,
- `eth` – dynamická funkce pro vytvoření ethernetového záhlaví, obsahuje parametry `daddr` (cílová MAC adresa), `saddr` (zdrojová MAC adresa) a `proto` (ethernetový typ),
- `ipv4` – dynamická funkce pro vytvoření IP záhlaví, obsahuje parametry `ttl` (Time To Live), `ver` (verze IP), `flags` (příznaky), `frag` (fragment offset), `df` („don't fragment“) a `da` (cílová IP adresa) s `sa` (zdrojová IP adresa),
- `udp` – dynamická funkce pro vytvoření UDP záhlaví, obsahuje parametry `sport` (zdrojový port) a `dport` (cílový port),
- DNS záhlaví – část konfiguračního souboru obsahující `drnd(2)` pro vygenerování náhodného ID, `const16(0x0100)` požadavek na rekurzivní vyřešení dotazu, požadavek o přeložení 1 domény, to že se jedná pouze o dotaz, nikoliv odpověď a 1 dodatečný záznam,
- Sekce DNS otázky – část konfiguračního souboru specifikující dotaz obsahující proměnnou pro samotnou doménu, typ DNS dotazu a třída záznamu IN definující, že se jedná o Internet,
- Dodatečný zánam DNS – doplňková sekce definuje EDNS0 záznam, který rozšiřuje možnosti DNS protokolu, jako podpora větších UDP paketů, EDNS verze 0 a žádná rozšiřující data jako např. DNSSEC.

Výpis 2.2: Konfigurační soubor pro DNS Query flood v IPv4.

```
#define ETH_P_IP  0x0800                                                    1
                                                                            2
{                                                                           3
  eth(daddr=##DnsQueryFloodSampler.dmac,                                    4
    saddr=##DnsQueryFloodSampler.smacTG, proto=ETH_P_IP),                   5
  ipv4(ttl=64, ver=4, flags=0, frag=0, df,                                  6
```

```
    da=##DnsQueryFloodSampler.targetIP,                          7
    sa=##DnsQueryFloodSampler.sourceIPTG),                       8
  udp(sport=##DnsQueryFloodSampler.sourcePortTG,                 9
    dport=##DnsQueryFloodSampler.dPort),                         10
                                                                 11
/*DNS header*/                                                   12
drnd(2), /*ID (randomized)*/                                     13
const16(0x0100), /*Flags RD=1*/                                  14
const16(1),  /*Question count*/                                  15
const16(0), /*Answer count*/                                     16
const16(0), /*Authority count*/                                  17
const16(1), /*Additional count*/                                 18
                                                                 19
/*DNS question section*/                                         20
##DnsQueryFloodSampler.queryTG                                   21
0x00, /*Question name*/                                          22
const16(1), /*Question type A, 28 - AAAA*/                       23
const16(1) /*Question class IN*/                                 24
                                                                 25
/*DNS additional section*/                                       26
0x00, /* "." */                                                  27
const16(41), /*OPT*/                                            28
const16(4096), /*UDP payload size*/                              29
0x00, /*Extended RCODE*/                                         30
0x00, /*EDNS version*/                                           31
0x00, 0x00, /*Z*/                                                32
const16(0)  /*Data length*/                                      33
}                                                                34
```

Další konfigurační soubory mají jinou strukturu znázorněno na výpisu 2.3. Jedná se o záplavové útoky zneužívající vlastnosti TCP protokolu, konkrétně se jedná o útoky ACK, FIN a RST flood. Každý z útoků má identickou strukturu, liší se pouze v definovaném příznaku.

- `ETH_P_IP` – proměnná označující EtherType, 0x0800 pro IPv4 a 0x86DD pro IPv6,
- `eth` – dynamická funkce pro vytvoření ethernetového záhlaví, obsahuje parametry `daddr` (cílová MAC adresa), `saddr` (zdrojová MAC adresa) a `proto` (ethernetový typ),
- `ipv4` – dynamická funkce pro vytvoření IP záhlaví, obsahuje parametry `ttl` (Time To Live), `ver` (verze IP), `flags` (příznaky), `frag` (fragment offset), `df` („don't fragment“) a `da` (cílová IP adresa) s `sa` (zdrojová IP adresa),

- `tcp` – dynamická funkce pro vytvoření TCp záhlaví, obsahuje parametry `sport` (zdrojový port), `dport` (cílový port), `seq` (sekvenční číslo), `aseq` (potvrzovací sekvenční číslo), `hlen` (délka záhlaví), `ack/rst/fin` (příznak definující typ TCP paketu) a `win` (délka okna),
- `fill` – dynamická proměnná pro nastavení obsahu paketu, díky parametru může uživatel nastavit velikost paketu v bajtech.

Výpis 2.3: Konfigurační soubor pro DNS ACK flood v IPv4.

```
#define ETH_P_IP  0x0800

{
  eth(daddr=##AckFloodSampler.dmac,
    saddr=##AckFloodSampler.smacTG, proto=ETH_P_IP),
  ipv4(ttl=##AckFloodSampler.ttl, ver=4,
    flags=0b01000000, frag=0, df,
    da=##AckFloodSampler.targetIP,
    sa=##AckFloodSampler.sourceIPTG),
  tcp(sport=##AckFloodSampler.sourcePortTG,
    dport=##AckFloodSampler.dPort, seq=drnd(), aseq=0,
    hlen=40, ack, win=##AckFloodSampler.winSize),
  fill('B', ##AckFloodSampler.payload),
}
```

2.2.2 Zahájení útoku v Apache JMeter

Před prvotním spuštěním Apache JMeter je nejdříve potřeba umístit sestavený modul `Ddos.jar` do adresáře `~/(cesta k Apache JMeter/lib/ext`. Po vložení souboru je již všechno připraveno k zahájení testování. Spustitelný soubor se nachází v `~/(cesta k Apache JMeter/bin/jmeter`, samotný program se spustí s příkazem `sudo ./jmeter`.

Po spuštění programu se zobrazí úvodní okno viz 1.8, jednotlivé DDoS útoky se můžou přidat kliknutím pravým myšítkem v levé části na `Test Plan` a následně `Add > Threads (Users) > DDoS Simple Thread Group`, tímto se přidá DDoS vlákno, které se objeví v levé části. Kliknutím pravým myšítkem na `DDoS Simple Thread Group` se zobrazí všechny útoky, které jsou součástí DDoS modulu.

Zpracování požadavků Apache JMeter je zobrazeno na diagramu 2.2. Uživatel resp. útočník zadá vstupní data jako je cílová IP adresa, MAC adresy, porty atd., následně se vstupní data zpracují a vloží se do proměnných v Java Sampleru, Java Sampler vyhodnotí vstupní data, přiřadí proměnné do parametrů správného konfiguračního souboru a nakonec spustí samotný záplavový útok na oběť.

2.2.3 Uživatelské rozhraní v Apache JMeter

V rámci bakalářské práce byla vytvořena 2 různé uživatelské rozhraní. Jedno pro DNS Query Flood a druhé rozhraní se znovupoužívá pro ACK, FIN a RST Flood. V následujících podkapitolách budou jednotlivá uživatelská rozhraní popsána včetně možností co umožňují. Každé textové pole má automaticky předvyplněné hodnoty, ale uživatel má svobodu hodnoty jakkoliv pozměnit dle svého uvážení.

DNS Query Flood GUI

Uživatelské rozhraní na obr. 2.1 je rozděleno do 4 částí. V první vrchní části se nachází název útoku, případně doplněn o komentář a síťové zařízení použité pro odesílání generovaného provozu do cíle, uživatel má možnost zobrazit všechny automaticky detekované síťové zařízení a zvolit, které použije.

V červeném rámečku se nachází nastavení na straně útočníka. Uživatel má možnost buď zadat jednu zdrojovou MAC adresu, nebo zvolit náhodné generování ze zadaného rozsahu. Stejnou možnost má i pro IPv4 a IPv6 adresy, nicméně pokud chce použít IPv6 adresu, tak nejprve bude muset s možností *Enable IPv6* v zeleném rámečku ji aktivovat. Jak v předchozích případech, tak i u portu má uživatel možnost zvolit buď jeden port, nebo zvolit náhodné generování ze zadaného rozsahu.

V zeleném rámečku jsou umístěné hodnoty oběti – DNS serveru. Uživatel může nastavit cílovou MAC adresu, IPv4 nebo IPv6 adresu, port, který obvykle bude vždy 53 v případě DNS serveru a textové pole domény pro překlad. Je tu zde i možnost výběru náhodného generování domény, po výběru této možnosti se uživatelovi zpřístupní výběr délky náhodně generovaného doménového jména a TLD.

V posledním modrém rámečku je nastavení intenzity útoku. První možnost je počet paketů a druhá rychlost posílání v *pps - packet per second*.

DNS ACK/FIN/RST Flood GUI

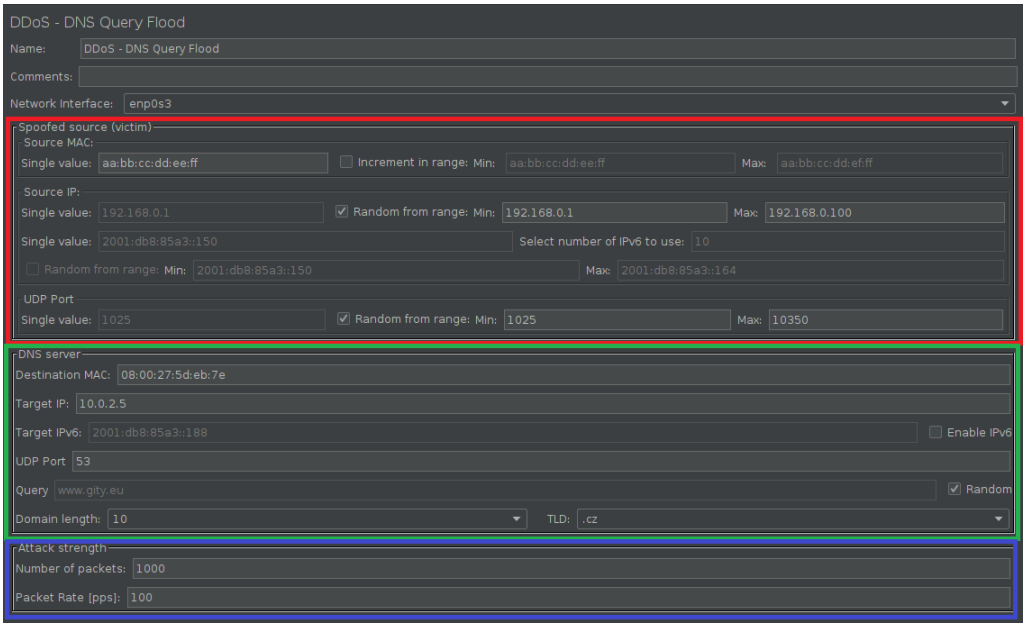
Uživatelské rozhraní na obr. 2.3 je shodné pro ACK, FIN i RST záplavový útok a je rozdělen do 5 částí. V první vrchní části se stejně jak v předchozím případě nachází název útoku, případně doplněn o komentář a síťové zařízení, které se použije pro odesílání generovaného provozu.

V červeném rámečku se nachází nastavení *linkové vrstvy*, zde uživatel nastaví buď jednu zdrojovou MAC adresu, nebo má možnost zvolit náhodné generování a MAC adresu adresáta záplavového útoku.

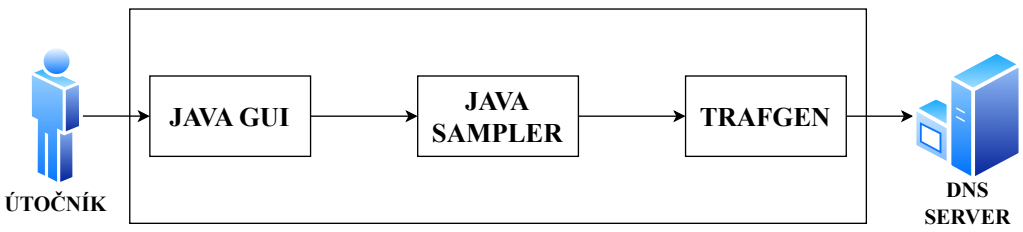
V zeleném rámečku se nastavují okolnosti týkající se *síťové vrstvy*. Uživatel zadá zdrojovou a cílovou IPv4 adresu, případně IPv6 pokud to povolí. V obou případech má zase možnost zvolit náhodné generování adres z rozsahu. Pokud uživatel bude chtít může zvolit i životnost paketu.

V modrém rámečku je *transportní vrstva*, kde se nastaví zdrojový port, včetně možnosti náhodného generování, a cílový port. Dá se tu také nastavit délka okna. Nakonec se v modrém rámečku ještě nachází *payload*, kde se dá nastavit velikost dat v bajtech.

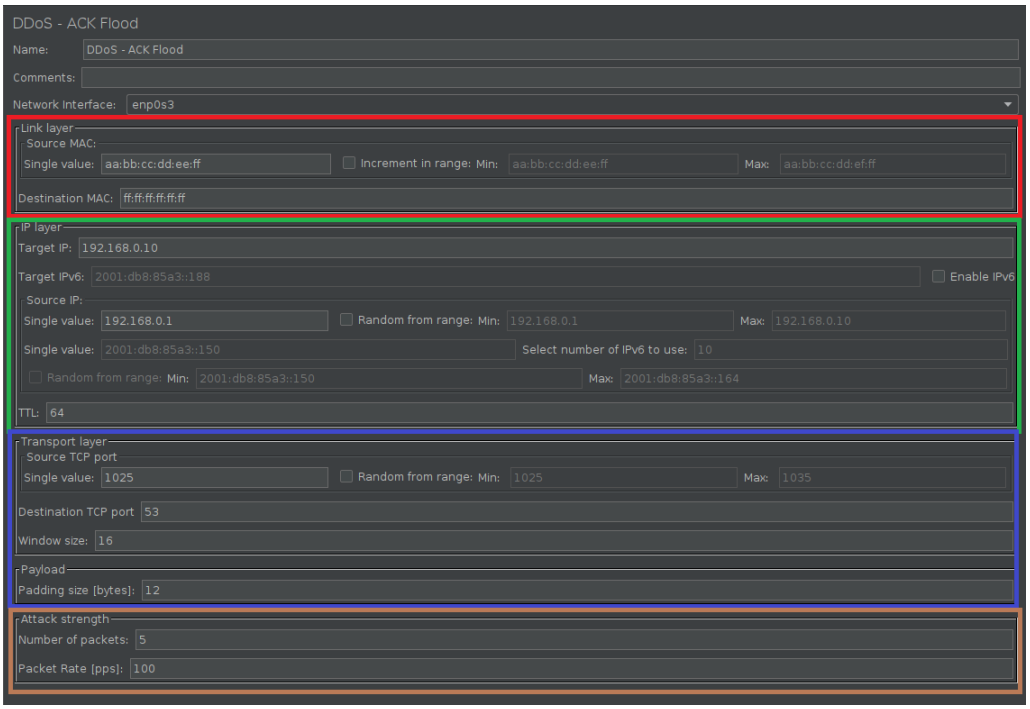
V posledním hnědém rámečku je nastavení intenzity útoku, stejně jak v předchozím uživatelském rozhraní.



Obr. 2.1: Uživatelské rozhraní DNS Query Flood.



Obr. 2.2: Diagram průběhu útoku.



Obr. 2.3: Uživatelské rozhraní DNS ACK Flood.

2.3 Testování útoků

V této kapitole se nachází jednotlivé výsledky samotného testování útoků. Každý realizovaný scénář obsahuje informace ohledně rychlosti posílaných paketů a rychlosti odesílaných dat, zatížení RAM paměti před a po útoku, ukázku příchozích a odchozích DNS dotazů a zatížení CPU.

2.3.1 Scénář č. 1 – DNS query flood

První testovací scénář je zaměřený na zahlcení DNS serveru pomocí žádostí o překlad na neexistující domény. Spočívá v tom, že DNS server musí vyhovět žádosti o překlad, takže bude muset prohledat své záznamy a následně odpovědět. Jelikož doména v DNS dotazu je neexistující a ještě navíc IP adresa žadatele je též neexistující, tak DNS server zbytečně využívá své prostředky. V případě jednoho dotazu je to zanedbatelné, ale pokud takových požadavků je v řádech desetitisíců a více, se stává server naprosto zahlcen a vzniká odepření služby legitimním uživatelům. V tab. 2.1 jsou uvedeny rychlosti odesílaných paketů a dat, které byly nastaveny v Apache JMeter a na obr. 2.4 zatížení paměti RAM před provedením útoku.

Tab. 2.1: Informace o útoku DNS query flood.

Počet paketů	Rychlost posílání paketů
10 000 paketů	1000 pps

```
MiB Mem :   1967.9 total,   1586.4 free,    302.0 used,    225.1 buff/cache
```

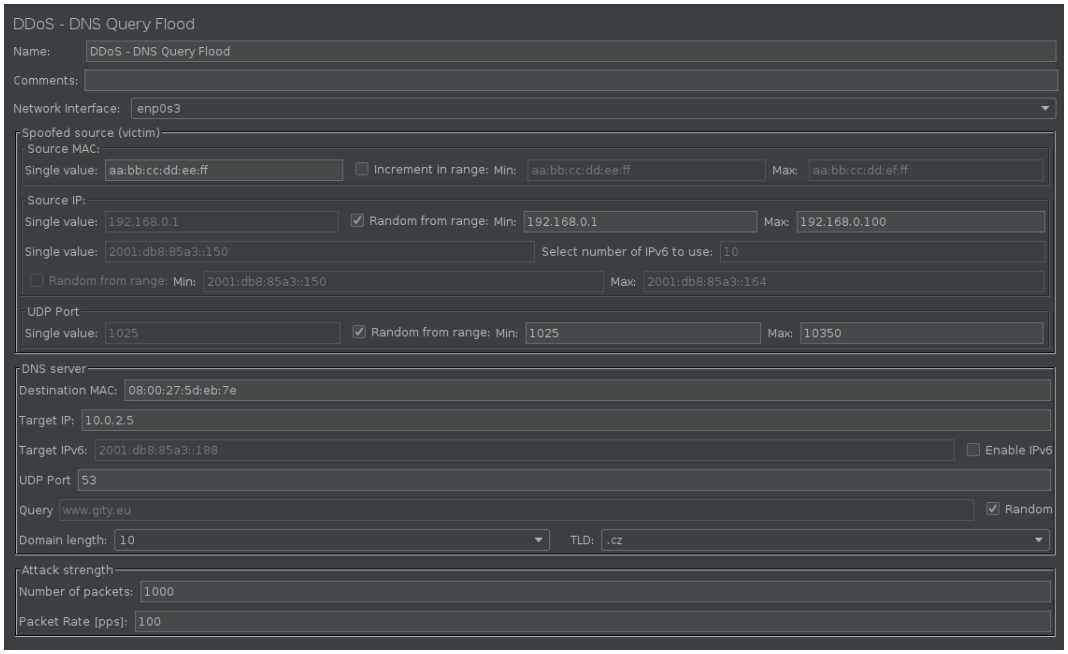
Obr. 2.4: Zatížení RAM před útokem DNS query flood.

Realizace prvního scénaře

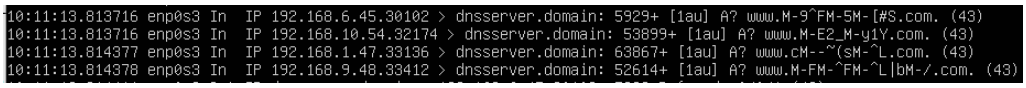
Před realizací samotného testu byla nejprve ověřena konektivita mezi zařízeními pomocí příkazu `ping`. Po úspěšném ověření viditelnosti byl přidán modul *DNS Query Flood*, jak bylo popsáno v podkapitole 2.2.2.

Byly vyplněny hodnoty potřebné pro zahájení testu viz obr. 2.5, bylo zvoleno náhodné generování doménových jmen a varianta pro IPv4 adresy.

Po spuštění testu bylo zahájeno generování provozu, na cílovém zařízení probíhalo monitorování příchozích DNS dotazů, měření zatížení CPU a sledování volné paměti RAM. Příchozí DNS dotazy jsou zobrazeny na obr. 2.6. Zatížení procesoru se pro všechna jádra v průměru pohybuje kolem 30 %, vyčteno z posledního sloupce na obr. 2.8. Viz obr. 2.7 je volná paměť RAM po provedení útoku 215 MB, tudíž 1371 MB z 1586 MB volné paměti bylo během testu zabráno.

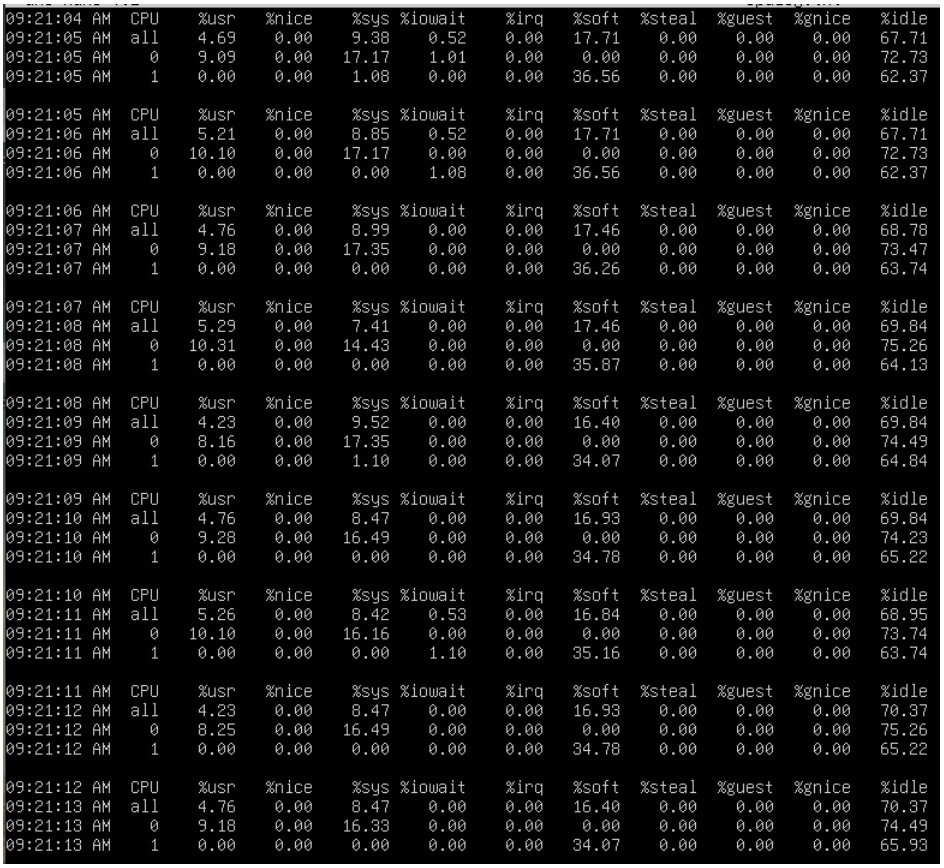


Obr. 2.5: Apache JMeter rozhraní připraveno pro realizaci prvního scénáře.



Obr. 2.6: Příchozí DNS dotazy na server u DNS query flood.

Obr. 2.7: Zatížení RAM po útoku DNS query flood.



09:21:04 AM	CPU	%usr	%nice	%sys	%iowait	%irq	%soft	%steal	%guest	%gnice	%idle
09:21:05 AM	all	4.69	0.00	9.38	0.52	0.00	17.71	0.00	0.00	0.00	67.71
09:21:05 AM	0	9.09	0.00	17.17	1.01	0.00	0.00	0.00	0.00	0.00	72.73
09:21:05 AM	1	0.00	0.00	1.08	0.00	0.00	36.56	0.00	0.00	0.00	62.37
09:21:05 AM	CPU	%usr	%nice	%sys	%iowait	%irq	%soft	%steal	%guest	%gnice	%idle
09:21:06 AM	all	5.21	0.00	8.85	0.52	0.00	17.71	0.00	0.00	0.00	67.71
09:21:06 AM	0	10.10	0.00	17.17	0.00	0.00	0.00	0.00	0.00	0.00	72.73
09:21:06 AM	1	0.00	0.00	0.00	1.08	0.00	36.56	0.00	0.00	0.00	62.37
09:21:06 AM	CPU	%usr	%nice	%sys	%iowait	%irq	%soft	%steal	%guest	%gnice	%idle
09:21:07 AM	all	4.76	0.00	8.99	0.00	0.00	17.46	0.00	0.00	0.00	68.78
09:21:07 AM	0	9.18	0.00	17.35	0.00	0.00	0.00	0.00	0.00	0.00	73.47
09:21:07 AM	1	0.00	0.00	0.00	0.00	0.00	36.26	0.00	0.00	0.00	63.74
09:21:07 AM	CPU	%usr	%nice	%sys	%iowait	%irq	%soft	%steal	%guest	%gnice	%idle
09:21:08 AM	all	5.29	0.00	7.41	0.00	0.00	17.46	0.00	0.00	0.00	69.84
09:21:08 AM	0	10.31	0.00	14.43	0.00	0.00	0.00	0.00	0.00	0.00	75.26
09:21:08 AM	1	0.00	0.00	0.00	0.00	0.00	35.87	0.00	0.00	0.00	64.13
09:21:08 AM	CPU	%usr	%nice	%sys	%iowait	%irq	%soft	%steal	%guest	%gnice	%idle
09:21:09 AM	all	4.23	0.00	9.52	0.00	0.00	16.40	0.00	0.00	0.00	69.84
09:21:09 AM	0	8.16	0.00	17.35	0.00	0.00	0.00	0.00	0.00	0.00	74.49
09:21:09 AM	1	0.00	0.00	1.10	0.00	0.00	34.07	0.00	0.00	0.00	64.84
09:21:09 AM	CPU	%usr	%nice	%sys	%iowait	%irq	%soft	%steal	%guest	%gnice	%idle
09:21:10 AM	all	4.76	0.00	8.47	0.00	0.00	16.93	0.00	0.00	0.00	69.84
09:21:10 AM	0	9.28	0.00	16.49	0.00	0.00	0.00	0.00	0.00	0.00	74.23
09:21:10 AM	1	0.00	0.00	0.00	0.00	0.00	34.78	0.00	0.00	0.00	65.22
09:21:10 AM	CPU	%usr	%nice	%sys	%iowait	%irq	%soft	%steal	%guest	%gnice	%idle
09:21:11 AM	all	5.26	0.00	8.42	0.53	0.00	16.84	0.00	0.00	0.00	68.95
09:21:11 AM	0	10.10	0.00	16.16	0.00	0.00	0.00	0.00	0.00	0.00	73.74
09:21:11 AM	1	0.00	0.00	0.00	1.10	0.00	35.16	0.00	0.00	0.00	63.74
09:21:11 AM	CPU	%usr	%nice	%sys	%iowait	%irq	%soft	%steal	%guest	%gnice	%idle
09:21:12 AM	all	4.23	0.00	8.47	0.00	0.00	16.93	0.00	0.00	0.00	70.37
09:21:12 AM	0	8.25	0.00	16.49	0.00	0.00	0.00	0.00	0.00	0.00	75.26
09:21:12 AM	1	0.00	0.00	0.00	0.00	0.00	34.78	0.00	0.00	0.00	65.22
09:21:12 AM	CPU	%usr	%nice	%sys	%iowait	%irq	%soft	%steal	%guest	%gnice	%idle
09:21:13 AM	all	4.76	0.00	8.47	0.00	0.00	16.40	0.00	0.00	0.00	70.37
09:21:13 AM	0	9.18	0.00	16.33	0.00	0.00	0.00	0.00	0.00	0.00	74.49
09:21:13 AM	1	0.00	0.00	0.00	0.00	0.00	34.07	0.00	0.00	0.00	65.93

Obr. 2.8: Zatížení CPU na serveru při DNS query flood.

2.3.2 Scénář č. 2 – DNS ACK flood

Druhý testovací scénář je zaměřený na zahlcení DNS serveru pakety o potvrzení spojení. Stejně jak v předchozím případě, DNS server musí zpracovat všechny příchozí pakety. Potvrzovací paket, který je zaslán na DNS server je vyhodnocen, ale protože obsah, který daný paket obsahuje je nevýznamný, tak je server zbytečně zahlcován. V případě posílání několika takových paketů se provoz stává saturovaný a je obtížné odlišit legitimní provoz od nelegitimního. V tab. 2.2 jsou uvedeny hodnoty počtu paketů, které budou odeslány a v jaké rychlosti budou posílány a na obr. 2.9 je zobrazena volná paměť RAM před provedením útoku.

Tab. 2.2: Informace o útoku DNS ACK flood.

Počet paketů	Rychlost posílání paketů
20 000 paketů	1000 pps

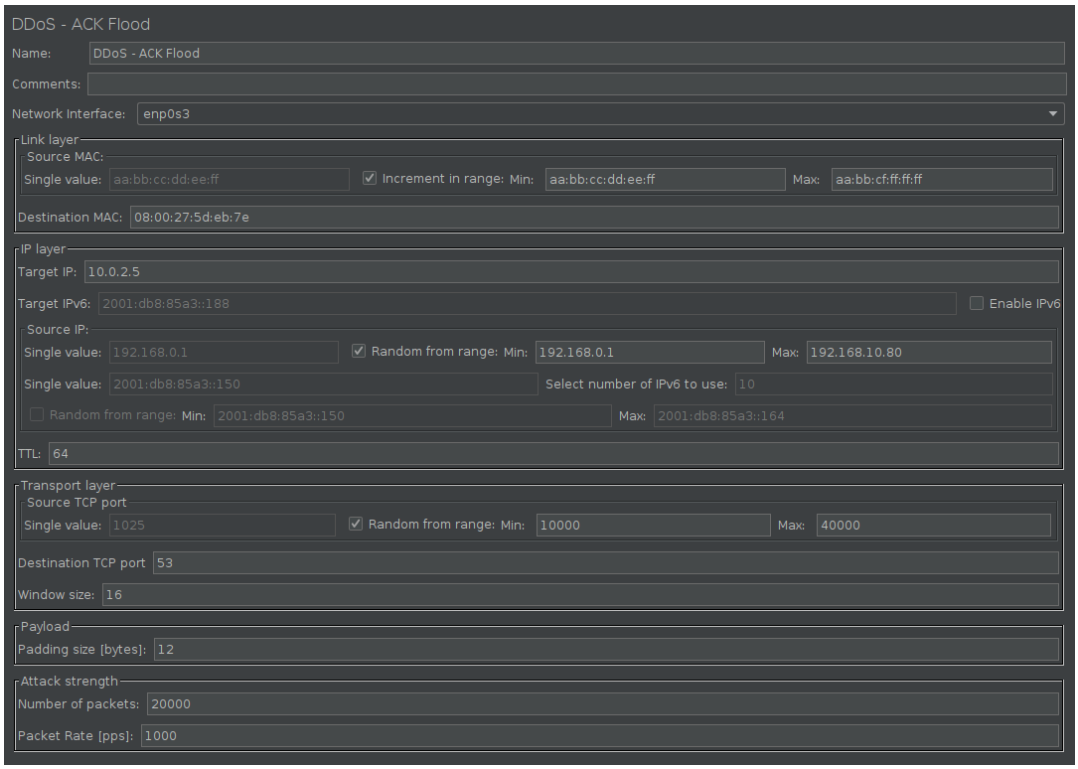
```
MiB Mem :   1967.8 total,   1579.6 free,    292.2 used,    241.9 buff/cache
```

Obr. 2.9: Zatížení RAM před útokem DNS ACK flood.

Realizace druhého scénáře

Nejprve se odstraní dříve přidané testy, případně se vytvoří nové vlákno v Apache JMeter, následně se přidá modul *DNS ACK Flood* a vyplní se potřebné hodnoty viz obr. 2.10.

Po spuštění testu bylo zahájeno generování provozu, na cílovém zařízení probíhalo monitorování příchozích ACK paketů, měření zatížení CPU a sledování paměti RAM. Příchozí ACK pakety jsou zobrazeny na obr. 2.11. Zatížení procesoru se pro všechna jádra v průměru pohybuje kolem 7 %, z obr. 2.13. Z obr. 2.12 lze vyčíst, že volná paměť RAM se nikoliv nezměnila.



Obr. 2.10: Apache JMeter rozhraní připraveno pro realizaci druhého scénáře.

```
19976 19.387482975 192.168.5.46 → 10.0.2.5     TCP 66 23867 → 53 [ACK] Seq=1 Ack=1 Win=16 Len=0
19977 19.387805195 192.168.10.50 → 10.0.2.5     TCP 66 34444 → 53 [ACK] Seq=1 Ack=1 Win=16 Len=0
19978 19.388325956 192.168.5.182 → 10.0.2.5     TCP 66 13724 → 53 [ACK] Seq=1 Ack=1 Win=16 Len=0
19979 19.388326126 192.168.5.137 → 10.0.2.5     TCP 66 39629 → 53 [ACK] Seq=1 Ack=1 Win=16 Len=0
19980 19.390359118 192.168.10.72 → 10.0.2.5     TCP 66 37028 → 53 [ACK] Seq=1 Ack=1 Win=16 Len=0
19981 19.390359508  192.168.2.6 → 10.0.2.5     TCP 66 31237 → 53 [ACK] Seq=1 Ack=1 Win=16 Len=0
19982 19.390698989 192.168.9.160 → 10.0.2.5     TCP 66 31628 → 53 [ACK] Seq=1 Ack=1 Win=16 Len=0
19983 19.390699169 192.168.0.84 → 10.0.2.5     TCP 66 15956 → 53 [ACK] Seq=1 Ack=1 Win=16 Len=0
19984 19.391019179 192.168.7.125 → 10.0.2.5     TCP 66 26395 → 53 [ACK] Seq=1 Ack=1 Win=16 Len=0
19985 19.391498899 192.168.5.29 → 10.0.2.5     TCP 66 14218 → 53 [ACK] Seq=1 Ack=1 Win=16 Len=0
19986 19.391818909 192.168.8.200 → 10.0.2.5     TCP 66 21722 → 53 [ACK] Seq=1 Ack=1 Win=16 Len=0
19987 19.399039966 192.168.4.110 → 10.0.2.5     TCP 66 16683 → 53 [ACK] Seq=1 Ack=1 Win=16 Len=0
19988 19.399040146 192.168.9.102 → 10.0.2.5     TCP 66 19535 → 53 [ACK] Seq=1 Ack=1 Win=16 Len=0
19989 19.399040206 192.168.8.168 → 10.0.2.5     TCP 66 18595 → 53 [ACK] Seq=1 Ack=1 Win=16 Len=0
19990 19.399040266 192.168.8.12 → 10.0.2.5     TCP 66 22595 → 53 [ACK] Seq=1 Ack=1 Win=16 Len=0
19991 19.399040326 192.168.0.183 → 10.0.2.5     TCP 66 22123 → 53 [ACK] Seq=1 Ack=1 Win=16 Len=0
19992 19.399040386 192.168.2.149 → 10.0.2.5     TCP 66 13322 → 53 [ACK] Seq=1 Ack=1 Win=16 Len=0
19993 19.399040446 192.168.4.65 → 10.0.2.5     TCP 66 11044 → 53 [ACK] Seq=1 Ack=1 Win=16 Len=0
19994 19.399040516 192.168.6.210 → 10.0.2.5     TCP 66 36548 → 53 [ACK] Seq=1 Ack=1 Win=16 Len=0
19995 19.399253416 192.168.9.22 → 10.0.2.5     TCP 66 23715 → 53 [ACK] Seq=1 Ack=1 Win=16 Len=0
19996 19.399253586 192.168.5.35 → 10.0.2.5     TCP 66 39744 → 53 [ACK] Seq=1 Ack=1 Win=16 Len=0
19997 19.399253646 192.168.5.105 → 10.0.2.5     TCP 66 12458 → 53 [ACK] Seq=1 Ack=1 Win=16 Len=0
19998 19.399253706 192.168.5.235 → 10.0.2.5     TCP 66 33825 → 53 [ACK] Seq=1 Ack=1 Win=16 Len=0
19999 19.399253766 192.168.4.85 → 10.0.2.5     TCP 66 21513 → 53 [ACK] Seq=1 Ack=1 Win=16 Len=0
20000 19.399253836 192.168.9.149 → 10.0.2.5     TCP 66 18514 → 53 [ACK] Seq=1 Ack=1 Win=16 Len=0
```

Obr. 2.11: Příchozí ACK pakety na server.

```
MiB Mem :   1967.8 total,   1536.5 free,    307.0 used,    270.5 buff/cache
```

Obr. 2.12: Zatížení RAM po útoku DNS ACK flood.

```
10:50:56 AM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
10:50:57 AM  all    0.00    0.00    0.00    0.55    0.00    7.73    0.00    0.00    0.00   91.71
10:50:57 AM    0    0.00    0.00    0.00    0.00    0.00    1.03    0.00    0.00    0.00   98.97
10:50:57 AM    1    0.00    0.00    0.00    1.19    0.00   15.48    0.00    0.00    0.00   83.33

10:50:57 AM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
10:50:58 AM  all    0.00    0.00    0.00    0.00    0.00    6.78    0.00    0.00    0.00   93.22
10:50:58 AM    0    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
10:50:58 AM    1    0.00    0.00    0.00    0.00    0.00   14.63    0.00    0.00    0.00   85.37

10:50:58 AM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
10:50:59 AM  all    0.00    0.00    0.00    0.00    0.00    6.15    0.00    0.00    0.00   93.85
10:50:59 AM    0    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
10:50:59 AM    1    0.00    0.00    0.00    0.00    0.00   13.25    0.00    0.00    0.00   86.75

10:50:59 AM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
10:51:00 AM  all    0.00    0.00    0.00    0.57    0.00    6.25    0.00    0.00    0.00   93.18
10:51:00 AM    0    0.00    0.00    0.00    1.05    0.00    0.00    0.00    0.00    0.00   98.95
10:51:00 AM    1    0.00    0.00    0.00    0.00    0.00   13.58    0.00    0.00    0.00   86.42

10:51:00 AM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
10:51:01 AM  all    0.00    0.00    0.00    0.00    0.00    4.00    0.00    0.00    0.00   96.00
10:51:01 AM    0    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
10:51:01 AM    1    0.00    0.00    0.00    0.00    0.00    8.86    0.00    0.00    0.00   91.14

10:51:01 AM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
10:51:02 AM  all    0.00    0.00    0.00    0.00    0.00    7.30    0.00    0.00    0.00   92.70
10:51:02 AM    0    0.00    0.00    0.00    0.00    0.00    1.04    0.00    0.00    0.00   98.96
10:51:02 AM    1    0.00    0.00    0.00    0.00    0.00   14.63    0.00    0.00    0.00   85.37

10:51:02 AM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
10:51:03 AM  all    0.00    0.00    0.00    0.00    0.00    6.74    0.00    0.00    0.00   93.26
10:51:03 AM    0    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
10:51:03 AM    1    0.00    0.00    0.00    0.00    0.00   14.63    0.00    0.00    0.00   85.37

10:51:03 AM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
10:51:04 AM  all    0.00    0.00    0.00    0.00    0.00    9.29    0.00    0.00    0.00   90.71
10:51:04 AM    0    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
10:51:04 AM    1    0.00    0.00    0.00    0.00    0.00   19.32    0.00    0.00    0.00   80.68

10:51:04 AM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
10:51:05 AM  all    0.00    0.00    0.00    0.00    0.00    8.47    0.00    0.00    0.00   91.53
10:51:05 AM    0    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
10:51:05 AM    1    0.00    0.00    0.00    0.00    0.00   17.78    0.00    0.00    0.00   82.22
```

Obr. 2.13: Zatížení CPU na serveru při DNS ACK flood.

2.3.3 Scénář č. 3 – DNS FIN flood

Třetí testovací scénář je zaměřený na zahlcení DNS serveru pakety o ukončení spojení. Pakety jsou posílané s falešnou IP adresou, takže při vyhodnocování FIN paketu server musí alokovat systémové zdroje, ale žádná dříve probíhající komunikace není na serveru nalezena. V tab. 2.3 jsou uvedeny hodnoty počtu paketů, které budou odeslány a v jaké rychlosti budou posílány a na obr.2.14 je zobrazena volná paměť RAM před provedením útoku.

Tab. 2.3: Informace o útoku DNS FIN flood.

Počet paketů	Rychlost posílání paketů
20 000 paketů	1000 pps

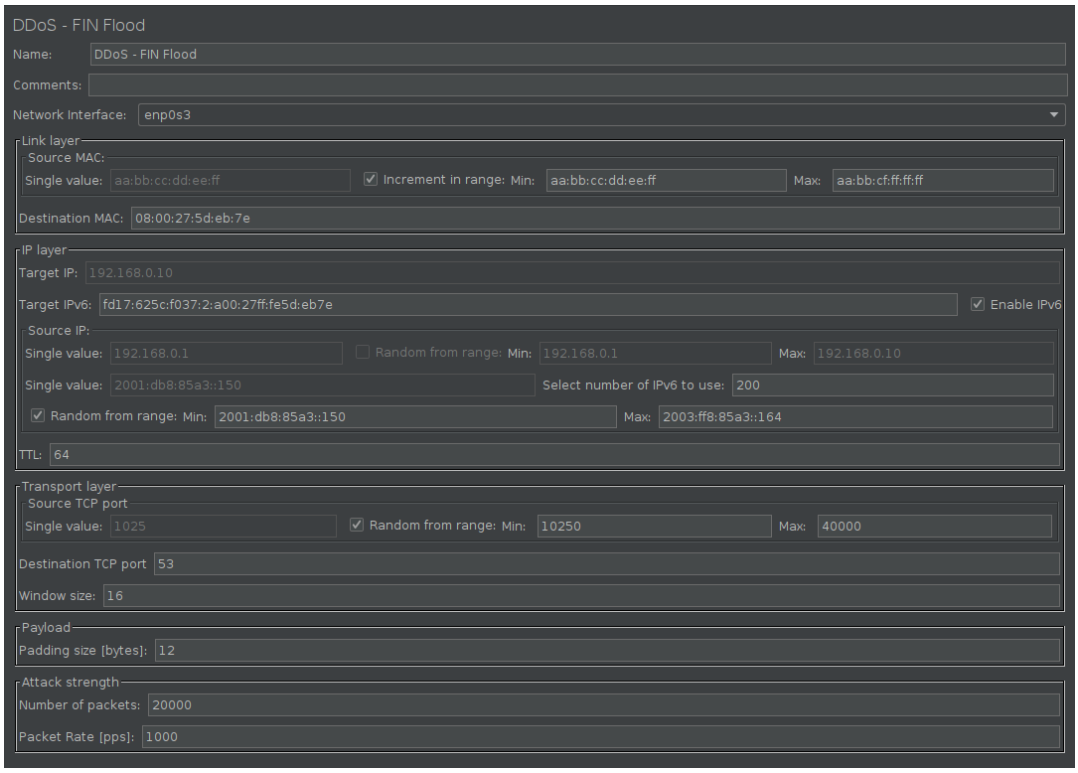
```
MiB Mem :   1967.8 total,   1531.7 free,    303.7 used,    279.2 buff/cache
```

Obr. 2.14: Zatížení RAM před útokem DNS FIN flood.

Realizace třetího scénáře

Nejprve se odstraní dříve přidané testy, případně se vytvoří nové vlákno v Apache JMeter, následně se přidá modul *DNS FIN Flood* a vyplní se potřebné hodnoty viz obr. 2.15. V tomto testu byla zvolena varianta pro IPv6 adresy.

Příchozí FIN pakety jsou zobrazeny na obr. 2.16. Zatížení procesoru se pro všechna jádra v průměru pohybuje kolem 16 %, z obr. 2.18. Z obr. 2.17 lze vyčíst, že volná paměť RAM se nijak signifikantně nezměnila.



Obr. 2.15: Apache JMeter rozhraní připraveno pro realizaci třetího scénáře.

```
19976 19.413721301 2001:c679:9aea:9095:1b8f:4a01:2e80:60ba → fd17:625c:f037:2:a00:27ff:fe5d:eb7e TCP 86 30235 → 53 [FIN] Seq=1 Win=16 Len=0
19977 19.413721361 2002:7511:c9f1:eb8f:59f6:cd9a:948b:f190 → fd17:625c:f037:2:a00:27ff:fe5d:eb7e TCP 86 22135 → 53 [FIN] Seq=1 Win=16 Len=0
19978 19.413721421 2002:f4c4:d9d5:d92:e43b:704d:b6ae:c8d2 → fd17:625c:f037:2:a00:27ff:fe5d:eb7e TCP 86 27092 → 53 [FIN] Seq=1 Win=16 Len=0
19979 19.413746430 2002:f8e:630f:4981:48b4:660e:4123:d409 → fd17:625c:f037:2:a00:27ff:fe5d:eb7e TCP 86 25653 → 53 [FIN] Seq=1 Win=16 Len=0
19980 19.413746500 2001:6dda:85aa:bf5e:680c:437f:8d2b:a56d → fd17:625c:f037:2:a00:27ff:fe5d:eb7e TCP 86 24795 → 53 [FIN] Seq=1 Win=16 Len=0
19981 19.413746570 2001:23d9:fad0:c12e:5e4e:7cde:af71:4132 → fd17:625c:f037:2:a00:27ff:fe5d:eb7e TCP 86 38077 → 53 [FIN] Seq=1 Win=16 Len=0
19982 19.413746640 2001:f47d:36f4:566e:8607:d263:951c:ced4 → fd17:625c:f037:2:a00:27ff:fe5d:eb7e TCP 86 14510 → 53 [FIN] Seq=1 Win=16 Len=0
19983 19.413746700 2001:d861:3182:7e96:3524:fafb:338b:d28c → fd17:625c:f037:2:a00:27ff:fe5d:eb7e TCP 86 32674 → 53 [FIN] Seq=1 Win=16 Len=0
19984 19.413746760 2001:6127:2fdd:9497:d39a:13bd:414:a01f → fd17:625c:f037:2:a00:27ff:fe5d:eb7e TCP 86 39065 → 53 [FIN] Seq=1 Win=16 Len=0
19985 19.413746820 2001:28cf:3e20:7cb4:a901:686e:1ccc:976e → fd17:625c:f037:2:a00:27ff:fe5d:eb7e TCP 86 33076 → 53 [FIN] Seq=1 Win=16 Len=0
19986 19.413746880 2003:6b2:83f1:9cec:2d1c:dbc8:6955:2288 → fd17:625c:f037:2:a00:27ff:fe5d:eb7e TCP 86 38754 → 53 [FIN] Seq=1 Win=16 Len=0
19987 19.413799161 2002:b036:8609:f728:e33b:f208:3f6e:3cbd → fd17:625c:f037:2:a00:27ff:fe5d:eb7e TCP 86 27604 → 53 [FIN] Seq=1 Win=16 Len=0
19988 19.413799321 2001:1e20:3ccd:127e:6dd3:8331:10ec:c487 → fd17:625c:f037:2:a00:27ff:fe5d:eb7e TCP 86 32390 → 53 [FIN] Seq=1 Win=16 Len=0
19989 19.413799381 2002:8380:5d2:16f9:e218:8fd7:52d2:cc22 → fd17:625c:f037:2:a00:27ff:fe5d:eb7e TCP 86 17753 → 53 [FIN] Seq=1 Win=16 Len=0
19990 19.413799441 2002:c333:d37f:298:55b5:fdc3:f53e:cc9 → fd17:625c:f037:2:a00:27ff:fe5d:eb7e TCP 86 14429 → 53 [FIN] Seq=1 Win=16 Len=0
19991 19.413799511 2002:dbd9:6e09:59f7:24a8:59fd:b278:ece4 → fd17:625c:f037:2:a00:27ff:fe5d:eb7e TCP 86 10955 → 53 [FIN] Seq=1 Win=16 Len=0
19992 19.413799571 2001:402a:34c8:85e4:453d:ee36:1e4a:928f → fd17:625c:f037:2:a00:27ff:fe5d:eb7e TCP 86 10955 → 53 [FIN] Seq=1 Win=16 Len=0
19993 19.413799631 2002:f4c4:d9d5:d92:e43b:704d:b6ae:c8d2 → fd17:625c:f037:2:a00:27ff:fe5d:eb7e TCP 86 17591 → 53 [FIN] Seq=1 Win=16 Len=0
19994 19.413799691 2001:d359:1854:ca5b:673f:95ee:7804:2746 → fd17:625c:f037:2:a00:27ff:fe5d:eb7e TCP 86 33662 → 53 [FIN] Seq=1 Win=16 Len=0
19995 19.413849581 2001:88c3:ecf0:a738:3fd3:fbe6:dd69:3d83 → fd17:625c:f037:2:a00:27ff:fe5d:eb7e TCP 86 13795 → 53 [FIN] Seq=1 Win=16 Len=0
19996 19.413849751 2002:3fc6:8a03:d976:170d:5dfc:aa19:fe91 → fd17:625c:f037:2:a00:27ff:fe5d:eb7e TCP 86 22206 → 53 [FIN] Seq=1 Win=16 Len=0
19997 19.413849811 2002:edbf:c82a:5a8f:12d0:743e:776d:e98f → fd17:625c:f037:2:a00:27ff:fe5d:eb7e TCP 86 29754 → 53 [FIN] Seq=1 Win=16 Len=0
19998 19.413849871 2002:df19:b536:cec6:1384:34df:1b9f:19db → fd17:625c:f037:2:a00:27ff:fe5d:eb7e TCP 86 35664 → 53 [FIN] Seq=1 Win=16 Len=0
19999 19.413849931 2001:d861:3182:7e96:3524:fafb:338b:d28c → fd17:625c:f037:2:a00:27ff:fe5d:eb7e TCP 86 14762 → 53 [FIN] Seq=1 Win=16 Len=0
20000 19.413849991 2002:4270:1305:dc0f:e919:9196:7616:41ad → fd17:625c:f037:2:a00:27ff:fe5d:eb7e TCP 86 14378 → 53 [FIN] Seq=1 Win=16 Len=0
```

Obr. 2.16: Příchozí FIN pakety na server.

```
MiB Mem :   1967.8 total,   1567.5 free,    305.4 used,    240.7 buff/cache
```

Obr. 2.17: Zatížení RAM po útoku DNS FIN flood.

```
11:56:58 AM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
11:56:59 AM  all    0.00    0.00    0.00    0.00    0.00   15.62    0.00    0.00    0.00   84.38
11:56:59 AM    0    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
11:56:59 AM    1    0.00    0.00    0.00    0.00    0.00   32.61    0.00    0.00    0.00   67.39

11:56:59 AM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
11:57:00 AM  all    0.00    0.00    0.00    0.00    0.00   14.74    0.00    0.00    0.00   85.26
11:57:00 AM    0    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
11:57:00 AM    1    0.00    0.00    0.00    0.00    0.00   30.77    0.00    0.00    0.00   69.23

11:57:00 AM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
11:57:01 AM  all    0.00    0.00    0.00    0.00    0.00   15.62    0.00    0.00    0.00   84.38
11:57:01 AM    0    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
11:57:01 AM    1    0.00    0.00    0.00    0.00    0.00   32.61    0.00    0.00    0.00   67.39

11:57:01 AM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
11:57:02 AM  all    0.00    0.00    0.52    0.00    0.00   15.10    0.00    0.00    0.00   84.38
11:57:02 AM    0    0.00    0.00    1.00    0.00    0.00    0.00    0.00    0.00    0.00   99.00
11:57:02 AM    1    0.00    0.00    0.00    0.00    0.00   31.52    0.00    0.00    0.00   68.48

11:57:02 AM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
11:57:03 AM  all    0.00    0.00    0.00    0.53    0.00   14.74    0.00    0.00    0.00   84.74
11:57:03 AM    0    0.00    0.00    0.00    1.01    0.00    0.00    0.00    0.00    0.00   98.99
11:57:03 AM    1    0.00    0.00    0.00    0.00    0.00   30.77    0.00    0.00    0.00   69.23

11:57:03 AM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
11:57:04 AM  all    0.00    0.00    0.00    0.00    0.00   15.62    0.00    0.00    0.00   84.38
11:57:04 AM    0    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
11:57:04 AM    1    0.00    0.00    0.00    0.00    0.00   32.26    0.00    0.00    0.00   67.74

11:57:04 AM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
11:57:05 AM  all    0.00    0.00    0.00    0.00    0.00   16.15    0.00    0.00    0.00   83.85
11:57:05 AM    0    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
11:57:05 AM    1    0.00    0.00    0.00    0.00    0.00   33.70    0.00    0.00    0.00   66.30

11:57:05 AM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
11:57:06 AM  all    0.00    0.00    0.00    0.52    0.00   15.54    0.00    0.00    0.00   83.94
11:57:06 AM    0    0.00    0.00    0.00    1.00    0.00    0.00    0.00    0.00    0.00   99.00
11:57:06 AM    1    0.00    0.00    0.00    0.00    0.00   32.26    0.00    0.00    0.00   67.74

11:57:06 AM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
11:57:07 AM  all    0.00    0.00    0.52    0.00    0.00   15.10    0.00    0.00    0.00   84.38
11:57:07 AM    0    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
11:57:07 AM    1    0.00    0.00    1.09    0.00    0.00   31.52    0.00    0.00    0.00   67.39
```

Obr. 2.18: Zatížení CPU na serveru při DNS FIN flood.

2.3.4 Scénář č. 4 – DNS RST flood

Čtvrtý testovací scénář je zaměřený na zahlcení DNS serveru pakety o resetování spojení. Princip útok je stejný jako u scénáře s FIN zahlcením v podkapitole 2.3.3, tudíž server musí alokovat systémové zdroje na neexistující komunikaci. V tab. 2.4 jsou uvedeny hodnoty počtu paketů, které budou odeslány a v jaké rychlosti budou posílány a na obr.2.19 je zobrazena volná paměť RAM před provedením útoku.

Tab. 2.4: Informace o útoku DNS RST flood.

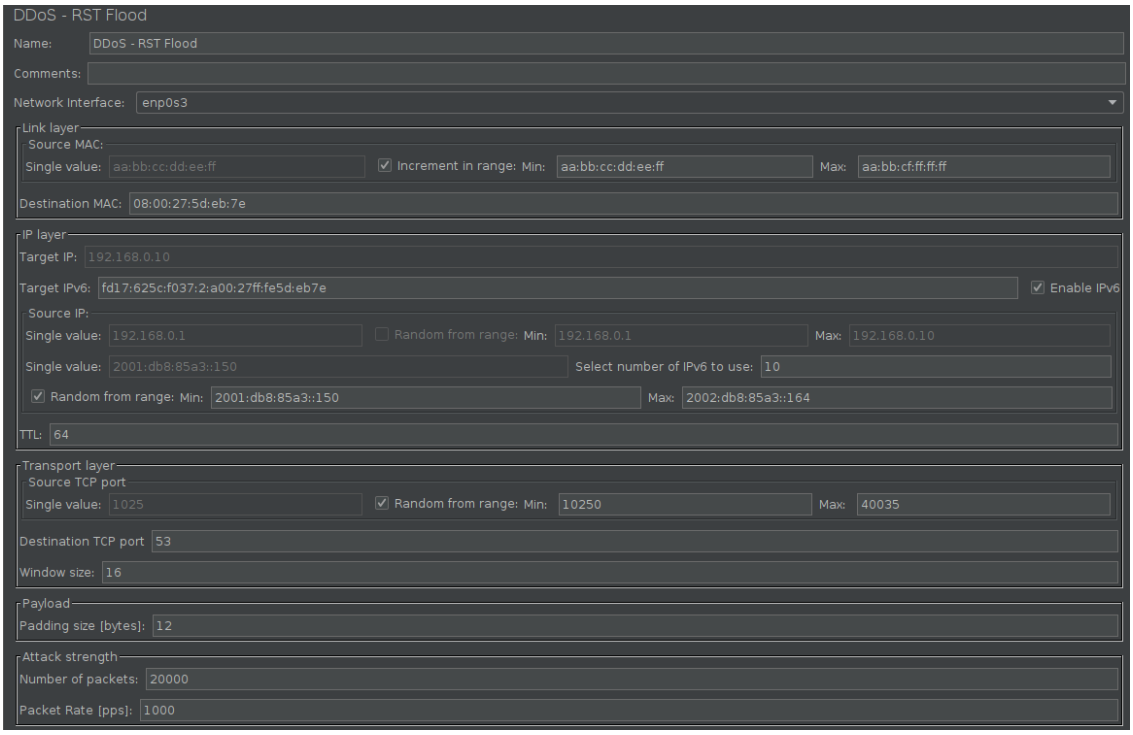
Počet paketů	Rychlost posílání paketů
20 000 paketů	1000 pps

Obr. 2.19: Zatížení RAM před útokem DNS RST flood.

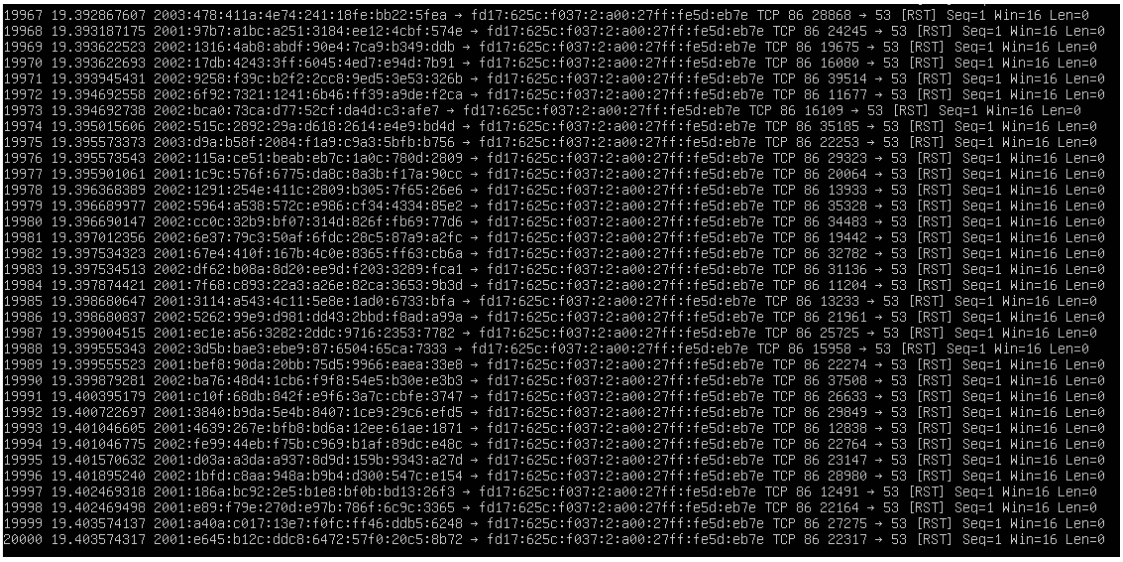
Realizace čtvrtého scénáře

Nejprve se odstraní dříve přidané testy, případně se vytvoří nové vlákno v Apache JMeter, následně se přidá modul *DNS RST Flood* a vyplní se potřebné hodnoty viz obr. 2.20. V tomto testu byla zvolena varianta pro IPv6 adresy.

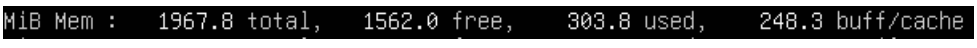
Příchozí RST pakety jsou zobrazeny na obr. 2.21. Zatížení procesoru se pro všechna jádra v průměru pohybuje kolem 16 %, z obr. 2.23. Z obr. 2.22 lze vyčíst, že volná paměť RAM se nijak signifikantně nezměnila.



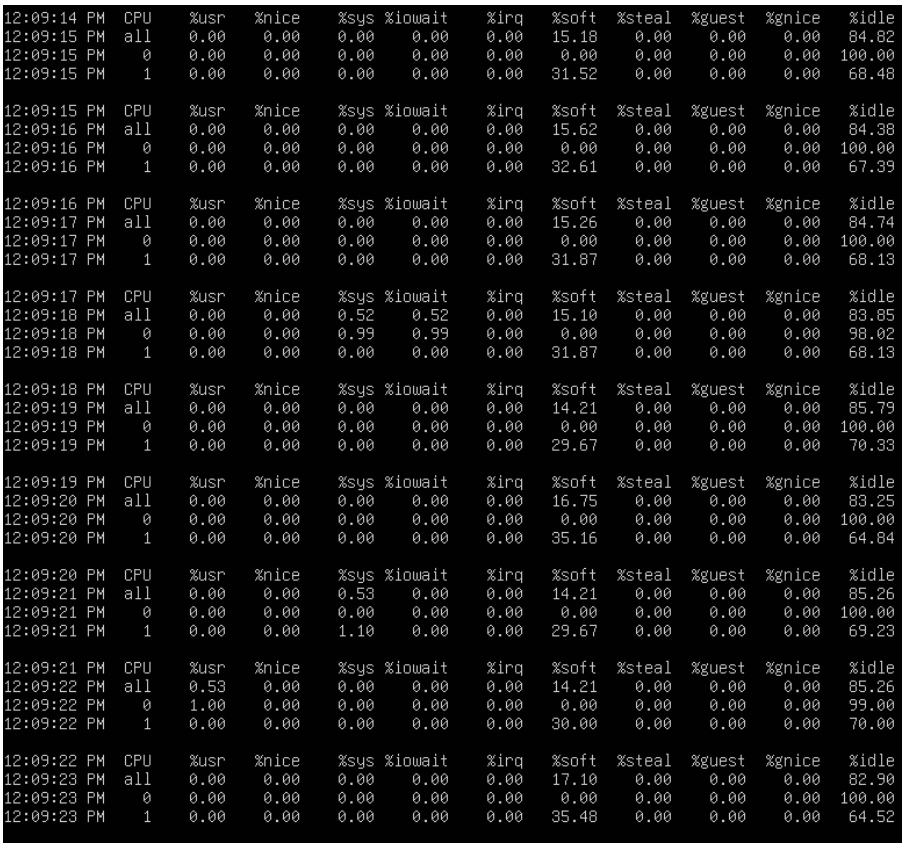
Obr. 2.20: Apache JMeter rozhraní připraveno pro realizaci čtvrtého scénáře.



Obr. 2.21: Příchozí RST pakety na server.



Obr. 2.22: Zatížení RAM po útoku DNS RST flood.



```
12:09:14 PM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
12:09:15 PM  all    0.00    0.00    0.00    0.00    0.00   15.18    0.00    0.00    0.00   84.82
12:09:15 PM    0    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
12:09:15 PM    1    0.00    0.00    0.00    0.00    0.00   31.52    0.00    0.00    0.00   68.48

12:09:15 PM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
12:09:16 PM  all    0.00    0.00    0.00    0.00    0.00   15.62    0.00    0.00    0.00   84.38
12:09:16 PM    0    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
12:09:16 PM    1    0.00    0.00    0.00    0.00    0.00   32.61    0.00    0.00    0.00   67.39

12:09:16 PM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
12:09:17 PM  all    0.00    0.00    0.00    0.00    0.00   15.26    0.00    0.00    0.00   84.74
12:09:17 PM    0    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
12:09:17 PM    1    0.00    0.00    0.00    0.00    0.00   31.87    0.00    0.00    0.00   68.13

12:09:17 PM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
12:09:18 PM  all    0.00    0.00    0.52    0.52    0.00   15.10    0.00    0.00    0.00   83.85
12:09:18 PM    0    0.00    0.00    0.99    0.99    0.00    0.00    0.00    0.00    0.00   98.02
12:09:18 PM    1    0.00    0.00    0.00    0.00    0.00   31.87    0.00    0.00    0.00   68.13

12:09:18 PM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
12:09:19 PM  all    0.00    0.00    0.00    0.00    0.00   14.21    0.00    0.00    0.00   85.79
12:09:19 PM    0    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
12:09:19 PM    1    0.00    0.00    0.00    0.00    0.00   29.67    0.00    0.00    0.00   70.33

12:09:19 PM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
12:09:20 PM  all    0.00    0.00    0.00    0.00    0.00   16.75    0.00    0.00    0.00   83.25
12:09:20 PM    0    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
12:09:20 PM    1    0.00    0.00    0.00    0.00    0.00   35.16    0.00    0.00    0.00   64.84

12:09:20 PM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
12:09:21 PM  all    0.00    0.00    0.53    0.00    0.00   14.21    0.00    0.00    0.00   85.26
12:09:21 PM    0    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
12:09:21 PM    1    0.00    0.00    1.10    0.00    0.00   29.67    0.00    0.00    0.00   69.23

12:09:21 PM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
12:09:22 PM  all    0.53    0.00    0.00    0.00    0.00   14.21    0.00    0.00    0.00   85.26
12:09:22 PM    0    1.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00   99.00
12:09:22 PM    1    0.00    0.00    0.00    0.00    0.00   30.00    0.00    0.00    0.00   70.00

12:09:22 PM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
12:09:23 PM  all    0.00    0.00    0.00    0.00    0.00   17.10    0.00    0.00    0.00   82.90
12:09:23 PM    0    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
12:09:23 PM    1    0.00    0.00    0.00    0.00    0.00   35.48    0.00    0.00    0.00   64.52
```

Obr. 2.23: Zatížení CPU na serveru při DNS RST flood.

2.3.5 Srovnání testovaných scénářů

Testované scénáře pouze zrcadlí DDoS útoky, které nastávají v realitě, jelikož zde nastává limitace použitých zařízení, jak výkonnostní tak i početná.

Ze všech čtyř scénářů byl nejvíce účinný první testovací scénář DNS query flood. Počet posílaných paketů byl nastaven na 10 000 paketů při rychlosti 1000 pps. Volná paměť RAM se před útokem pohybovala okolo 1586 MB, po útoku 214 MB. Zatížení CPU dosáhlo průměrně na 30 % s mírnými odchylkami.

Počet posílání paketů u dalších tří scénářů bylo nastaveno na 20 000 při rychlosti 1000pps. Všechny tři testy jsou zaměřeny na výpočetní zahlcení DNS serveru, tudíž na paměť RAM to nemělo znatelný význam. Zatížení CPU u testu s ACK příznakem dosáhl průměrně na 7 %, u FIN 16 % a u RST též 16 %.

Závěr

Bakalářská práce se zabývala implementací záplavových útoků vytvořených během semestrální práce do programu Apache JMeter. Jedná se o rozšíření pro již existující DDoS modul.

Byly vytvořeny 4 různé útoky, pro verzi s IPv4 i IPv6, a ke každému útoku samostatný modul. Ke každému modulu bylo naprogramováno GUI a tzv. Java Sampler. Jednotlivých 8 konfiguračních souborů bylo přizpůsobeno, aby byly kompatibilní a zachovaly jednotu. Pro všechny útoky je využit jako generátor provozu open-source nástroj trafgen. Útoky byly otestovány a byla zaznamenána jejich účinnost.

Úvodní kapitola se věnovala představení protokolů a hlavně struktuře jejich hlaviček, které bylo potřeba znát pro sestavení konfiguračních souborů a pochopení samotných útoků. Následně se představily nástroje a programy, které jsou klíčové pro zaznamenávání informací spojených s otestováním.

Praktická část se zaměřuje na vytvoření samotných modulů a následnou realizaci útoků. Vysvětluje, jak se vytvořený modul přidá do programu Apache JMeter a jak se spustí útok. Vytvořené útoky jsou DNS query flood, ACK flood, FIN flood a RST flood.

Mnoho DDoS útoků na DNS servery v současnosti využívají mechanismy DNS Amplification nebo Reflection, jelikož mnohonásobně zvětšují sílu útoku. Samotný DNS Reflection, by bylo možné implementovat jako rozšíření Ddos modulu. Útok by vyžadoval schopnost zadat podvrženou zdrojovou IP adresu a dále nastavení IP adres veřejných DNS serverů. V případě útoku na DNS server by byla zvolena IP adresa DNS serveru jako zdrojová.

Literatura

[1] BLANK, Andrew G. *TCP/IP Foundations*. 1. vydání. San Francisco: Sybex, 2004. ISBN 0-7821-4370-9.

[2] POSTEL, J. *Internet Protocol*. RFC 791. RFC Editor. 1981. Dostupné z: `https://www.rfc-editor.org/rfc/rfc791.txt`. [cit. 2024-11-27].

[3] Dr. DEERING, Steve E., HINDEN, Bob. *Internet Protocol, Version 6 (IPv6) Specification*. RFC 8200. RFC Editor. 2017. Dostupné z: `https://www.rfc-editor.org/rfc/rfc8200.txt`. [cit. 2024-11-27].

[4] POSTEL, J. *Transmission Control Protocol*. RFC 793. RFC Editor. 1981. Dostupné z: `https://www.rfc-editor.org/rfc/rfc793.txt`. [cit. 2024-11-27].

[5] POSTEL, J. *User Datagram Protocol*. RFC 768. RFC Editor. 1980. Dostupné z: `https://www.rfc-editor.org/rfc/rfc768.txt`. [cit. 2024-11-27].

[6] KRČMÁŘ, Petr. *Proč není NAT totéž co firewall*. Online. ROOT.CZ. 2007. Dostupné z: `https://www.root.cz/clanky/proc-neni-nat-totez-co-firewall/`. [cit. 2024-11-27].

[7] MIRKOVIC, Jelena, DIETRICH, Sven, DITTRICH, David. *Internet Denial of Service: Attack and Defense Mechanisms*. 1. vydání. London: Pearson, 2004. ISBN 978-0131475731.

[8] ESET SOFTWARE SPOL. S R.O. *Botnet*. Online. ESET SOFTWARE SPOL. S.R.O. ESET. C1992–2024. Dostupné z: `https://www.eset.com/cz/botnet/`. [cit. 2024-11-27].

[9] LEI, Fang. *A Comprehensive Analysis of DDoS attacks based on DNS*. Journal of Physics: Conference Series. September 2021, vol. 2024, no. 1, s. 1-8. ISSN 1742-6596.

[10] Netscout Systems, Inc. *What is a State-Exhaustion DDoS Attack?*. Online. NETSCOUT. C2025. Dostupné z: `https://www.netscout.com/what-is-ddos/state-exhaustion-attacks`. [cit. 2025-05-26].

[11] CZ.NIC. *O doménách a DNS*. Online. CZ.NIC. C2024. Dostupné z: `https://www.nic.cz/page/312/o-domenach-a-dns/`. [cit. 2024-11-27].

[12] CZ.NIC. *Jak funguje DNSSEC*. Online. CZ.NIC. C2024. Dostupné z: `https://www.nic.cz/page/444/jak-funguje-dnssec/`. [cit. 2024-11-27].

[13] GUPTA, Dr. B. B. *An Introduction to DDoS Attacks and Defense Mechanisms: An Analyst's Handbook*. Germany: Academic Publishing, 2011. ISBN 978-3-8465-9569-5.

[14] BORKMANN, Daniel. *Trafgen(8) — Linux manual page*. Online. Michael Kerrisk man7.org. 3 March 2013. Dostupné z: `https://man7.org/linux/man-pages/man8/trafgen.8.html`. [cit. 2024-11-27].

[15] *Netsniff-ng toolkit*. Online. Dostupné z: `http://netsniff-ng.org/`. [cit. 2024-11-27].

[16] *Wireshark*. Online. Dostupné z: `https://www.wireshark.org/`. [cit. 2024-11-27].

[17] Linux. *Mpstat(1) — Linux manual page*. Online. Michael Kerrisk man7.org. 3 March 2013. Dostupné z: `https://man7.org/linux/man-pages/man1/mpstat.1.html`. [cit. 2024-11-27].

[18] *Bind 9*. Online. Dostupné z: `https://bind9.net/`. [cit. 2024-11-27].

[19] *Apache JMeter*. Online. Dostupné z: `https://jmeter.apache.org/`. [cit. 2025-04-27].

[20] ict-tester. jmeter-ddos-plugin. Online. Dostupné po přihlášení z: `https://gitlab.utko.feec.vutbr.cz/ict-tester/jmeter-plugins/jmeter-ddos-plugin`. [cit. 2025-05-26].

Seznam symbolů a zkratek

TCP/IP	Transmission Control Protocol/Internet Protocol – Protokol řízení přenosu/Internetový protokol
HTTP	Hypertext Transfer Protocol – Hypertextový přenosový protokol
FTP	File Transfer Protocol – Protokol pro přenos souborů
DNS	Domain Name System – Systém doménových jmen
TTL	Time To Live – Doba života
TCP	Transmission Control Protocol – Protokol řízení přenosu
SYN	Synchronization flag – Synchronizační příznak
FIN	Finish flag – Koncový příznak
RST	Reset flag – Resetovací příznak
PSH	Push flag – Příznak push
ACK	Acknowledgment flag – Potvrzovací příznak
URG	Urgent flag – Příznak urgentní zprávy
FIN	Finish flag – Ukončovací příznak
RST	Reset flag – Příznak resetu
IP	Internet Protocol – Internetový protokol
UDP	User Datagram Protocol – Uživatelský datagramový protokol
IPV4	Internet Protocol version 4 – Internetový protokol verze 4
IPV6	Internet Protocol version 6 – Internetový protokol verze 6
DOS	Denial of service – Odepření služby
DDOS	Distributed Denial of service – Distribuované odepření služby
NAT	Network Address Translation – Překlad síťových adres
TLD	Top Level Domain – Doména nejvyšší úrovně
RR	Resource Records – Zdrojové záznamy

DNSSEC Domain Name System Security Extensions – Rozšíření bezpečnosti systému doménových jmen

PRSD Pseudorandom Subdomain – Pseudonáhodná subdoména

PTR Pointer – Ukazatel

NXDOMAIN Non-existent domain – Neexistující doména

CPU Central Processing Unit – Centrální procesorová jednotka

GUI Graphical User Interface – Grafické uživatelské rozhraní

MAC Medium Access Control Address – MAC adresa

A Obsah elektronické přílohy

Příloha je v podobě .zip souboru a obsahuje všechny níže vypsané soubory.

- `/` ..... kořenový adresář přiloženého archivu
  - `java` ..... adresář s java třídami
    - `ackflood`
      - `AckFloodGui.java`
      - `AckFloodSampler.java`
    - `dnsqueryflood`
      - `DnsQueryFloodGui.java`
      - `DnsQueryFloodSampler.java`
    - `finflood`
      - `FinFloodGui.java`
      - `FinFloodSampler.java`
    - `rstflood`
      - `RstFloodGui.java`
      - `RstFloodSampler.java`
  - `trafgen_cfg` ..... adresář s konfiguračními soubory trafgen
    - `AbstractAckFlood.cfg`
    - `AbstractAckFloodIpv6.cfg`
    - `AbstractDnsQueryFlood.cfg`
    - `AbstractDnsQueryFloodIpv6.cfg`
    - `AbstractFinFlood.cfg`
    - `AbstractFinFloodIpv6.cfg`
    - `AbstractRstFlood.cfg`
    - `AbstractRstFloodIpv6.cfg`
  - `Ddos.jar` ..... modul Apache JMeter