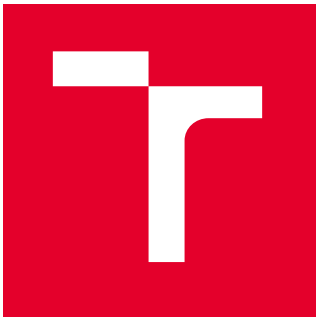


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

TESTER ICT

ICT TESTER

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE — **Bc. Martin Tatar**

AUTHOR

VEDOUCÍ PRÁCE — **doc. Ing. Václav Zeman, Ph.D.**

SUPERVISOR

BRNO 2023

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní program **Informační bezpečnost**

Ústav telekomunikací

Student: Bc. Martin Tatar ***ID:*** 211815

Ročník: 2 ***Akademický rok:*** 2022/23

NÁZEV TÉMATU:

Tester ICT

POKYNY PRO VYPRACOVÁNÍ:

Tester ICT je komplexní systém pro testování kyberbezpečnosti ICT v oblastech funkčního testování, zátěžového testování a odolnosti vůči různým typům DoS útoků. Cílem práce je navrhnout testovací scénáře pro jednotlivé typy testů, které lze pomocí testeru ICT realizovat, vytvořit uživatelský manuál testeru včetně modelových případů testování a doplnit tester o možnost digitálně podepisovat generované výstupy.

DOPORUČENÁ LITERATURA:

[1] HALILI, Emily H. Apache JMeter: A practical beginner's guide to automated testing and performance measurement for your websites. Packt Publishing Ltd, 2008.

[2] ERINLE, Bayo. Performance Testing with JMeter 2.9. Packt Publishing Ltd, 2013.

Termín zadání: 6.2.2023 ***Termín odevzdání:*** 19.5.2023

Vedoucí práce: doc. Ing. Václav Zeman, Ph.D.

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

ABSTRAKT

Diplomová práce se zabývá rozšířením zásuvného modulu Testeru ICT o možnost digitálně podepisovat generované výstupy a testováním vybraných DoS útoků, které jsou v něm možné realizovat. V teoretické části jsou rozebrány vybrané typy DoS útoků, jak fungují, jak je možné se proti nim bránit, příklady významných útoků, které skutečně proběhly a je popsán Apache JMeter. V praktické části jsou vytvořeny a otestovány vybrané útoky. Poté je implementována nová funkcionalita pro digitální podpis.

KLÍČOVÁ SLOVA

zátěžové testování, Apache JMeter, DoS, modul Report generator, PDF, digitální podpis, Java

ABSTRACT

The diploma thesis deals with the extension of the ICT Tester plugin module which allows the user to digitally sign the generated outputs and with the testing of selected DoS attacks that are implemented in it. In the theoretical part of the thesis, selected types of DoS attacks are analyzed, it is described how they work, how to defend against them, examples of significant attacks that actually took place are included and Apache JMeter is described. In the practical part selected attacks are created and tested. Then the new funcionality for digital signature is implemented.

KEYWORDS

load testing, Apache JMeter, DoS, Report generator module, PDF, digital signature, Java

TATAR, Martin. *Tester ICT*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2023, 74 s. Diplomová práce. Vedoucí práce: doc. Ing. Václav Zeman, Ph.D.

Prohlášení autora o původnosti díla

Jméno a příjmení autora:	Bc. Martin Tatar
VUT ID autora:	211815
Typ práce:	Diplomová práce
Akademický rok:	2022/23
Téma závěrečné práce:	Tester ICT

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 diplomové práce panu doc. Ing. Václavu Zemanovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Obsah

Seznam obrázků

Seznam tabulek

Úvod

Během tohoto století došlo k velkému rozvoji informačních technologií. Mimo technologický pokrok se dá hovořit i o rozšíření mezi čím dál větší část lidské populace díky klesajícím nákladům na pořízení.

Technologie dnes využívají firmy po celém světě k zefektivnění výroby, zvýšení dosahu prodeje svých produktů a obecně k automatizaci a dosažení vyšších zisků. Na druhé straně jsou lidé ve svém osobním životě, kteří si na nový komfort také zvykli. Dnes se kvůli nákupu nemusí jezdit do obchodního centra, ale vše se dá pořídit na pár kliknutí ať už na počítači, či například chytrém telefonu. Tato situace je oboustranně výhodná, firma prodá své zboží a zákazník je spokojený, protože ušetřil čas.

Toto je jen jeden z mnoha příkladů. Lidé jsou zvyklí na instantní zprávy svým přátelům, využívání navigace v telefonu na cestu kamkoliv, kde ještě nebyli, čtení nejnovějších zpráv na internetových portálech, či sledování multimediálního obsahu online.

Pokud by tyto služby byly dočasně nedostupné, tak dojde k nepříjemnostem na straně koncových uživatelů. Na straně poskytovatelů služeb však dochází ke škodám finančním, které se v závislosti na velikosti firmy mohou vyšplhat k závratným částkám. V dnešní době se podstatná část ekonomických aktivit odehrává právě na internetu.

Mimo již uvedené příklady se však informační systémy využívají i v kritičtějších odvětvích jako je zdravotnictví, energetika, či finanční sektor. Výpadky v těchto případech mají ještě vážnější následky.

K nedostupnosti může dojít z různých důvodů. Ať je to selhání zařízení, lidská chyba, nebo z důvodu kybernetických útoků. Právě těmi se tato práce zabývá, konkrétně útoky na odepření služby, které se také označují jako DoS útoky.

Teoretická část se věnuje DoS útokům, upřesňuje rozdíl mezi DoS a DDoS útokem, popisuje několik vybraných typů útoků a v další části se věnuje různým způsobům obrany. K ucelení a přiblížení rozsahu takových útoků jsou přidány i reálné případy. Následně je krátce popsán nástroj Apache JMeter.

Ve druhé kapitole je práce zaměřena na práci s Testerem ICT, který je vyvíjen rozšiřováním nástroje Apache JMeter o nové zásuvné moduly. Nejprve byly navrženy testovací scénáře pro různé typy útoků, které je možné realizovat. Ty byly podrobně popsány a v rámci práce byl vytvořen také uživatelský manuál.

Dále byly vytvořené scénáře testovány a byly popsány jejich výstupy. Podrobný report jednotlivých útoků je v příloze.

Poslední část kapitoly se věnuje rozšíření zásuvného modulu o možnost digitálně podepisovat generované výstupy. Ty jsou generovány ve formě HTML stránky. Nej-

prve je řešena konverze do formátu PDF, následně je implementován digitální podpis a upraveno grafické uživatelské rozhraní.

1 DoS útoky

Útok na odepření služby, u kterého se také často využívá jeho anglický název *denial-of-service* (DoS), je kybernetický útok, který má za cíl vyřadit z provozu zařízení, či síť a tím znemožnit legitimním uživatelům přístup.

Útočník se tohoto snaží dosáhnout zahlcením svého cíle nadměrným množstvím provozu, který vyčerpá hardwarové, nebo softwarové prostředky oběti, tak aby oběť nebyla schopna dále reagovat na požadavky reálných uživatelů, nebo využitím špatné, či zastaralé konfigurace, kde využije známé zranitelnosti, která povede k pádu systému.

Motivace k útoku může být různá, snaha způsobení škody konkurenci a získání obchodní výhody, vydírání, snaha způsobit škodu státu, nebo například aktivismus. V závislosti na oběti se velikost dopadu může značně různit. Útoky na odepření služby zpravidla nemají za cíl získat citlivá data, nebo trvale poškodit aktiva oběti. Poškozeného ve výsledku ale mohou stát mnoho času a peněz. Častými oběťmi DoS útoků tak jsou webové stránky ministerstev, banky, online obchody nebo zpravodajské portály. [1][2][3]

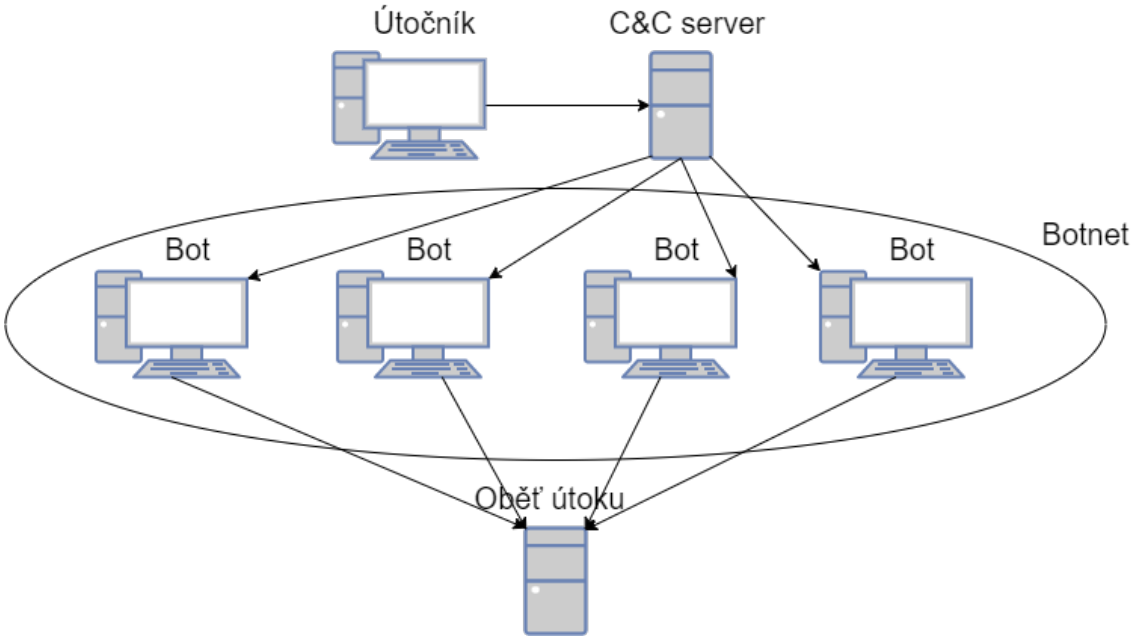
1.1 DDoS útok

Pokud je útok prováděn z více zařízení různě rozmístěných v síti, tak se nazývá *distributed denial-of-service (DDoS)*. K dosažení tohoto většího počtu zařízení využívá útočník takzvaný *botnet*. Pomocí malwaru infikuje cizí stanice, které se potom na útoku podílejí, aniž by to majitel zamýšlel nebo o tom věděl. Takto kompromitované zařízení se nazývá *bot* nebo také *zombie*.

Ke kontrole botnetu útočníkovi slouží *command and control server*. Pomocí něj dává botům příkazy k útoku na oběť. Jak toto funguje je v menším měřítku zobrazeno na obrázku 1.1.

Díky využití celého botnetu, který může běžně čítat tisíce i více zařízení, se útočník snaží vyčerpat omezené zdroje oběti. Zpravidla se snaží vyčerpat konečný počet požadavků, který je server oběti schopen vyřídit zároveň, nebo šířku pásma kterou využívá k připojení k internetu. Pokud se toto útočníkovi podaří překonat, tak dojde k výraznému zpomalení odbavování požadavků, nebo k naprosté nedostupnosti služby.

Rozeznat provoz od útočníků a provoz od běžných uživatelů je složité, protože boti jsou v důsledku legitimní zařízení v síti. Stejně tak je těžké odhalit strůjce útoku, jelikož následný provoz botnetu přichází od jednotlivých zařízení s jejich IP adresami.[5][6]



Obr. 1.1: DDoS útok

1.2 Typy DoS útoků

DoS útoky se dají rozdělit na 3 základní typy:

- vyčerpání omezených zdrojů
- poškození konfiguračních informací
- fyzické poškození síťových komponent

Vyčerpání omezených zdrojů

Tento typ útoku je z výše zmíněných nejčastěji využívaný. Útoky mají za cíl vyčerpat síťové, nebo systémové zdroje. U systémových zdrojů se jedná o snahu přetížit procesor, nebo paměť RAM. U síťových zdrojů se útočník snaží spotřebovat dostupnou šířku pásma oběti vygenerováním velkého provozu. V obou případech jde o to, aby oběť byla zaměstnána s požadavky útočníka a nebyla schopna odbavovat požadavky legitimních uživatelů.

Poškození konfiguračních informací

Špatná konfigurace může vést ke špatnému výkonu, nebo dokonce k nepoužitelnosti zařízení. Příkladem může být změna směrovacích informací, která povede k nedostupnosti sítě.

Fyzické poškození síťových komponent

Kromě samotných útoků vedených z jiných zařízení je důležité bránit se i před fyzickým poškozením. Fyzickou bezpečnost je potřeba zařídit tak, aby nedošlo k neautorizovaným přístupům k serverům, počítačům, směrovačům, napájení a dalším důležitým prvkům. Fyzická bezpečnost se nevztahuje pouze k DoS útokům, ale i k mnoha dalším. Je tedy potřeba zahrnout všude, kde by bez ní mohlo dojít k nežádoucím následkům.[4]

1.2.1 SYN Flood

Útok SYN flood k dosažení odepření služeb využívá protokol TCP. TCP je spolehlivý, spojově orientovaný prokol, který před započetím komunikace ustanoví spojení mezi komunikujícími stranami pomocí *three-way handshake*. Tento mechanismus sestává z těchto tří kroků:

1. Na začátku klient zašle serveru SYN paket, aby inicializoval začátek sestavení spojení.
2. Na tuto zprávu server odpoví zasláním SYN/ACK, čímž potvrzuje přijetí zprávy od klienta a souhlas se zahájením spojení.
3. V posledním kroku zasílá klient ACK zpět serveru, aby došlo k potvrzení předchozího kroku. Tímto je *three-way handshake* dokončen a spojení je otevřeno a připraveno k posílání dat.

Při útoku však dochází pouze k prvním dvěma krokům. Útočník zahájí komunikaci zasláním SYN paketu a server odpoví SYN/ACK, poté ho však útočník nechá čekat na poslední ACK zprávu, která nikdy nepřijde. Rozdíl mezi *three-way handshake* a útokem je zobrazen na obrázku 1.2

Tím vzniká polootevřené spojení, na které si server musí vyhradit prostředky a stanovit na jakém portu bude komunikace probíhat. Každé takové polootevřené spojení blokuje kapacitu serveru, dokud nevyprší časový limit k jeho dokončení a nezavře se.

Útočník se tedy snaží zahájit tolik spojení, aby vyčerpal kapacitu oběti a zablokoval tak přístup legitimním uživatelům.

SYN flood útok se dá realizovat třemi způsoby.

Přímý útok

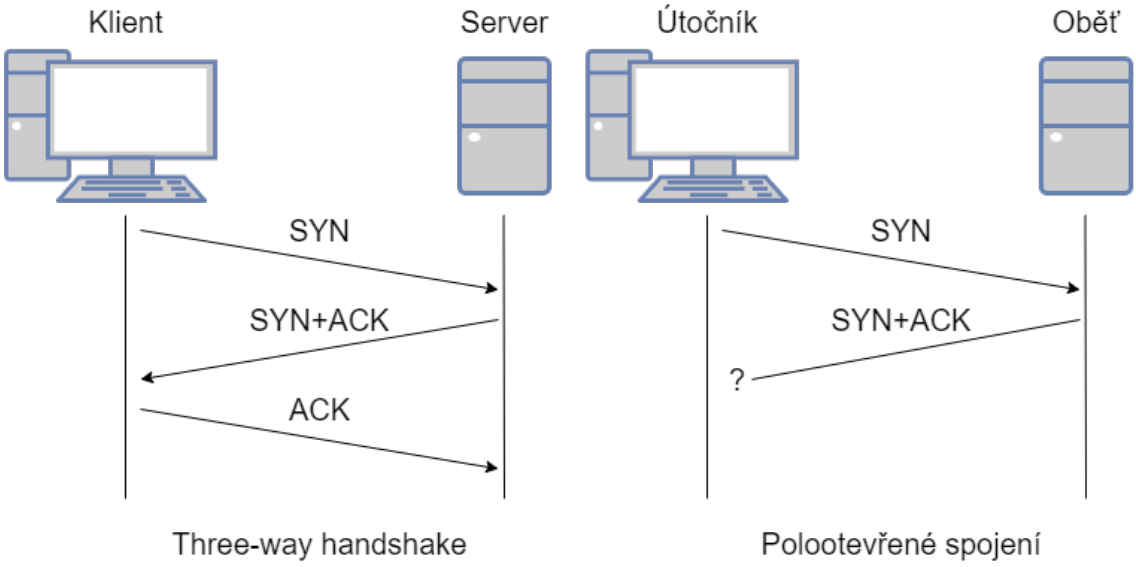
V tomto případě se útočník nesnaží podvrhnout svou IP adresu a útočí tedy přímo z jednoho zařízení. K realizaci lze využít nastavení firewallu, tak aby nepustil příchozí SYN/ACK pakety, nebo neodesílal jiné než SYN. V důsledku využití jediné IP adresy je tento útok snadno řešitelný jejím zablokováním.

Podvržený útok

Zde útočník využívá podvržené IP adresy, jednotlivá spojení se tedy zdají být od různých uživatelů, zatímco přicházejí stále od jediného útočníka. Díky tomu je útok těžší nejen detekovat, ale i vystopovat.

DDoS SYN Flood

SYN flood lze provést i jako distribuovaný útok, který byl popsán v předchozí části. Při útoku je tedy využit botnet, který generuje provoz a otevírá nová spojení u oběti. Zde lze využít jak přímý útok, tak útok s podvrženými IP adresami. V tomto případě je vystopování útočníka nejtěžší.[7][8]



Obr. 1.2: Three-way handshake a polootevřené spojení u SYN Flood útoku

1.2.2 UDP Flood

User Datagram Protocol (UDP) je nespojově orientovaný protokol, který nevyžaduje žádný mechanismus před započetím komunikace. Vyznačuje se svou jednoduchostí, což přispívá k rychlosti tohoto protokolu. Nezaručuje doručení, ani správné pořadí, přesto má své využití, zejména u aplikací běžících v reálném čase jako je například *VoIP*.

Kromě tohoto legitimního využití se však dá zneužít i k DoS útoku, při kterém je oběť zahlcena UDP pakety, tak aby nebyla schopna reagovat na další provoz od běžných uživatelů.

UDP flood zneužívá běžné chování serveru při přijetí UDP paketu. Každé přijetí paketu serverem znamená využití jeho zdrojů k obsluze, která sestává ze dvou následujících kroků:

1. Server zkontroluje, zda v daný okamžik na konkrétním portu naslouchá některý program.
2. V případě, že na daném portu aktuálně neprobíhá příjem žádných paketů, odešle server ICMP zprávu *destination unreachable*, aby o tom informoval odesílatele.

Samotný útok spočívá v odesílání velkého objemu UDP paketů na různé porty cílené oběti. K tomu útočník zpravidla využije podvrženou IP adresu, aby nebyl lehce odhalen a aby mu nepřicházely odpovědi. Útok je také zobrazen na obrázku 1.3

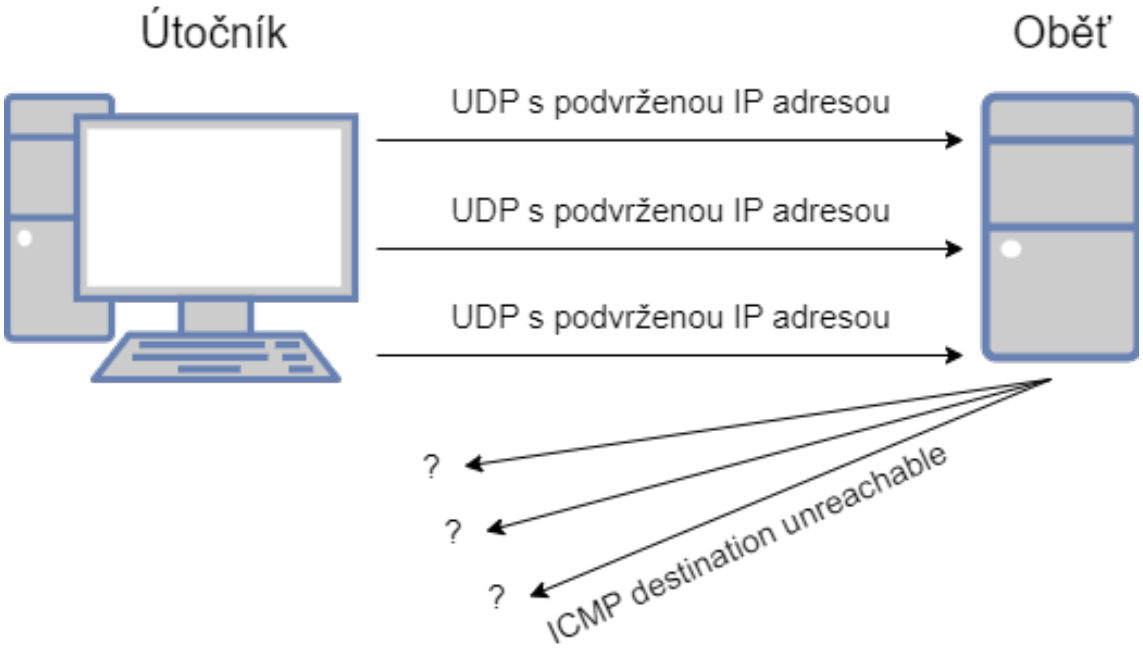
V důsledku toho, že server musí při každém přijatém paketu zkontrolovat, zda je port dostupný a teprve poté odpovědět, může při velkém náporu nových požadavků dojít k přetížení serveru a jeho následné nedostupnosti.

UDP flood útok je možno provést také jako distribuovaný útok (DDoS), při kterém je využito více zdrojů útoku ze vzdáleně ovládaného botnetu, který může čítat i tisíce infikovaných počítačů, což zvyšuje dostupnou kapacitu útočníka k navýšení provozu.

To může vést kromě zamýšleného vyčerpání kapacity serveru i k překonání dostupné šířky pásma oběti. Takto vznikne úzké hrdlo, což dále zhoršuje dopad útoku a činí ho nebezpečnějším.

Intenzita útoku se dá vyjádřit objemem datového provozu v bitech za sekundu, nebo paketech za sekundu.

Jednou z metod obrany proti UDP flood útoku je omezení počtu odesílaných ICMP zpráv. To však může mít i negativní účinek, jelikož se nebude rozlišovat mezi provozem od útočníka a legitimního uživatele, takže se může stát, že se odpověď nedostane právě k legitimnímu uživateli.[9][10][11]



Obr. 1.3: UDP Flood útok

1.2.3 HTTP Flood

HTTP flood je útok probíhající na aplikační vrstvě. *Hypertext Transfer Protocol (HTTP)* je internetový protokol na aplikační vrstvě sloužící ke komunikaci s *world wide web (WWW)* servery. Slouží k načítání webových stránek v internetovém prohlížeči, přenosu hypertextových dokumentů v různých formátech a je jedním z nejvyužívanějších protokolů na internetu.

Útok HTTP flood zpravidla probíhá jako *distributed denial-of-service (DDoS)*, tudíž útočník využívá infikovaný botnet, aby mohl využít větší počet útočících zařízení.

HTTP Flood má 2 varianty, podle toho který požadavek tohoto protokolu zneužívá:

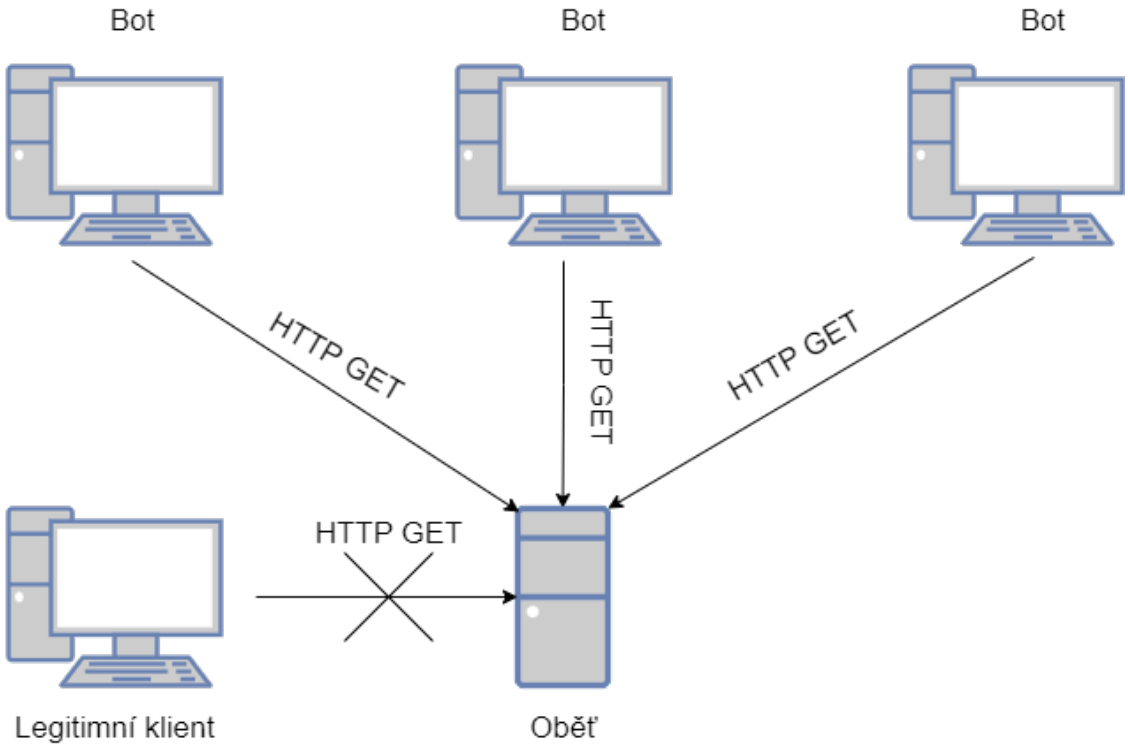
1. HTTP GET - v této variantě útoku se využívá požadavek GET na server, který slouží ke stažení souboru, obrázku, či jiných dat. GET požadavky jsou jednodušší na odbavení, ale i na tvorbu útoku a snadněji se využívají spolu s botnetem. Útok je zobrazen na obrázku 1.4.
2. HTTP POST - POST požadavek je náročnější na odbavení a spotřebovává více zdrojů serveru. Slouží například ke zpracování odeslaného formuláře. V takovém případě je potřeba nahrát daná data do databáze. Útok zneužívá toho, že zpracování odeslaných dat a obsluha databáze je výpočetně výrazně náročnější, než samotné odeslání POST požadavku, což snižuje nároky na zdroje

útočníka.

Jelikož jsou při útoku využívány běžné URL požadavky, je velmi složité rozlišit mezi běžným provozem a tím, který přichází od útočníka. Jedním ze způsobů, jak se proti tomuto útoku bránit je přidání výpočetní výzvy pomocí JavaScriptu.

V dnešní době je již naprostá většina internetového provozu šifrována. Místo HTTP se tedy používá jeho zabezpečená verze HTTPS, která je vylepšena o využití *Transport Layer Security (TLS)*.

To však pouze přidává na síle útoku, jelikož jsou šifrované zprávy složitější na zpracování a využívají více zdrojů serveru. Stejně tak ztěžují obranu před tímto útokem, jelikož obranné mechanismy nemohou HTTPS požadavky kontrolovat před tím, než se dešifrují.[12][13][14]



Obr. 1.4: HTTP Flood útok

1.2.4 DNS Amplification

Domain Name System (DNS) je neoddělitelnou součástí internetu. Slouží k převodu doménových jmen na IP adresy a opačně. Bez této služby by bylo velmi složité zapamatovat si jednotlivé IP adresy webových stránek.

DNS je decentralizovaný, hierarchický systém, který ke svému fungování využívá jednotlivé DNS servery, které po určitou dobu uchovávají záznamy žádaných adres ve své paměti.

Při DNS Amplification útoku využívá útočník právě DNS servery ke znásobení provozu, tak aby plně zahltil svůj cíl a znemožnil mu ostatní provoz a dostupnost ostatním uživatelům.

DNS Amplification útok se dá rozdělit do čtyř kroků:

1. Útočník použije podvrženou IP adresu a zašle dotaz na DNS server. K tomu se běžně používá UDP protokol. Jako podvržená IP adresa se použije adresa oběti.
2. Cílem dotazu na DNS server je dosažení co největšího objemu odpovědi, aby došlo k zesílení původního provozu od útočníka. K tomu lze použít argument *ANY*, který slouží k získání všech dostupných záznamů serveru pro konkrétní doménu.
3. Po obdržení požadavku ho zpracovává DNS resolver, který na něj reaguje a odesílá odpověď na podvrženou IP adresu.
4. Oběť útoku obdrží nevyžádanou odpověď od DNS serveru, obsahující velké množství záznamů, což zaměstnává jak kapacitu oběti, tak kapacitu síťového připojení a může v důsledku vést k nedostupnosti služeb.

Přestože lze odpovědí od DNS serveru zesílit provoz až 50krát (ve výjimečných případech i vícekrát), DNS dotazy mají obecně malou velikost. Proto se tento útok provádí distribuovaně (DDoS), aby bylo možno dosáhnout dostatečně velkého provozu a skutečně zahltit oběť, tak aby ve výsledku došlo k zamýšlené nedostupnosti služeb.

Vzhledem k tomu, že přicházející odpovědi jsou legitimní data od běžně využívaných serverů, je obrana proti tomuto útoku složitá.[15][16]

1.2.5 NTP Amplification

Network Time Protocol (NTP) je protokol sloužící k synchronizaci času mezi zařízeními komunikujícími v síti. NTP Amplification útok je záplavový útok, který tento protokol zneužívá.

Stejně jako DNS Amplification útok se snaží využít rozdíl nákladů na šířku pásma mezi útočníkem a obětí. Jde tedy o to dostat z malých žádostí co největší odpovědi. Když se toto zkombinuje s botnetem, tak dochází k výraznému zvýšení provozu a odhalení útočníka je mnohonásobně složitější.

Při útoku se využívá příkaz *monlist*, který vrací seznam posledních 600 připojení k serveru. To zajišťuje znatelné znásobení následné odpovědi, která přichází na jednoduchý dotaz. Konkrétněji tento příkaz dokáže původní zprávu od útočníka znásobit přibližně 200krát. To v praxi znamená že s rychlostí připojení pouhých 100 Mb/s lze vygenerovat provoz směrem k oběti o velikosti zhruba 20 Gb/s. K útoku

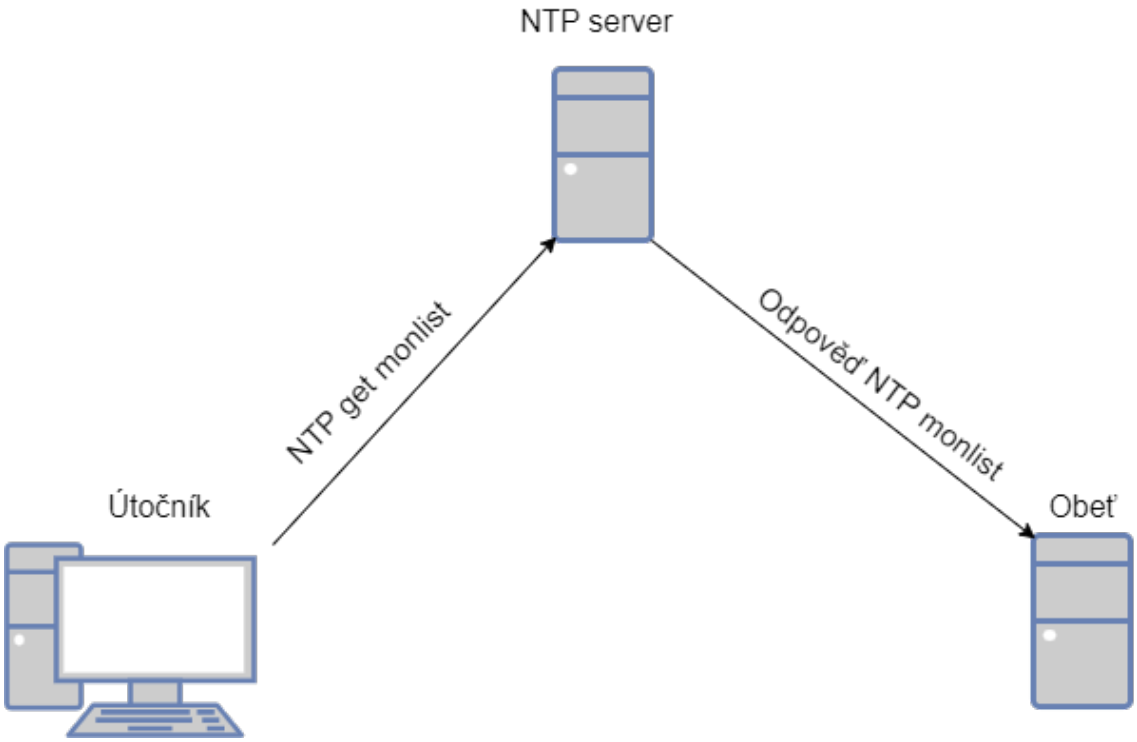
je tedy potřeba vybrat NTP server, který má příkaz *monlist* povolen. Starší servery povolovaly tento příkaz ve svém výchozím nastavení.

NTP Amplification útok je zobrazen na obrázku 1.5 a podobně jako DNS Amplification lze rozdělit do čtyř následujících kroků:

1. V prvním kroku využívá útočník botnet k zasílání UDP paketů na NTP servery, které mají povolen příkaz *monlist*. Při tom podvrhuje IP adresu, tak aby odpovědi byly směrovány na oběť.
2. Příchozí provoz na NTP server obsahuje *monlist*, což způsobuje velkou odpověď od serveru.
3. Server odpovídá na podvrženou adresu, tedy adresu oběti, kam zasílá své záznamy.
4. Oběť je zahlcena velkým provozem, který má za následek překonání jejich kapacit a následuje nedostupnost služeb.

Jelikož je k NTP dotazům využíván protokol UDP, u kterého nedochází k navázání spojení, tak server automaticky odpovídá, aniž by kontroloval zda jsou dotazy autentické. Výsledný provoz tak vypadá jako legitimní, protože přichází od běžně využívaných serverů.

V důsledku tohoto je obrana proti NTP Amplification útoku bez blokování NTP serverů, včetně jejich normálního využití, velmi složitá.[17][18]



Obr. 1.5: NTP Amplification útok

1.2.6 ICMP Flood

ICMP je podpůrný protokol v TCP/IP sítích, díky kterému si zařízení mohou vyměňovat servisní informace jako nedostupnost, nebo nedosažitelnost. ICMP obstarává například ping a traceroute.

Ping se běžně používá ke kontrole kvality, případně dostupnosti, spojení mezi zařízeními.

ICMP Flood se také nazývá „Ping Flood“, jelikož se při něm využívají echo požadavky v jednom směru a echo odpovědi ve směru druhém. Obě tyto zprávy využívají zdroje serveru, který přijímá echo požadavek, zpracovává ho a následné na něj odpovídá pomocí echo odpovědi. Stejně tak je tímto generován provoz který konzumuje šířku pásma.

Princip útoku tedy spočívá v tom, aby byl počet dotazů natolik velký, že dojde k vyčerpání zdrojů oběti a ta nebude dále schopna zpracovávat další požadavky, nebo vytvořit dostatečně velký provoz k zahlcení dostupné šířky pásma oběti, tak aby se k ní nedostávaly žádné jiné dotazy od legitimních uživatelů.

Jelikož zde nedochází k žádnému zesílení provozu, je účinnost tohoto útoku úměrná počtu odeslaných požadavků na oběť. Stejně tak je provoz úměrný velikosti provozu od útočníka. Z tohoto důvodu je útočník motivován k použití co největšího botnetu, tak aby zajistil dostatečné prostředky k vyčerpání zdrojů oběti.

Útočníci dříve používali podvržené IP adresy, aby tím ztížili své vystopování, to už však při použití botnetu není nutné. Jelikož je botnet složen z jednotlivých reálných zařízeních, tak není potřeba skrývat IP adresy jednotlivých botů a zároveň to pomáhá maskovat strůjce útoku.

Výsledný útok se tedy skládá ze dvou jednoduchých kroků, které se neustále opakují:

1. Útočník pomocí mnoha zařízení v botnetu odesílá ICMP echo požadavky na svou oběť.
2. Oběť musí všechny příchozí požadavky zpracovat a poté na ně jednotlivě odpovídat na dané IP adresy pomocí ICMP echo odpovědi.

Tento proces probíhá tak dlouho, dokud nedojde k přetížení oběti, nebo vyčerpání její šířky pásma a útočník nedocílí nedostupnosti služeb.

Nejjednodušší obranou proti tomuto útoku je vypnutí ICMP funkcionality cíleného směrovače, počítače nebo jiného zařízení. Tím dojde k zablokování příchozích dotazů a nebude docházet ani k odesílání odpovědí. V důsledku toho však nebude možné využívat ani žádné další funkcionality ICMP jako je například traceroute, který trasuje cestu od zdroje až k cíli.[19][20]

1.2.7 Slowloris

Slowloris je program vytvořený Robertem Hansenem, který slouží jako nástroj k přetížení cíleného serveru pomocí otevíraní a udržování mnoha spojení mezi útočníkem a daným serverem.

Slowloris ke svému cíli využívá neúplné HTTP požadavky a pracuje tedy na aplikační vrstvě. Princip útoku spočívá v otevírání spojení a poté ve snaze udržet je otevřená co možná nejdéle.

Na rozdíl od většiny DoS útoků se Slowloris nesnaží vytvořit co největší provoz, nebo zahltit oběť co nejvíce požadavky naráz. Místo toho spadá do kategorie „low and slow“ útoků. Tento nástroj umožňuje jedinému útočícímu zařízení vyřadit server s využitím relativně malé kapacity připojení.

Slowloris se snaží pouze o vyčerpání možných současně otevřených spojení, které je server schopen udržovat zároveň. Server samozřejmě otevřená spojení po určitém čase bez aktivity uzavírá, tomu se však Slowloris snaží zabránit posláním dalšího neúplného HTTP požadavku, tak aby předešel tomuto časovému intervalu a spojení zůstalo otevřené.

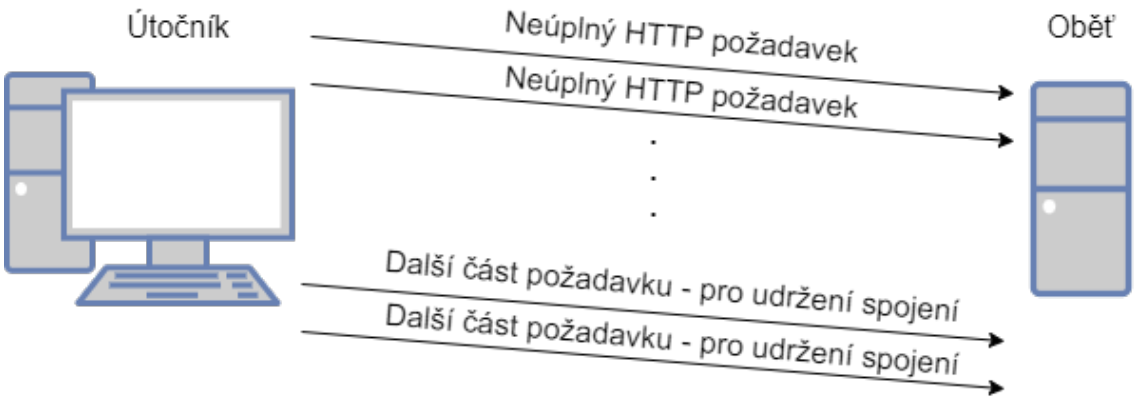
Útok se tedy dá rozdělit na 4 kroky:

1. Útočník otevírá co největší množství spojení pomocí neúplných HTTP požadavků.
2. Aby server každý požadavek mohl obsloužit, otevře pro něj samostatné spojení. Z důvodu omezené kapacity serveru je každé spojení časově omezeno a po dané době se uzavřeno, aby se uvolnilo pro jiné požadavky.
3. Před uzavřením spojení se útočník brání posíláním dalších neúplných zpráv, tak aby se spojení pouze zdálo pomalé.
4. Pokud server není schopen tato spojení ukončovat, dojde nakonec k vyčerpání počtu možných spojení, což bude mít za následek neschopnost přijetí dalších požadavků a dojde k nedostupnosti služeb. Útok je také zobrazen na obrázku 1.6

Díky chytrému řešení útoku je nenáročný nejen na zdroje útočníka, ale v porovnání se záplavovými útoky i na další služby.

Útok je také nenápadný, jelikož se tváří jako běžný provoz. Z tohoto důvodů je jeho detekce složitá.

Vzhledem k tomu, že spojení zůstávají otevřená, tak nedochází ani k žádnému logování provozu, protože to nastává až po odbavení požadavku. Při kontrole logů tedy může být probíhající útok přehlédnut i když je server už nedostupný a nereaguje na další provoz.[21][22]



Obr. 1.6: Princip útoku Slowloris

1.2.8 R.U.D.Y.

„R U Dead Yet?“ je nástroj k DoS útokům, který podobně jako Slowloris patří do kategorie „low and slow“ útoků.

R.U.D.Y. se tedy nesnaží zahltit svou oběť velkým počtem rychlých dotazů, ale zaměřuje se na zdržování serveru velmi pomalým zasíláním dat skrze dostupné formuláře.

Samotný útok se skládá ze čtyř kroků:

1. R.U.D.Y. hledá dostupné pole, kam může zadat vstup.
2. Následně vytváří HTTP POST požadavek, který se tváří jako běžný provoz, ale v hlavičce signalizuje, že následuje objemný obsah.
3. R.U.D.Y. poté využívá náhodné intervaly kolem 10 sekund k zasílání jednotlivých zpráv, které mohou mít pouze jeden byte. Tímto maximálně natahuje proces komunikace a zdržuje server.
4. Spojení zůstává otevřené a R.U.D.Y. dál posílá data po malých dávkách. Díky použití náhodných intervalů se může zdát že data přicházejí pouze od uživatele, který má pomalé připojení.

K ukončení spojení ze strany útočníka však nikdy nedochází. Naopak dochází k otevření mnoha dalších stejných spojení, tak aby se vyčerpala kapacita cíleného serveru a ten následně nebyl schopen reagovat na další požadavky a došlo tedy k nedostupnosti služeb pro legitimní uživatele.

V případě, že se jedná o robustnější server, který je schopen držet mnoho otevřených spojení, lze útok provést také jako distribuovaný (DDoS), čímž se dosáhne většího počtu útočících zařízení a tím pádem i více otevřených spojení, tak aby se zdroje serveru vyčerpaly.[23][24][25]

1.3 Obrana proti DoS útokům

V této části práce bude probráno, jak se úspěšně připravit na obranu proti DoS, případně DDoS útokům spolu s několika vybranými konkrétními technikami.

Co se cloudových poskytovatelů týče, obrana proti distribuovanému útoku se dělí do 4 fází, tento princip však lze aplikovat i obecněji:

1. Detekce - nejprve je potřeba rozpoznat, že k útoku dochází a úspěšně ho rozlišit od zvýšeného legitimního provozu.
2. Reakce - zde je potřeba na útok adekvátně reagovat a zahazovat provoz od botů, zatímco běžní uživatelé budou nedotčeni.
3. Směrování - aby nedošlo k nedostupnosti služeb, je možné rozdělit zbývající provoz na menší části, které budou snazší na obsluhu.
4. Poučení - analýzou lze zjistit opakující se události. Z toho se lze připravit na případné další útoky. Příkladem může být zneužití určitých protokolů, nebo útoky přicházející z konkrétní země.[26]

Mezi obecné rady, které platí pro všechny typy kybernetických útoků patří udržování správné konfigurace všech zařízení, využití firewallu a pravidelné aktualizace softwaru, tak aby byly vyřešeny případné záplaty na objevené zranitelnosti.

Co se týče útoků, které míří na vyčerpání vyčerpání zdrojů oběti, zde se může jevit navýšení těchto zdrojů jako jednoduché řešení.

Navýšení kapacity serveru, větší šířka pásma nebo větší počet síťových zařízení k obsluze zajisté pomůže, avšak tyto kroky se pojí s dalšími finančními náklady a útočník má také možnost rozšířit počet zařízení ve svém botnetu a tím zesílit útok, což by vedlo k situaci, kterou rozhodne kdo nakonec disponuje více dostupnými prostředky.

1.3.1 Směrování do černé díry

Jedním z nejjednodušších řešení obrany proti útoku je směrování provozu do „černé díry“ (anglicky *blackholing*), čímž je provoz ztracen. Při využití spojově orientovaného protokolu TCP je o tom odesílatel informován, při využití UDP však k žádné notifikaci nedochází.

Tento způsob má však mnoho nevýhod. Hlavní nevýhodou je, že bez rozlišování provozu dochází k zahazování jak komunikace od útočníka, tak i od legitimních uživatelů. V takovém případě lze útočníkův cíl považovat za splněný.

Z tohoto důvodu tento typ obrany není běžně samostatně využíván.[27]

1.3.2 Filtrování provozu

Dříve než se škodlivý provoz stihne dostat k oběti, je možné ho odstranit pomocí filtrování provozu.

U větších podniků, jako je například poskytovatel internetového připojení (často používána anglická zkratka ISP), se mnohdy využívá „scrubbing“ centrum, které analyzuje provoz a odděluje ten škodlivý. Toto se nevztahuje pouze na D(D)oS útoky, ale také na další známé slabiny.

Zpravidla pomocí DNS a BGP se při útoku provoz odkloní do scrubbing centra, kde se analyzuje a vyčistí. Zbylý legitimní provoz se poté navrací do sítě a normálně pokračuje k cíli.[28]

1.3.3 SYN Cookies

Tato technika se využívá při obraně proti SYN Flood útoku, který se zaměřuje na vytváření polootevřených spojení, tak aby vyčerpal kapacitu zařízení.

Při této strategii se však nečeká na poslední potvrzení ACK od potenciálního útočníka, ale server vytvoří cookie a po tom co odpoví SYN/ACK zahodí původní SYN od iniciátora spojení.

Tímto se uvolní příslušný port, který je připraven k obsloužení dalšího spojení. Pokud poslední ACK skutečně přijde, tak dojde k rekonstrukci spojení a původní spojení se naváže.[7]

1.3.4 Verifikace zdrojové IP adresy

Zde se jedná o útoky, které podvrhují zdrojovou IP adresu za adresu oběti jako je například DNS Amplification útok.

Zařízení v botnetu podvrhují svou zdrojovou adresu a zaměňují ji za adresu oběti, tak aby následné odpovědi zahltily právě oběť útoku. Jako řešení se nabízí kontrola zdrojové IP adresy poskytovatelem internetového připojení. Pokud pakety přicházejí ze sítě, ale tváří se, že pocházejí odjinud, lze je považovat za podvržené a tudíž zahodit.

Toto řešení by znatelně snížilo efektivitu zesilovaných útoků.[15]

1.3.5 Využití IP reputace

Existuje mnoho nástrojů, které poskytují informace od jednotlivých IP adresách a jejich chování - tedy reputaci. Hlavní účel těchto nástrojů je držet informace o IP adresách se špatnou reputací, tedy o takových, které provádí nežádoucí činnosti.

Na základě toho dostávají skóre, které napovídá o tom, co lze od daného zařízení očekávat.

Díky těmto informacím je možné nastavovat pravidla pro zařízení se špatnou reputací, nebo je přímo blokovat.

IP reputace se nevztahuje jen k obraně proti botnetům a tedy DDoS útokům, ale využívá se například i u emailové komunikace a na základě ní mohou některé adresy končit ve spamu.[29][30]

1.3.6 Komplexní řešení

Existuje velké množství útoků a stejně tak mnoho různých technik obrany. Na trhu je také několik společností, které nabízí komplexní obranu proti D(D)oS útokům.

Tyto služby jsou samozřejmě placené, avšak útoky na odepření služby mohou mít dalekosáhlejší následky, které nemusí být pouze finanční.

1.4 Významné DDoS útoky

DDoS útoky se stávají čím dál častěji a s časem postupně také získávají na intenzitě. V této kapitole bude popsáno několik historicky významných a známých DDoS útoků.

1.4.1 Mafiaboy

Tento útok se stal v roce 2000 a nazývá se podle přezdívky útočníka, který jej provedl.

Tím byl teprve 15letý středoškolák Michael Calce. K útoku využil servery několika univerzit a podařilo se mu vyřadit z provozu několik známých webových stránek jako je eBay, CNN, Dell, nebo Yahoo.[31]

1.4.2 Útok na vládní stránky Estonska

Estonsko se brzy začalo věnovat státní digitalizaci a mimo jiné umožňovalo také online volby.

V roce 2007 však došlo k významnému útoku na vládní služby, finanční instituce a zpravodajské servery.

Spekuluje se, že útok přišel v reakci na přemístění památníku z druhé světové války, což mělo za následek politické napětí s Ruskem. Za útok byl zatčen Estonec s ruskými kořeny a ze zapojení byla podezřelá i ruská vláda. Ta však nepovolila žádné vyšetřování na svém území.[31]

1.4.3 Útok na Spamhaus

Spamhaus je organizace zabývající se bojem proti spamu na internetu - zejména v podobně emailů. To se z povahy činnosti Spamhausu nelíbí těm, které to ovlivňuje a tak v roce 2013 proběhl na Spamhaus útok.

K útoku byly zneužity veřejné nezabezpečené DNS servery a útok dosahoval provozu o velikosti až 300 Gb/s, což byla v té době rekordní velikost. Spamhaus však využíval ochranu od Cloudflare a útočníkům se tak nepodařilo vyřadit servery mimo službu.

V důsledku toho se útočníci začali zaměřovat na poskytovatele připojení pro Cloudflare. Nakonec se útočníkům nepodařilo svého cíle dosáhnout, ale došlo maximálně k lokálním problémům.[32]

1.4.4 Útok na Cloudflare

V roce 2014 proběhl útok na jednoho ze zákazníků společnosti Cloudflare, která poskytuje IT služby, včetně kybernetické bezpečnosti.

Konkrétně se jednalo o NTP Amplification útok, který zneužívá NTP servery a dokáže mnohonásobně zvětšit výsledný provoz. Tento konkrétní útok dosahoval intenzity bezmála 400 Gb/s.

K tomu bylo využito 4529 NTP serverů z 1298 sítí. To znamená, že jeden server průměrně generoval 87 Mb/s.[33]

1.4.5 Útok na GitHub 2018

GitHub je populární platforma pro vývoj softwaru, kterou využívají desítky milionů uživatelů.

To z ní dělá potenciální terč útoků, kterých za svou existenci zažila mnoho. V roce 2018 však došlo k útoku, který dosahoval intenzity až 1,35 Tb/s nebo skoro 127 milionů paketů za vteřinu. Využit k tomu byl memcached útok, který využívá databázový memcache systém k zesílení výsledného provozu.

Útok o takové velikosti GitHub zaskočil a nejprve přetížil, ale jelikož GitHub využívá obranu proti DDoS útokům, tak tento konkrétní nakonec trval pouze 20 minut.[31][34]

1.4.6 Amazon Web Services

V roce 2020 došlo k útoku na jednoho ze zákazníků *Amazon Web Services (AWS)*, který nebyl blíže specifikován.

K útoku byly využity webové servery *Connection-less Lightweight Directory Access Protocol (CLDAP)*, které útočníkovi pomáhají zesílit útok až sedmdesátkrát. Výsledný provoz dosahoval až 2,3 Tb/s a trval 3 dny. Přesto mu dokázala služba *AWS Shield* odolat.

Toto bylo popsáno ve čtvrtletní zprávě přímo od Amazonu. Ve stejném čtvrtletí *AWS Shield* zaznamenal přes 300 000 útoků. Naprostá většina z nich však nepřesahovala 43 Gb/s.[35]

1.4.7 DDoS útoky jako služba

Vedení DDoS útoků se dnes už nevztahuje pouze na lidi s příslušnými znalostmi jak je provést, ale stává se dostupným v podstatě pro kohokoliv, díky tomu, že se DDoS útoky začaly nabízet jako placená služba, kterou je možné si objednat na takzvaných temnějších částech internetu.

Provozovatelé těchto služeb často mají webové stránky, na kterých si lze útok objednat, podívat se na cenu a jako zajímavost lze uvést, že některé weby dokonce nabízí věrnostní slevy.

Platby zpravidla probíhají v kryptoměnách, například v Bitcoinech. Výsledná cena závisí na několika parametrech, hlavně na scénáři útoku, zemi ze které je útok objednán, cíli a zdroji útoku a celkové délce útoku.

Obecně kratší útoku menšího rozsahu začínají už na nižších desítkách dolarů, avšak v závislosti na zmíněných parametrech se může cena znatelně navýšit.[36][37]

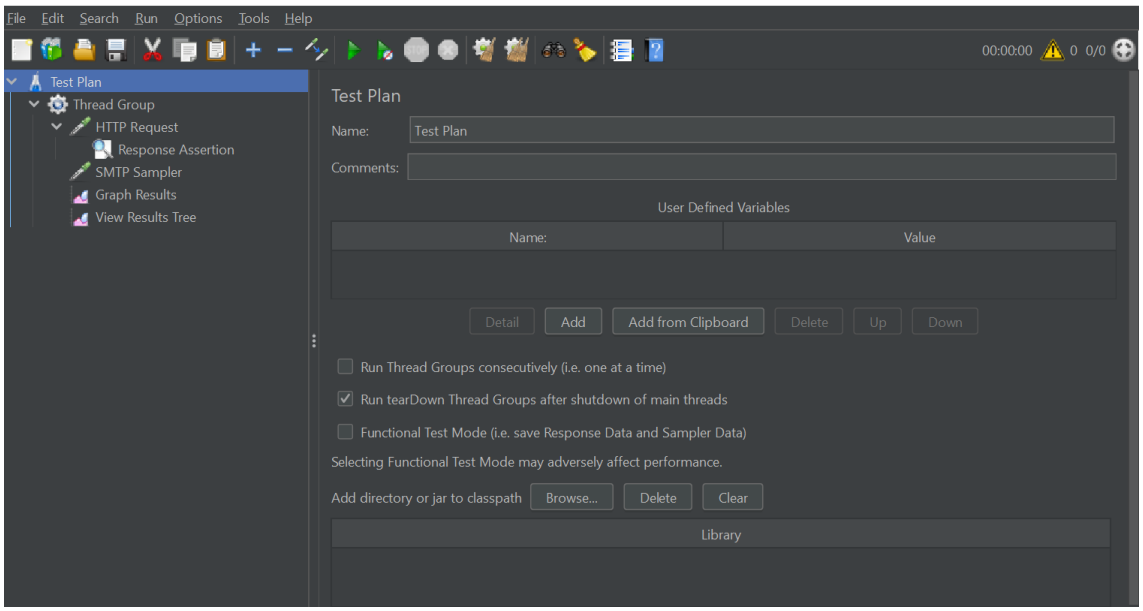
1.5 Apache JMeter

Apache JMeter je open source nástroj, vytvořený v programovacím jazyce Java, který slouží k zátěžovému testování a měření výkonnosti webových aplikací a mnoha dalšího.

Vyvíjen je organizací The Apache Software Foundation, která ho distribuuje pod licencí Apache License 2.0. Nejnovější verzí je Apache JMeter 5.5.

Tento nástroj lze využít k jak statickému, tak i dynamickému testování. K otestování odolnosti a analýze výkonnosti lze použít pro jednotlivé servery, nebo skupiny serverů, pro sítě, nebo síťové prvky pod různými stupni zátěže.

JMeter nabízí grafické uživatelské rozhraní, které je možné vidět na obrázku 1.7. Kromě již zmíněných funkcí a možností podporuje také přidávání dalších naprogramovaných modulů.[38]



Obr. 1.7: Grafické uživatelské rozhraní

1.5.1 Základní prvky

Testovací plán v JMeteru sestává z několika prvků, které jsou zde krátce popsány:

- **Test Plan** - hlavní prvek, do kterého se následně vkládají další komponenty.
- **Thread Group** - udává počet virtuálních uživatelů.
- **Sampler** - slouží k simulování požadavků na systém přes daný protokol.
- **Logic Controller** - udává pořadí samplerů.
- **Config Element** - slouží k nastavování proměnných.
- **Listener** - ukládá a zobrazuje výsledky ve formě grafu, stromového diagramu, nebo tabulky.
- **Timer** - umožňuje vkládání prodlevy mezi posíláním jednotlivých požadavků.
- **Assertion** - může srovnat očekávanou a reálnou odpověď, což slouží k validaci.
- **Pre Processor** - pro operace, které je potřeba udělat před spuštěním.
- **Post Processor** - pro operace, které je potřeba udělat po dokončení.[39]

2 Praktická část

2.1 Koncepce modulu pro digitální podpis

V této části bude popsána koncepce pro rozšíření modulu o digitální podepisování výstupů. Práce bude zakládat na předchozím řešení a potřebně ho rozšíří.

K řešení bude využit světově rozšířený formát *Portable Document Format* (PDF), který byl vytvořen tak, aby fungoval nezávisle na aplikačním softwaru, hardwaru, nebo používaném operačním systému. PDF je dnes standard pod záštitou organizace ISO (Mezinárodní organizace pro normalizaci). Vyvíjen je známou společností Adobe.

PDF umožňuje ukládání textu, obrázků jak v rastrovém, tak vektorovém provedení, pole formulářů, odkazů, metadat o autorovi a mnoho dalšího. Důležitá pro tuto práci je také podpora elektronických podpisů, kterou Portable Document Format umožňuje.[40]

2.1.1 Digitální podpis PDF

Společnost Adobe poskytuje dokumentaci k digitálním podpisům, kde popisuje jak jsou reprezentovány a které možnosti jsou podporovány. Samotný mechanismus sestává z těchto kroků:

1. Dokument, který má být podepsán je převeden na proud bytů.
2. Soubor je zapsán na disk spolu s ponecháním dostatečné rezervy pro podpis a hodnot v poli *ByteRange*.
3. Po určení lokace se doplní příslušné hodnoty.
4. Vypočítá se hash k příslušnému dokumentu. K tomu se využívá hashovací algoritmus, například SHA-256.
5. Hash je podepsán příslušným privátním klíčem a je vygenerován standard PKCS#7.
6. Podpis je vložen do dokumentu.

Ověření lze provést například v programu Adobe Reader od stejné společnosti, která vyvíjí PDF.[41]

2.1.2 Certifikát X.509

Aby byl digitální podpis využitelný, je potřeba také distribuovat příslušné veřejné klíče. Aby nedocházelo k jednoduchému (v praxi velmi složitému, často až nemožnému) vyměňování klíčů mezi jednotlivými subjekty, využívá se PKI (*Public Key Infrastructure)*. PKI obsahuje dvě možnosti - síť důvěry, nebo systém certifikačních autorit.

Systém certifikačních autorit je hierarchický systém složený z množství důvěryhodných subjektů. U certifikační autority (CA) je následně možné si nechat vystavit certifikát. Tento proces zahrnuje vygenerování klíčů, spočítání hashe a podepsání soukromým klíčem. Po vytvoření žádosti o certifikát ji certifikační autorita ověří a při splnění podmínek je certifikát udělen, spolu s podepsáním privátním klíčem od certifikační autority, aby bylo možné certifikát ověřit.

Nejrozšířenějším řešením je certifikát X.509, konkrétně ve své třetí verzi, který spadá pod Mezinárodní telekomunikační unii (ITU). Mezi hlavní položky certifikátu patří:

- Verze: certifikát má tři různé verze, nejrozšířenější je momentálně jeho třetí verze, která obsahuje vše co předcházející dvě, ale je dále rozšířena.
- Sériové číslo: jedinečné číslo pro každý certifikát vystavený certifikační autoritou.
- Algoritmus podpisu CA.
- Vystavitel certifikátu.
- Doba platnosti.
- Údaje o subjektu: název subjektu reprezentovaného certifikátem.
- Veřejný klíč majitele certifikátu.

Mezi další rozšiřující položky patří například použití klíče, zásady certifikátů, identifikátor klíče autority, nebo klíče subjektu.[42][43]

2.1.3 Apache PDFBox

Pro práci s PDF dokumenty byla vybrána knihovna Apache PDFBox od organizace The Apache Software Foundation.

Jedná se o open source knihovnu pro jazyk Java s nástroji pro práci s PDF dokumenty. Knihovna umožňuje vytváření nových dokumentů, manipulaci a úpravy stávajících, extrakci obsahu a mimo jiné také digitální podpis.

Nejnovější verze byla vydána v září 2022 a je označena 2.0.27. Apache PDFBox je vydán pod licencí Apache License v2.0.[44]

2.1.4 Bouncy Castle

Bouncy Castle je kryptografická knihovna pro programovací jazyky Java a C#. K využití je zdarma, jedná se o open source a organizace uvádí, že pomáhá s kryptografickými prostředky už téměř 20 let.

Podporuje mnoho různých standardů a protokolů, mimo jiné také již zmiňovaný X.509 ve verzi 3.[45]

2.2 Tvorba testovacích scénářů

V této části budou popsány vytvořené testovací scénáře, jejich specifikace a následně použité parametry, tak aby odpovídaly zadání.

Scénáře slouží jak k otestování funkčnosti ICT Testeru JMeter, tak mohou následně sloužit k reálnému testování serverů, síťových prvků, či webových aplikací, ačkoliv většina z nich představuje spíše nižší zátěž.

Princip každého z dále popsaných útoků byl vysvětlen v teoretické části této práce.

2.2.1 Tvorba uživatelského manuálu

V rámci této práce byl vytvořen uživatelský manuál k zásuvným modulům „DDoS Simple Thread Group“ a „DDoS Stairs Thread Group“. Tyto zásuvné moduly slouží k vytváření DDoS útoků v testovacím nástroji Apache JMeter. Samotné moduly obsahují několik implementovaných samplerů, které umožňují realizaci konkrétních DDoS útoků.

V rámci manuálu je popsáno v čem se liší moduly „DDoS Simple Thread Group“ a „DDoS Stairs Thread Group“ a k čemu slouží. Je zde také vysvětleno jak je vložit do testovacího plánu i jak následně přidat sampler.

Dále se manuál už věnuje samotným samplerům. Každý z nich je podrobně popsán, včetně doplňujících obrázků. Některé části mají samplery stejné, ty nebyly popisovány pro každý zvlášť, ale bylo zmíněno, že tyto části již byly popsány a fungují stejným způsobem.

V manuálu je obsažen i modul „Thread Group“, do kterého byl zařazen sampler „DDoS - Sampler with interface“, který slouží k vytváření HTTP(S) Flood útoku. Z důvodu jeho fungování byl zařazen právě do Thread Group, na rozdíl od ostatních samplerů.

Tento manuál je k práci přidán jako příloha, z tohoto důvodu nebudou při popisu vytváření testovacích scénářů popsány všechny části sampleru, ale pouze ty relevantní, které bylo potřeba při útoku nastavit.

2.2.2 SYN Flood 100 Mb/s

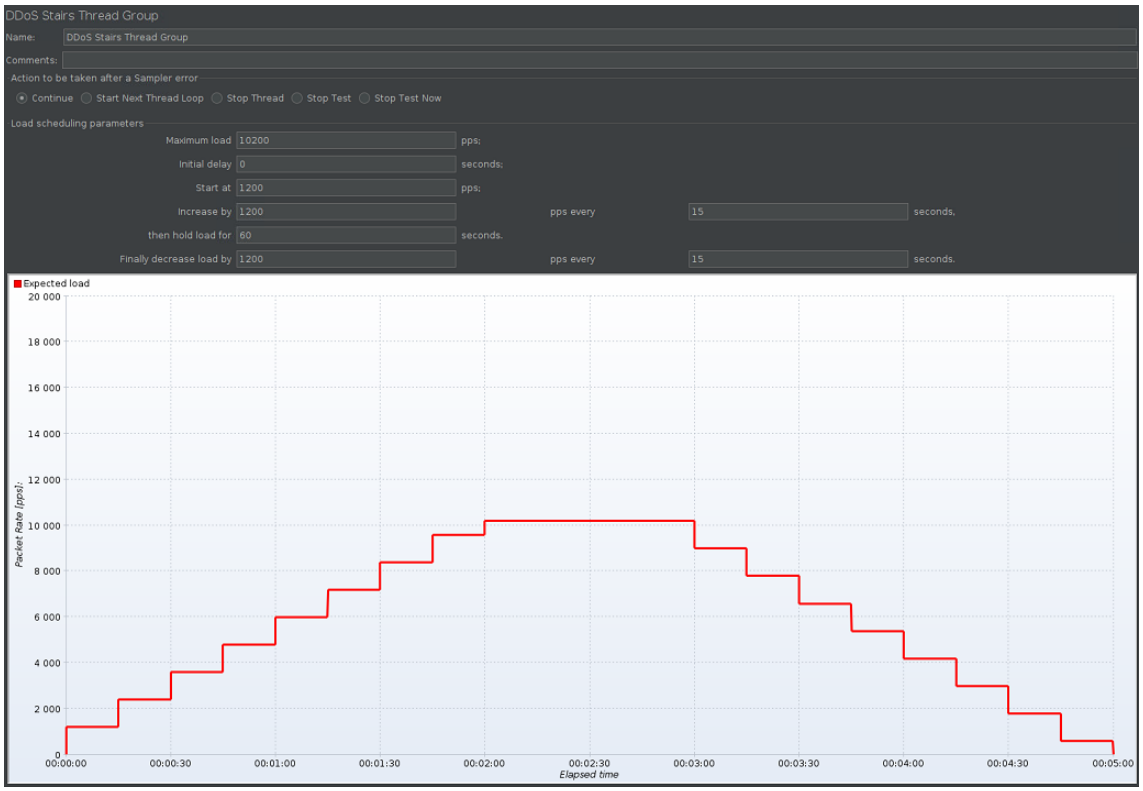
První útok byl zadán s požadavky na to, aby se jednalo o SYN Flood útok v délce 5 minut o intenzitě provozu 100 Mb/s, aby bylo možné zvolit zdrojovou IP adresu v rozsahu 10 možných a aby byl útok vytvořen v komponentě „DDoS Stairs Thread Group“. Poslední zmíněný požadavek odkazuje na to, že bude mít útok schodovitý tvar s postupně narůstající intenzitou a k zadaným 100 Mb/s se tedy dostane až po určité době.

Schodovitý charakter je u DDoS útoku vhodný obzvlášť k tomu, že je možné postupně kontrolovat dostupnost cíle a při případné nedostupnosti lépe identifikovat při jaké zátěži k ní došlo.

Na obrázku 2.1 je zobrazeno nastavení útoku v komponentě „DDoS Stairs Thread Group“ a vykreslený graf provozu.

Je zde možné vidět, že maximální zátěž je nastavena na 10200 paketů za vteřinu. Útok však začíná na pouhých 1200 paketech za vteřinu.

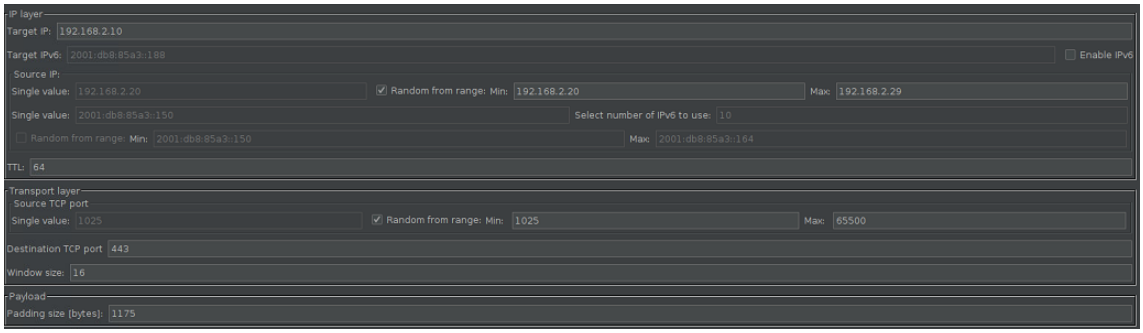
Následně bude narůstat o 1200 pps každých 15 vteřin, dokud se nedostane na zvolenou maximální hodnotu. Jakmile ji dosáhne, bude ji držet po dobu jedné minuty. Následně bude opět každých 15 vteřin klesat o 1200 pps, dokud se nedostane na nulu a test neskončí. Interval 15 vteřin pro nárůst a snižování byl zvolen tak, aby byl dostatečný čas k pozorování případných změn v odezvě serveru a aby byla změna lehce rozpoznatelná v grafu.



Obr. 2.1: Nastavení DDoS Stairs Thread Group pro SYN Flood

Dále bylo potřeba nastavit samotný sampler „DDoS - SYN Flood“. Bylo zvoleno síťové rozhraní, ze kterého bude útok probíhat. Další nastavená pole jsou zobrazena na obrázku 2.2, ostatní zůstalo ve výchozím nastavení. Nastavení tedy obsahuje IP adresu, na kterou útok cílí, zvolení zdrojové IP adresy, což bylo dle zadání provedeno zvolením možnosti náhodné IP adresy z rozsahu 10 možných. Zdrojový port byl také nastaven náhodně z rozsahu, konkrétně 1025-65500. Cílový port je zde nastaven na

443, na kterém naslouchá cílový HTTPS server. Pro zvýšení provozu útoku byl nastaven „Padding size“ na 1175 bajtů.



Obr. 2.2: Nastavení DDoS - SYN Flood

2.2.3 SYN Flood 1 Gb/s

Další ze zadaných útoků je SYN Flood o intenzitě 1 Gb/s. Tento útok má také trvat 5 minut a mít schodovitý nárůst provozu. Stejně tak má používat náhodně 10 různých zdrojových IP adres.

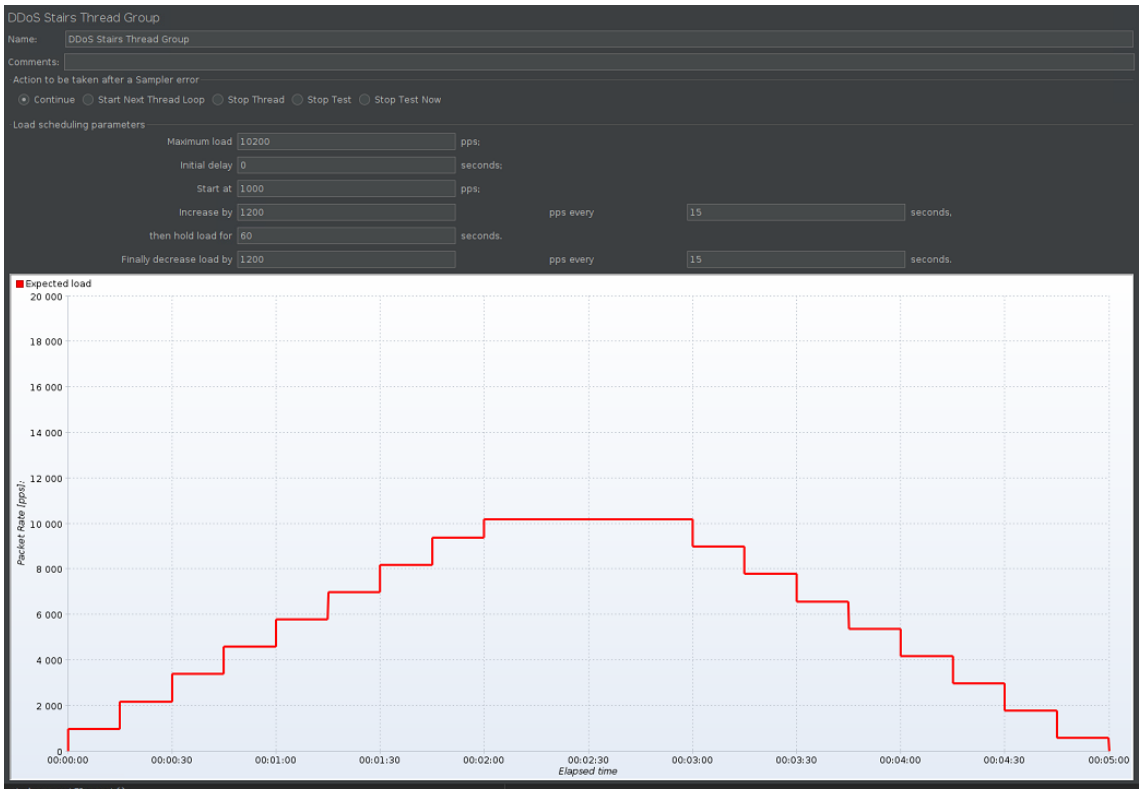
Zadání je tedy velmi podobné jako u předchozího útoku. V tomto případě však má být zvýšen generovaný provoz. V maximu tak tento útok odesílá 102000 paketů za vteřinu. Interval nárůstu a následného snižování zůstal 15 vteřin, tentokrát však dochází ke změně o 12000 pps. Při dosažení maximálního provozu útok tuto intenzitu drží po dobu jedné minuty. Graf má tedy stejný tvar a průběh.

Nastavení samotného sampleru je stejné jako v předchozím útoku - je zde vybráno síťové rozhraní a nastavena IP adresa serveru, na který útok probíhá. Poté náhodný rozsah zdrojových IP adres a zdrojových portů. Cílový port je v tomto případě opět 443. „Padding size“ zůstal 1175 bajtů.

2.2.4 UDP Flood 100 Mb/s

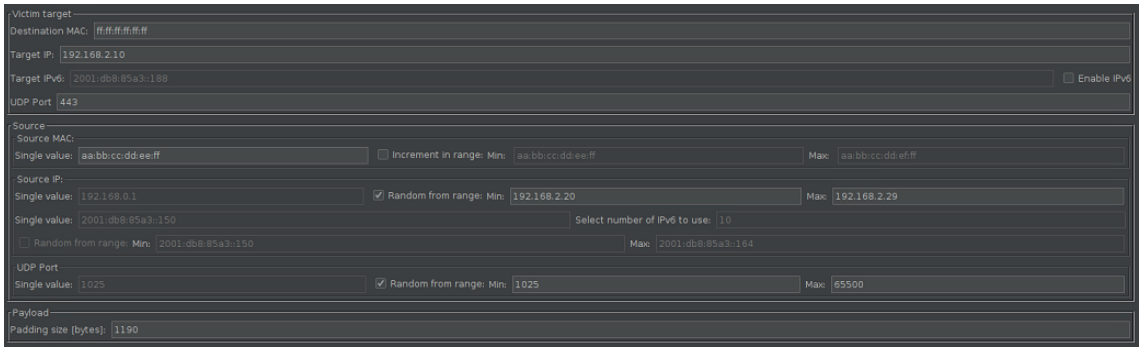
UDP Flood byl zadán v zásuvném modulu „DDoS Stairs Thread Group“, má dosahovat intenzity provozu 100 Mb/s, trvat 5 minut a útok má být veden z 10 náhodných IP adres v zadaném rozsahu.

Jak je možné vidět na obrázku 2.3, maximální počet paketů odesílaných za vteřinu je 10200. Počáteční rychlost je 1000 pps, ta však každých 15 vteřin vzroste o 1200 paketů za vteřinu. Jakmile se útok dostane na svou maximální intenzitu, tak ji bude udržovat po dobu jedné minuty. Poté začne klesat ve stejném intervalu 15 vteřin o 1200 pps, dokud se nedostane na nulu a útok neskončí.



Obr. 2.3: Nastavení DDoS Stairs Thread Group pro UDP Flood 100 Mb/s

Další nastavení bylo v samotném sampleru, nejprve bylo vybráno síťové rozhraní ze kterého bude útok probíhat. Na obrázku 2.4 je zobrazeno nastavení sampleru „DDoS - UDP Flood“.



Obr. 2.4: Nastavení DDoS - UDP Flood

Byla nastavena cílová IP adresa oběti a cílový port. Zdrojová IP adresa byla nastavena pomocí rozsahu, tak aby útok přicházel z 10 různých IP adres. Zdrojový port byl také nastaven tak, aby se náhodně měnil ze zadaného rozsahu.

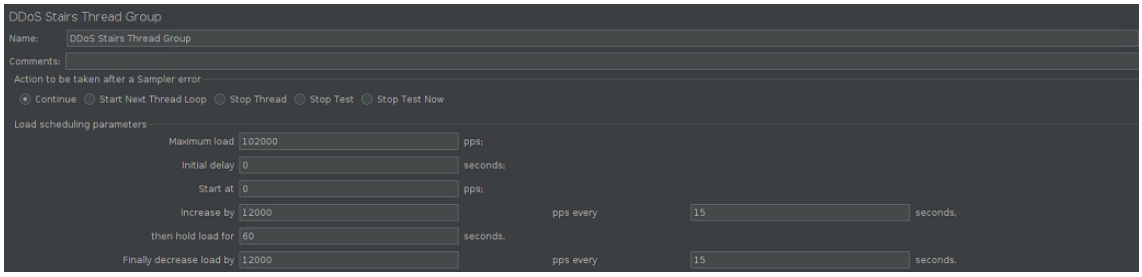
Aby bylo dosaženo požadovaného provozu byl v sekci „Payload“ nastaven „Padding size“ na 1190 bajtů.

2.2.5 UDP Flood 1 Gb/s

Podobně jako v předchozím útoku se jedná o UDP Flood útok, který má trvat 5 minut. Zde má však maximální intenzita útoku dosahovat 1 Gb/s. Útok má být opět veden z 10 různých IP adres.

Jak je vidět na obrázku 2.5, bylo potřeba zvýšit maximální počet posílaných paketů za vteřinu. Ten byl nastaven na 102000 pps. K navyšování rychlosti dochází opět postupně každých 15 vteřin o 12000 pps.

Jakmile útok dosáhne maximální rychlosti, zůstane tak po dobu jedné minuty. Následně začne stejným tempem - každých 15 vteřin o 12000 pps klesat.



Obr. 2.5: Nastavení DDoS Stairs Thread Group pro UDP Flood 1 Gb/s

V samotném sampleru je nastaveno síťové rozhraní použité pro útok. Dále je zvolena IP adresa oběti a příslušný port.

Zdrojové adresy jsou nastaveny náhodně z předdefinovaného rozsahu. Zdrojový port stejně tak.

Pro dosažení požadovaného provozu je nastaveno pole „Padding size“ na 1190 bajtů.

2.2.6 HTTPS Flood 1 dotaz za vteřinu

Při tomto útoku má být nastaven malý provoz, tak aby chodil pouze jeden dotaz za vteřinu a to po dobu dvou minut.

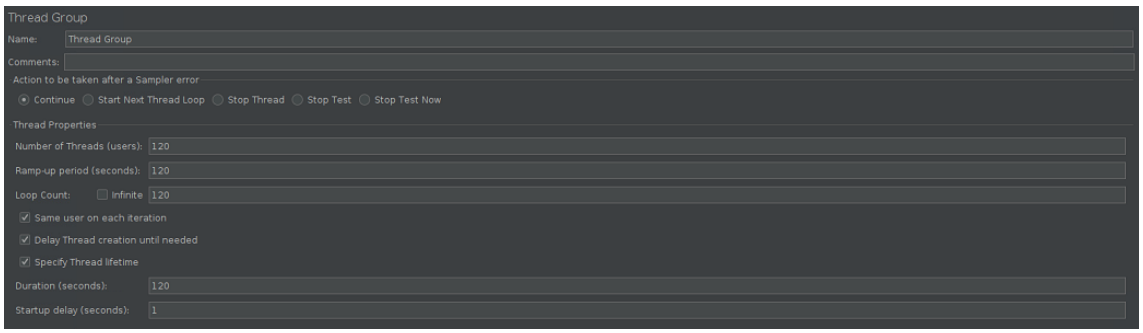
HTTP(S) Flood je na rozdíl od předchozích útoku zařazen do „Thread Group“ pod názvem „DDoS - Sampler with interface“.

Grafické uživatelské rozhraní spolu s nastavenými hodnotami je zobrazeno na obrázku 2.6. Zde je nastaveno 120 vláken s postupným náběhem 120 vteřin. Počet opakování je také nastaven na hodnotu 120.

Spolu s tím je nastaveno využití stejného uživatele při každé iteraci a odložení vzniku vlákna až do jeho využití.

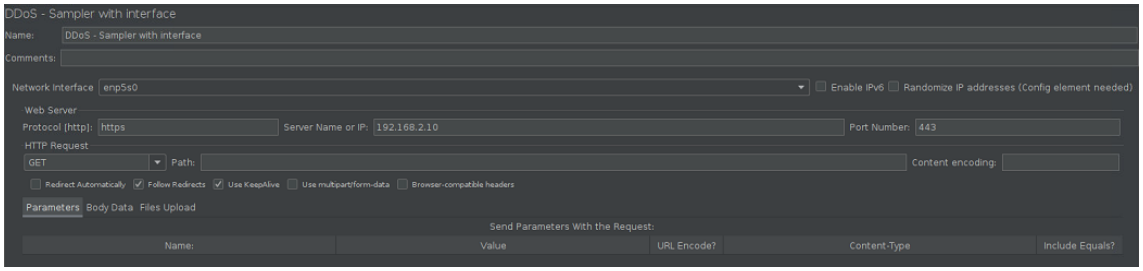
Toto nastavení dohromady způsobuje zapnutí pouze jednoho vlákna každou vteřinu a poté následné vypnutí a čekání další vteřiny.

Nakonec je zaškrtnuta i možnost specifikování doby útoku a to na 120 vteřin, tedy na zadané 2 minuty.



Obr. 2.6: Nastavení Thread Group

Do „Thread Group“ je následně přidán sampler „DDoS - Sampler with interface“, jehož grafické uživatelské rozhraní je zobrazeno na obrázku 2.7.



Obr. 2.7: Nastavení DDoS - Sampler with interface

Zde je pro tento útok nastaveno síťové rozhraní ze kterého bude probíhat útok. Dále je vyplněna část pro nastavení protokolu jako HTTPS což je potřeba specifikovat, jelikož výchozí nastavení je HTTP. Následně je vyplněna IP adresa serveru, na který bude útok realizován a příslušný port - 443, který odpovídá již zmíněnému protokolu HTTPS.

Zbytek možností je ponechán ve výchozím nastavení. Jako „HTTP Request“ je takto nastaven požadavek GET.

2.2.7 HTTPS Flood 100 Mb/s

Zadání tohoto útoku vyžaduje zátěž 100 Mb/s po dobu 5 minut. Zdrojová adresa je opět náhodná s 10 možnostmi v zadaném rozsahu.

Na obrázku 2.8 je zobrazeno nastavení „Thread Group“. Lze zde vidět, že byla nastavena 4 vlákna bez časového náběhu.

Opakování bylo zaškrtnuto na nekonečno, ale spolu s tím byla zaškrtnuta možnost specifikovat časové trvání útoku, což se dělá ve vteřinách - tedy 300 vteřin.

Obr. 2.8: Nastavení Thread Group

Následně byl přidán příslušný sampler, jehož grafické uživatelské rozhraní je zobrazeno na obrázku 2.9.

Obr. 2.9: Nastavení DDoS - Sampler with interface

Zde bylo potřeba nastavit několik věcí. Pro začátek opět síťové rozhraní ze kterého bude útok probíhat. Aby bylo možné útok realizovat z vícero zdrojových IP adres je potřeba zaškrtnout tlačítko „Randomize IP addresses“. S tím se pojí přidání příslušného konfiguračního elementu, což bude popsáno níže.

Dále je nutné specifikovat HTTPS protokol, IP adresu, nebo jmenný název domény oběti a port, což je v tomto případě 443. Útok využívá požadavek GET.

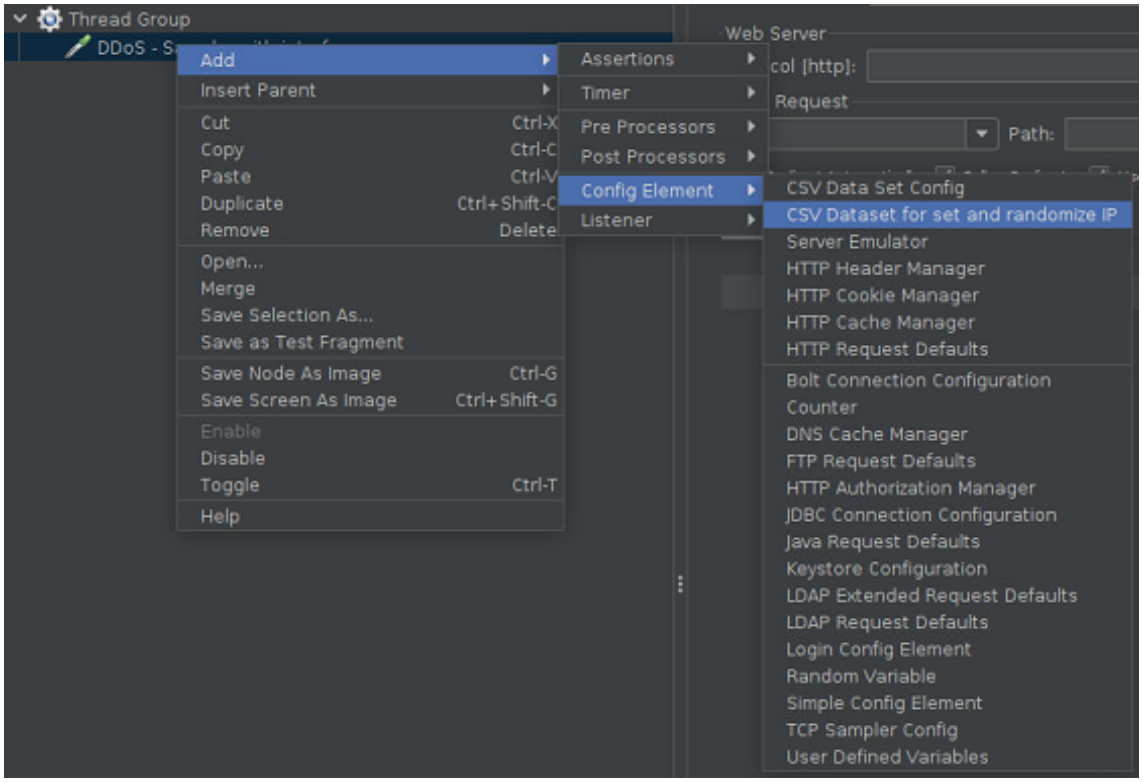
V poslední části nastavení je možné vidět 3 záložky, zde je otevřena záložka „Files Upload“ a v ní je přidán PDF soubor s názvem „100Mbps“.

Během testování bylo zjištěno, že počet nastavených vláken v komponentě Thread Group od jisté hranice, která je pouze několik desítek, již nezvyšuje výsledný odchozí provoz, tedy zátěž.

Přidáváním vláken se lze dostat pouze na zátěž zhruba 75 Mb/s, od této hranice nezáleží na tom, zda se přidají další vlákna.

Takto by nebylo možné dosáhnout požadované zátěže 100 Mb/s. Právě z tohoto důvodu byl v sampleru přidán zmiňovaný soubor. Jeho přidáním se navýší velikost generovaného provozu a zároveň je důvodem proč v tomto scénáři stačí zvolit 4 vlákna. V závislosti na velikosti přiloženého souboru se dá výsledný provoz řídit právě počtem vláken a naopak.

Aby útok probíhal z 10 různých zdrojových adres je nutné doplnit konfigurační element s názvem „CSV Dataset for set and randomize IP“. Jak jej přidat do testovacího plánu je zobrazeno na obrázku 2.10.

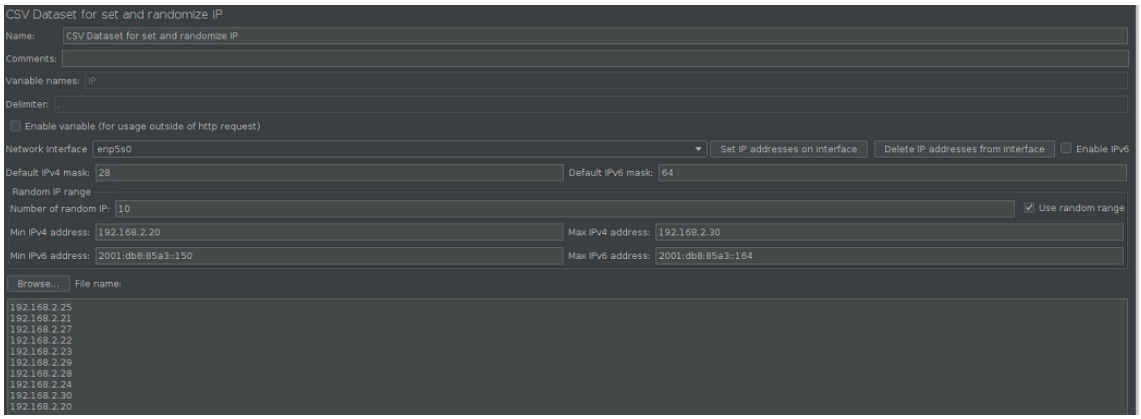


Obr. 2.10: Přidání konfiguračního elementu

V tomto elementu, jenž je zobrazen na obrázku 2.11, se nastavuje síťové rozhraní, počet požadovaných IP adres a jejich rozsah. Poté je nutné zaškrtnout „Use random range“ a nakonec zmáčknout tlačítko „Set IP addresses on interface“. Vygenerované IP adresy se zobrazí ve spodní části.

Po nastavení je potřeba zkontrolovat, zda je v sampleru „DDoS - Sampler with interface“ zaškrtnuto tlačítko „Randomize IP addresse“.

Při testování bylo zjištěno, že se někdy samo deaktivuje při opětovném načtení scénáře.



Obr. 2.11: CSV Dataset for set and randomize IP

2.2.8 HTTPS Flood 1 Gb/s

Tento útok je specifikován jako HTTPS Flood o zátěži 1 Gb/s a má běžet po dobu 5 minut. Zdrojových adres je 10 náhodných ze zadaného rozsahu.

Na obrázku 2.12 je vidět, že tentokrát bylo použito pouze jedno vlákno. To je možné díky použití podstatně většího přiloženého souboru.

Dále je zde nastaven nekonečný počet opakování a specifikován čas trvání testu, což je 300 s.

Obr. 2.12: Nastavení Thread Group

Nastavení sampleru je zobrazeno na obrázku 2.13. Zde je nastaveno síťové rozhraní, použití konfiguračního prvku k využití náhodných IP adres, specifikován protokol HTTPS, zadána IP adresa oběti a specifikován port.

Požadavkem je opět výchozí GET. Ve spodní části je vidět přiložený soubor, který řeší problém s nedostatečným výkonem útoku při pouhém zvyšování počtu vláken.

Pro správnou funkčnost tohoto útoku je ještě nutné doplnit konfigurační prvek, který zajišťuje použití náhodných zdrojových IP adres. V tomto prvku je potřeba

Obr. 2.13: Nastavení DDoS - Sampler with interface

nastavit síťové rozhraní, počet a rozsah IP adres, zaškrtnout použití náhodného rozsahu a vygenerovat IP adresy pro zvolené rozhraní.

2.3 Testování vytvořených scénářů

Pro testování vytvořených scénářů byly využity dva počítače, které byly propojeny pomocí 10 Gb/s optických kabelů přes přepínač, který tuto přenosovou rychlost podporuje.

První počítač sloužil pro realizaci útoků a měl IP adresu 192.168.2.20 a na druhém byl emulován server, který sloužil jako oběť útoku s IP adresou 192.168.2.10.

Specifikace prvního počítače jsou následující:

- OS Ubuntu 22.04 LTS
- CPU Intel i7 11700F
- GPU NVIDIA GeForce GTX 1650
- 64 GB RAM

Procesor tohoto počítače obsahuje 8 jader s 16ti vlákny. Pracovní frekvenci má 2,5 GHz, maximální frekvence je však až 4,9 GHz.

Specifikace druhého počítače:

- OS Ubuntu 22.04 LTS
- CPU Intel Xeon E5-2650 v4
- GPU NVC1
- 128 GB RAM

Tento procesor disponuje 12 jádry s 24 vlákny. Pracovní frekvenci má 2,2 GHz a může se zvýšit na 2,9 GHz.

2.3.1 Network Analyzer

K získávání statistik z jednotlivých testů slouží modul *Network Analyzer*, který mimo jiné sleduje provoz ze zadaného síťového rozhraní a ukládá ho do XML souboru. Aktuální provoz také zobrazuje během testu a to jak odesílaný tak příchozí.

Mezi další sledované statistiky patří:

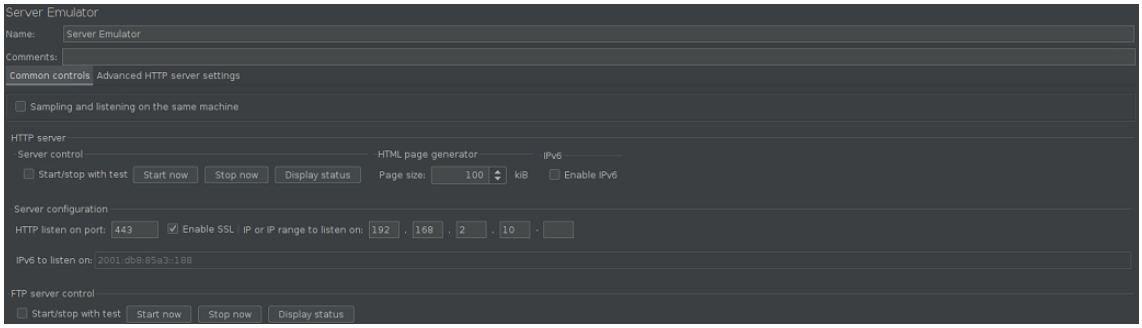
- Průměrná rychlost přenosu
- Celkový objem odeslaných dat
- Využití operační paměti
- Vytížení procesoru

Pro grafické zpracování těchto statistik slouží modul *Report generator*, který vygeneruje přehledné tabulky a grafy ve formátu HTML.

2.3.2 Server emulator

V rámci vývoje JMeteru byl vytvořen zásuvný modul *Server emulator*, který umožňuje emulovat HTTP(S) server.

Jak je možné vidět na obrázku 2.14, pro testování byla nastavena velikost stránky 100 kB, což emulátor řeší přidáním příslušného počtu znaků na vygenerovanou stránku. Dále byl nastaven příslušný port 443 pro HTTPS a zaškrtnuta možnost povolení SSL, tak aby výsledné nastavení odpovídalo zadání. Poté byla nastavena IP adresa - 192.168.2.10, což je adresa druhého počítače, na kterém byl server emulován. Nakonec je potřeba server spustit tlačítkem „Start now“. Pro případ ukončení slouží druhé tlačítko. Pokud je potřeba změnit nastavení serveru, je nutné ho vypnout, provést změnu a poté znovu zapnout.



Obr. 2.14: Nastavení Server emulator

Jak je vidět na obrázku 2.15, spuštěný HTTPS server je možné si zobrazit i ve webovém prohlížeči, jak bylo zmíněno, jedná se o jednoduchou stránku, která mění svou velikost přidáním odpovídajícího počtu znaků. Velikost se zadává v kB.

Obr. 2.15: Webová stránka

2.3.3 Měření odezvy serveru

Kromě sbíraných statistik pomocí modulu *Network analyzer* byla při testování měřena také odezva serveru, aby bylo možné posoudit, zda daný útok má vliv na schopnost serveru obsloužit přicházející dotazy.

To bylo realizováno programem napsaným v jazyce Python, který vytvořil pan Ing. Marek Sikora.

Pro jeho zprovoznění v testovacím scénáři bylo potřeba doinstalovat příslušné knihovny a provést nastavení odpovídající použití v testovacím prostředí.

Nejprve bylo potřeba doinstalovat knihovny pro jazyk Python, to je možné provést zadáním těchto příkazů do terminálu:

```
$ sudo apt install python3-pandas
$ pip install bokeh
```

K tomu ještě interaktivní shell ipython pomocí následujícího příkazu:

```
$ pip install ipython
```

Dále je potřeba v **respTime.py** nastavit adresu serveru, v tomto případě tedy 'https://192.168.2.10/', což je možné vidět ve výpisu 2.1.

Program se spouští v terminálu pomocí příkazu:

```
$ python3 main.py
```

Zde však nastal problém v tom, že program hlásil error „SSLError certificate verify failed: self-signed certificate“.

Problém byl tedy v tom, že emulovaný server pro protokol HTTPS používá samo podepsaný certifikát, který není ověřený žádnou certifikační autoritou.

Aby bylo možné program úspěšně spustit a využívat bylo potřeba provést jednu úpravu v kódu v **respTime.py**.

Výpis 2.1: Úprava respTime.py

```
1  import subprocess, time, interval, functools, requests
2
3  properties = ("request_time", "response_time",
4  "request_duration")
5
6  url = 'https://192.168.2.10/'
7
8  def map_property(status, property):
9      try:
10         value = status[property]
11
12         return str(value)
```

```
13
14      except:
15          return ""
16
17  def get_stats(startTime):
18      reqTime= time.time()
19
20      r = requests.get(url, timeout=60, verify=False)
```

Úprava je viditelná na řádku 20, kde je potřeba doplnit „verify=False“, díky čemuž nebude docházet k ověřování certifikátu. Program bude stále vypisovat varování, ale jeho funkčnost to již neovlivní.

Při běhu program ukládá data do CSV souboru, pro ukončení programu se používají klávesy ctrl + c.

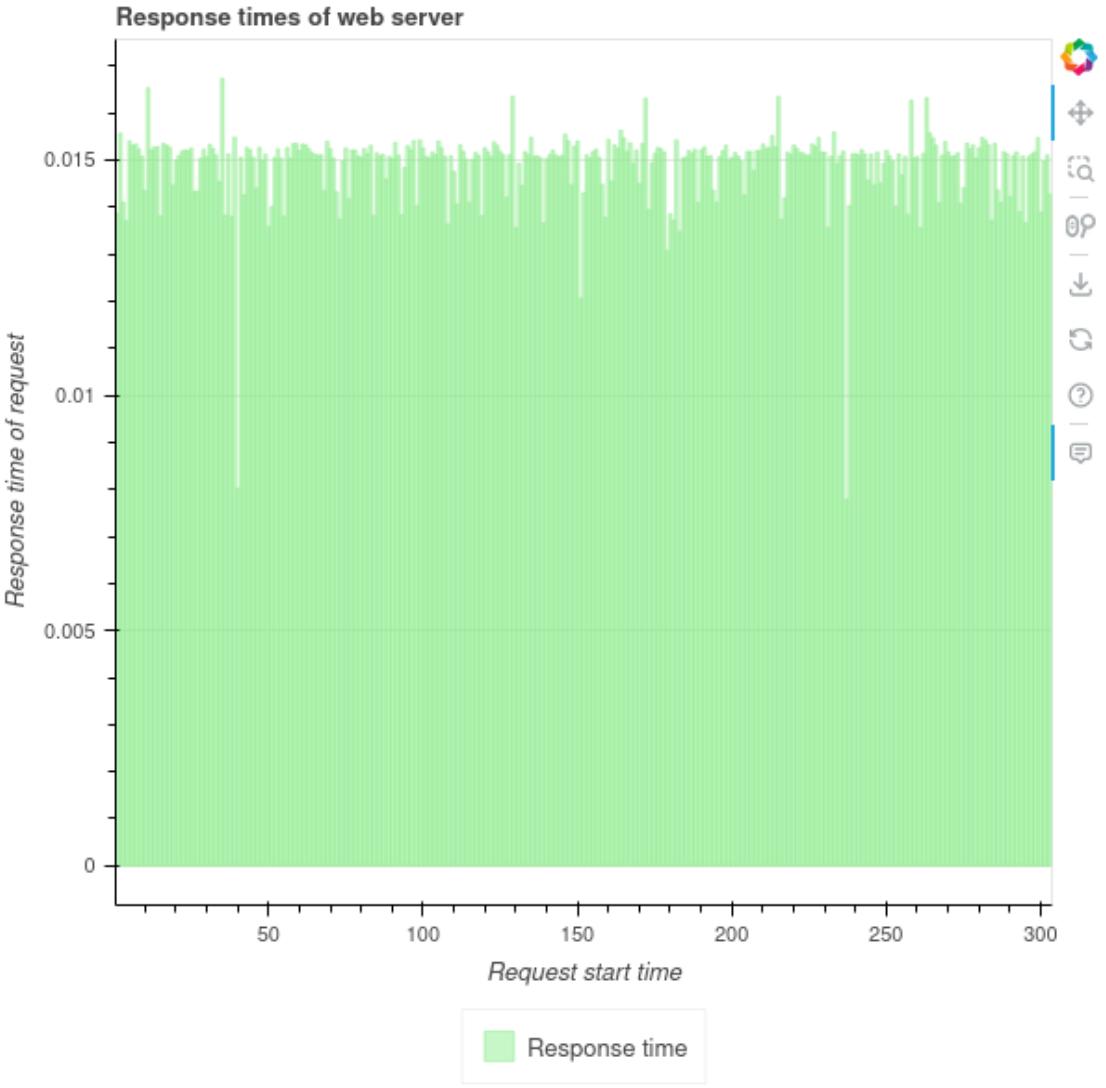
Uložené hodnoty lze vykreslit do grafu na HTML stránce pomocí příkazu:

```
$ python3 graph.py
```

Výsledný graf je zobrazen na obrázku 2.16. Na něm je vidět, že na ose **x** je zobrazen časový průběh testu v sekundách, zatímco na ose **y** je hodnota odezvy serveru, také v sekundách. Z tohoto grafu lze vyčíst, že test běžel po dobu 5 minut a průměrná odezva serveru byla přibližně 15 ms.

Při samotném testování vytvořených scénářů bylo tedy měřeno množství statistik, zatímco útok probíhal pomocí Apache JMeter, ve kterém modul *Network Analyzer* měřil provoz, vytížení paměti RAM a procesoru, tímto programem byla samostatně kontrolována odezva serveru, na který byl útok realizován.

Díky tomu bylo možné sledovat vývoj a případnou změnu v čase odezvy serveru na základě vlivu příchozího útoku.



Obr. 2.16: Odezva serveru

2.3.4 SYN Flood 100 Mb/s

Nastavení útoku již bylo popsáno v předchozí části. Jelikož měl útok schodovitý charakter s postupným nástupem, tak byla průměrná rychlost 65 Mb/s. Za 5 minut tak bylo odesláno 2,4 GB dat. Využití operační paměti se drželo na 7 procentech a využití procesoru průměrně pouze kolem jednoho procenta. Hodnoty jsou také zapsány do tabulky 2.1.

Při kontrole odezvy serveru nebyla zaznamenaná žádná výchylka, odezva byla po celý běh testu přibližně 15 ms. Lze tedy konstatovat, že tento útok neměl dostatečnou intenzitu a oběť nijak nelimitoval.

Průměrná rychlost [Mb/s]	Objem odeslaných dat [GB]	Využití RAM [%]	Využití CPU [%]
65	2,4	7	1

Tab. 2.1: Naměřené hodnoty pro SYN Flood 100 Mb/s

2.3.5 SYN Flood 1 Gb/s

Druhý ze SYN Flood útoků má podobné parametry, intenzita je však desetkrát vyšší. Průměrná rychlost má odpovídající nárůst a byla 650 Mb/s. Stejně tak celkový objem odeslaných dat, který činí 24,3 GB. K vyššímu využití RAM však nedošlo, naopak se snížilo na pouhých 6 procent. Využití procesoru zůstalo přibližně stejné a opět se pohybovalo pouze kolem jednoho procenta.

Data jsou opět zpracována také do tabulky 2.2. Odezva serveru opět nebyla ovlivněna a měla přibližně stejnou hodnotu 15 ms, jako u slabší varianty útoku.

Průměrná rychlost [Mb/s]	Objem odeslaných dat [GB]	Využití RAM [%]	Využití CPU [%]
650	24,3	6	1

Tab. 2.2: Naměřené hodnoty pro SYN Flood 1 Gb/s

2.3.6 UDP Flood 100 Mb/s

UDP Flood o maximální intenzitě 100 Mb/s byl proveden ve schodovitém tvaru. V tabulce 2.3 lze vidět, že jeho průměrná rychlost byla 65,7 Mb/s. Během jeho pětiminutového trvání bylo odesláno 2,5 GB dat. Průměrné využití operační paměti bylo pouze 8 procent. Průměrné využití procesoru pouze kolem jednoho procenta.

Průměrná odezva serveru nebyla nijak ovlivněna a držela se kolem hodnoty 15 ms.

Průměrná rychlost [Mb/s]	Objem odeslaných dat [GB]	Využití RAM [%]	Využití CPU [%]
65,7	2,5	8	1

Tab. 2.3: Naměřené hodnoty pro UDP Flood 100 Mb/s

2.3.7 UDP Flood 1 Gb/s

Při tomto útoku docházelo k většímu kolísání odchozího provozu. Jak je vidět v tabulce 2.4 průměrná rychlost byla o trochu více než desetinásobně větší než v předchozím případu a to 680,9 Mb/s. Bylo odesláno 25,5 GB dat. Nic z toho však neovlivnilo využití operační paměti, které zůstalo na 8 procentech. Využití procesoru se však zvedlo na přibližně 2 procenta.

Odezva serveru nebyla útokem nijak ovlivněna a držela se přibližně na hodnotě 15 ms.

Průměrná rychlost [Mb/s]	**Objem odeslaných dat** [GB]	**Využití RAM** [%]	**Využití CPU** [%]
680,9	25,5	8	2

Tab. 2.4: Naměřené hodnoty pro UDP Flood 1 Gb/s

2.3.8 HTTPS Flood 1 dotaz za vteřinu

Jak je možné vidět v tabulce 2.5 tento lehký útok měl průměrnou rychlost pouhých 0,9 Mb/s. Během dvou minut bylo odesláno pouze 0,005 GB dat. Využití operační paměti bylo 8 procent a využití procesoru přibližně jedno procento.

Již podle zadání testu bylo možné předpokládat nulový vliv na server, což se prokázalo sledováním jeho odezvy.

Průměrná rychlost [Mb/s]	**Objem odeslaných dat** [GB]	**Využití RAM** [%]	**Využití CPU** [%]
0,9	0,005	8	1

Tab. 2.5: Naměřené hodnoty pro HTTPS Flood 1 dotaz za vteřinu

2.3.9 HTTPS Flood 100 Mb/s

V tabulce 2.6 lze vidět že průměrná rychlost tohoto útoku byla 105,2 Mb/s, což je způsobeno lehkým kolísáním odesílaného provozu. Za 5 minut běhu bylo celkově odesláno 3,9 GB dat. Využití operační paměti bylo v průměru z osmi procent. Průměrné využití procesoru se zvedlo na 7,8 procent, kdy nejvytíženější vlákno mělo průměrné využití 26,9 procent.

Odezva serveru nebyla nijak ovlivněna a držela se kolem hodnoty 8 ms.

Průměrná rychlost [Mb/s]	**Objem odeslaných dat** [GB]	**Využití RAM** [%]	**Využití CPU** [%]
105,2	3,9	8	7,8

Tab. 2.6: Naměřené hodnoty pro HTTPS Flood 100 Mb/s

2.3.10 HTTPS Flood 1 Gb/s

Jak je zobrazeno v tabulce 2.7, průměrná rychlost provozu byla 1014,8 Mb/s. Během 5 minut útoku bylo odesláno 37,8 GB dat. Průměrné využití operační paměti bylo pouze 7 procent. Využití procesoru bylo přibližně 2 procenta.

Ani vyšší provoz při HTTPS Flood útoku cílený server nijak neovlivnil a jeho odezva měla hodnotu přibližně 8 ms.

Průměrná rychlost [Mb/s]	Objem odeslaných dat [GB]	Využití RAM [%]	Využití CPU [%]
1014,8	37,8	7	2,2

Tab. 2.7: Naměřené hodnoty pro HTTPS Flood 1 Gb/s

2.3.11 Shrnutí

V této podkapitole byly popsány výsledky testování vytvořených scénářů, které byly popsány v předchozí podkapitole.

Dle popsaných výsledků lze vidět, že realizace testů odpovídá zadání, výsledky se liší pouze o malé hodnoty, jelikož generovaný provoz v průběhu lehce kolísá.

Mimo popsání výsledků v této podkapitole byl vytvořen i podrobný report všech testovaných útoků, který obsahuje podrobné statistiky z modulu *Network Analyzer*, které byly zpracovány pomocí modulu *Report generator*. Report je doplněn o vykreslené grafy jak z modulu *Report generator*, tak vykreslené grafy odezvy serveru. Obsahem reportu je také teoretický úvod k testovaným scénářům. Report je k práci přidán v příloze.

Vzhledem k popsaným parametrům testů, které jsou poměrně malé a vysokým specifikacím strojů, na kterých byly testy provedeny nebylo překvapením, že nedošlo k odepření služby.

2.3.12 HTTPS Flood 9,8 Gb/s

Vzhledem k tomu, že u žádného z testovaných scénářů nedošlo k odepření služby, byl vytvořen další scénář ve snaze tohoto výsledku dosáhnout.

Jedná se opět o útok HTTPS Flood, tentokrát však s větším provozem. Jak je možné vidět na obrázku 2.17 pro útok bylo nastaveno 500 vláken s postupným náběhem po dobu jedné minuty. Počet opakování je nekonečno a celkový čas útoku je specifikován na 300 s.

Další nastavení v sampleru včetně přidání PDF souboru pro zvýšení generovaného provozu a použití náhodných zdrojových adres je stejné jako u útoku „HTTPS Flood 1 Gb/s“.

Takto specifikovaný útok dosahuje provozu až 9,8 Gb/s, což je limitováno rychlostí optického kabelu a během 5 minut bylo odesláno 367,7 GB dat. Využití RAM bylo pouze 10 procent, ale využití procesoru znatelně vzrostlo, jednotlivá vlákna se pohybovala v rozmezí 33-51 procent.

Při realizaci tohoto útoku se odezva serveru zvedne na hodnotu kolem 0,26 sekund, což je značný rozdíl oproti předchozím testům, ale při načtení stránky ve webovém prohlížeči není odezva znatelná.

Obr. 2.17: Nastavení Thread Group pro HTTPS Flood

Aby došlo k dalšímu zvýšení doby odezvy, bylo změněno nastavení emulovaného serveru. Jak je možné vidět na obrázku 2.18, v záložce pokročilého nastavení je možné zaškrtnout možnost vlastního nastavení parametrů serveru. Zde byly všechny hodnoty nastaveny na 1.

Obr. 2.18: Nastavení Server emulator

S takto nastaveným serverem byl útok opakován. Vlivem omezeného serveru se odchozí provoz snížil na přibližně 2 Gb/s, což odpovídá 76,7 GB za 5 minut.

Na obrázku 2.19 lze vidět, že nově omezený server má zvýšenou odezvu, ta dosahuje až 2 s. Na grafu lze vidět postupný náběh vláken, který byl nastaven na jednu minutu. Spolu s ním se zvyšuje i doba odezvy serveru. Po tom co útok dosáhne své maximální intenzity lze prodlevu v odezvě serveru poznat i při načtení stránky ve webovém prohlížeči.

V práci bylo zmíněno, že navýšení počtu vláken v HTTPS Flood útoku od jisté míry nezvyšuje generovaný provoz. Testováním však bylo zjištěno, že má vliv na zvýšení odezvy serveru, proto bylo v tomto útoku nastaveno 500 vláken. Například pro 100 vláken je odezva 0,4 s.

1000 vláken

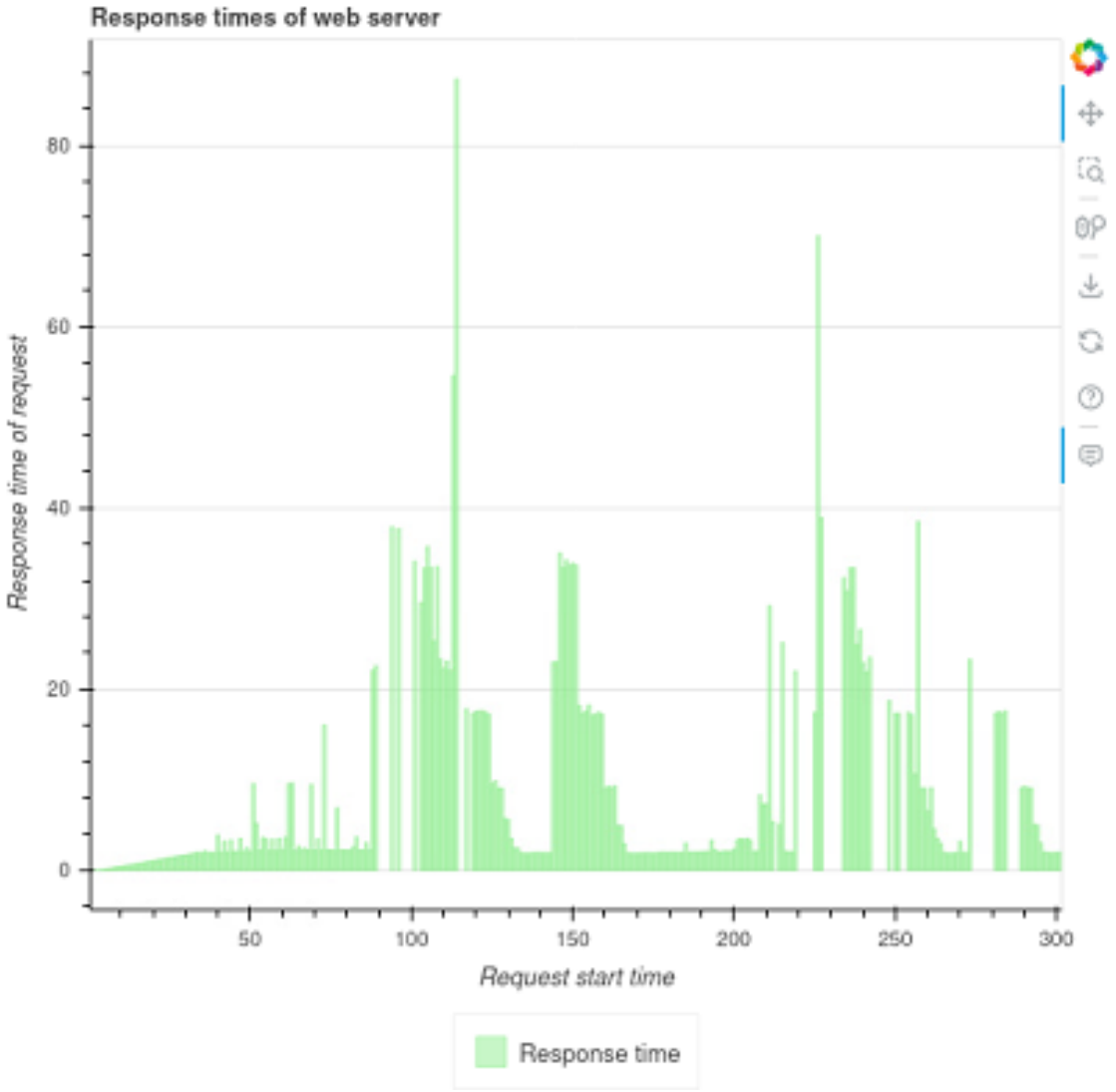
Vzhledem ke zjištěnému poznatku o vlivu navýšení vláken na dobu odezvy byl počet vláken zvýšen na 1000. V tomto nastavení útok způsobuje odepření služby. Na obrázku 2.20 lze vidět dobu odezvy serveru. V grafu jsou hodnoty v řádu desítek

Obr. 2.19: Odezva serveru s novým nastavením a 500 vlákny

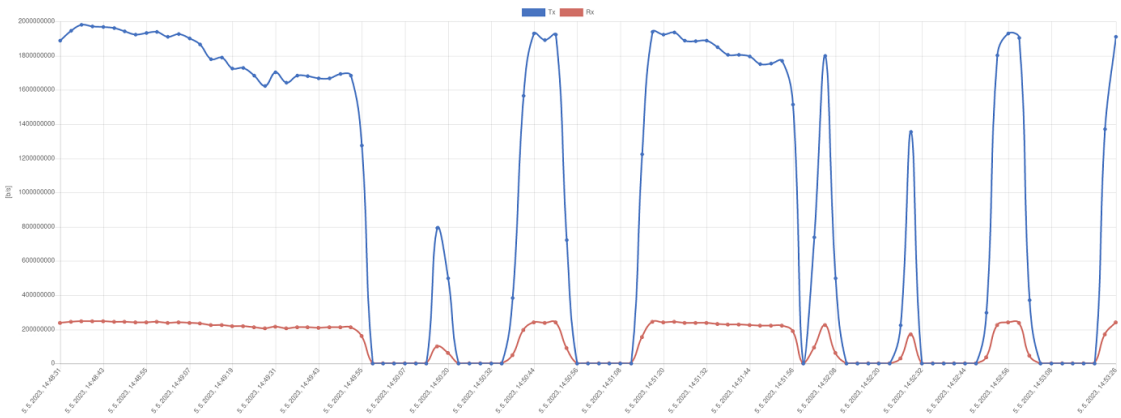
vteřin, ale i chybějící hodnoty, což znamená že odpověď nedorazila. Stejně tak nebylo možné načíst stránku v prohlížeči.

Že došlo k přerušení provozu lze vidět také na grafu z modulu *Report generator*, který je vidět na obrázku 2.21. Na něm je vidět několik výpadků.

Pro dosažení tohoto výsledku je však nutné ponechat omezené nastavení serveru. Při jeho výchozím nastavení tento útok dosáhne zvýšení odezvy pouze na 0,5 s.



Obr. 2.20: Odezva serveru s novým nastavením a 1000 vlákny



Obr. 2.21: Graf provozu při nastavení 1000 vláken

2.4 Konverze reportu do formátu PDF

Dalším z cílů práce bylo doplnit tester o možnost digitálního podepisování generovaných výstupů. V rámci Testeru ICT jsou tyto výstupy řešeny pomocí modulu *Report generator*, který zpracovává uložená data z průběhu testu do přehledného zobrazení ve formě HTML stránky. Právě v tomto zásuvném modulu budou potřebná rozšíření provedena.

Výsledný podepsaný dokument má však být v běžně používaném a světově rozšířeném formátu PDF. Nejprve je tedy potřeba vygenerovaný report převést právě do zmiňovaného formátu PDF.

Při průzkumu dostupných knihoven pro konverzi HTML do PDF se jako nejvhodnější varianta zdála knihovna iText. Toto bylo hodnoceno dle výsledků hledání na vývojářských diskuzních fórech, či webových stránkách popisujících danou problematiku.

Jako nejvhodnější varianta se jevila knihovna iText, která se v těchto zdrojích objevovala často a zároveň je z jejích stránek zjevné, že je dále aktivně vyvíjena.

Pro celkový projekt Testeru ICT je však nevhodná kvůli své licenci AGPL, proto bylo rozhodnuto, že tato knihovna využita nebude.

Po tomto rozhodnutí byla zvolena knihovna Flying Saucer pro programovací jazyk Java, která byla vydána pod vhodnou licencí LGPL ve verzi 2.1. Ta již zajišťuje možnost požadovaného využití.

Kromě knihovny Flying Sauce je ke konverzi do PDF využita ještě knihovna Jsoup, která je vydaná pod licencí MIT.

Pro konverzi je potřeba pouze zvolit cestu k HTML souboru, který má být převeden do formátu PDF. Tento parametr požaduje metoda, která konverzi realizuje. Metoda dále potřebuje znát cestu k CSS souborům a kam uložit konvertované PDF. To je řešeno ze zmiňované cesty k HTML souboru. Pomocí metod *lastIndexOf()* a *substring()* jsou vytvořeny nové *Stringy*, které obsahují požadovanou cestu. Dále už jsou využity metody použitých knihoven k dokončení konverze. Výsledný PDF dokument bude mít stejný název jako původní HTML soubor a bude uložen do stejné složky.

Momentální řešení modulu *Report generator* ve výsledné HTML stránce obsahuje grafy, které jsou dynamicky vykreslovány pomocí JavaScriptu na prvek *canvas*. Tyto grafy provedená implementace nezvládne konvertovat. Tento problém byl konzultován. V rámci širšího vývoje Testeru ICT bude implementováno nové řešení generování reportů. V tomto novém řešení by měly obsažené grafy být ve formátu PNG, což tento problém eliminuje. Konverze HTML souboru obsahujícího PNG obrázky byla otestována. Metoda byla zařazena do třídy ReportGeneratorGUI.

Kromě zmiňovaných grafů byl problém také s některými českými symboly, například „č“. Nejedná se však obecně o všechna písmena s háčkem. Tento problém je možné vyřešit úpravou CSS souboru main, zde lze doplnit definici *@font-face* a doplnit *font-family* v části body. Ve vytvořené metodě, ve které konverze probíhá bude ponechána zakomentovaná část kódu, ve které se odpovídající font nastavuje, tak aby bylo po vytvoření nového generátoru reportů možné toto sjednotit.

2.5 Digitální podpis

Po konverzi do formátu PDF je pro splnění cíle této práce nutná možnost ho následně digitálně podepsat. K tomu byly využity již dříve popsané knihovny PDFBox a Bouncy Castle.

Zdrojové kódy vycházejí ze vzorových příkladů [46], které zveřejnili vývojáři knihovny PDFBox. Ty byly také využity při řešení v rámci Testeru v [47]. Následně byly upraveny, tak aby bylo možné jejich využití v rámci rozšíření modulu *Report Generator*.

Do tohoto modulu byly ve výsledku přidány třídy *CMSProcessableInputStream*, *CreateSignature*, *CreateSignatureBase*, *CreateVisibleSignature*, *PDFSigner* a *SigUtils*.

Výsledná metoda pro vytvoření digitálního podpisu je obsažena ve třídě *PDFSigner*. Konkrétně se jedná o dvě přetížené metody *signPDF()*.

První z nich obsahuje tři parametry - cestu k PDF dokumentu, cestu k certifikátu a heslo k certifikátu. Certifikát je možné použít v archivu ve formátu PKCS #12 (přípona .p12).

Tato metoda digitálně podepíše zvolený dokument zvoleným certifikátem. Pro kontrolu této varianty je potřeba výsledný dokument otevřít například v programu Adobe Reader, kde je možné podpis zkontrolovat v panelu podpisů.

Druhá varianta metody obsahuje čtyři parametry - cestu k PDF dokumentu, cestu k certifikátu, heslo k certifikátu a cestu ke zvolenému obrázku.

V případě této metody vznikne digitální podpis včetně grafického zobrazení. Zda bude zobrazení včetně vlastního obrázku je na uživateli. Pokud nedojde k zadání cesty k obrázku, ale bude zvolen podpis s grafickým zobrazením, bude výsledný podpis vypadat jako na obrázku 2.22. Pokud bude zvolen obrázek, může výsledný podpis vypadat například jako na obrázku 2.23.

Zda bude použita varianta s grafickým zobrazením, nebo bez je řešeno *JCheckBoxem* v grafickém uživatelské rozhraní. Zda bude použit i vlastní obrázek už záleží pouze na uživateli, zda k němu zadá cestu, či nikoliv.

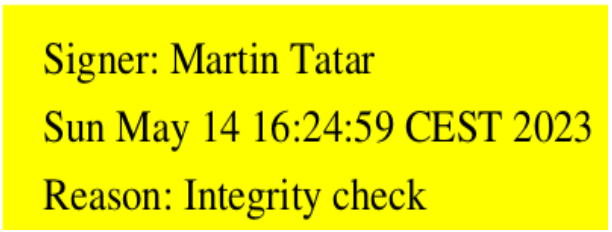
Ve třídě *ReportGeneratorGUI* byla vytvořena metoda *convertToPdfAndSign()*, která nejprve konvertuje vygenerovaný report v HTML do PDF a následně je vytvořen objekt ze třídy *PDFSigner* a na základě zadaných vstupů od uživatele je výsledný PDF dokument podepsán.

Pro vygenerování certifikátu je možné využít například knihovnu OpenSSL pomocí následujících příkazů:

```
$ openssl req -x509 -newkey rsa:4096 -keyout key.pem
$ openssl req -x509 -out cert.pem -days 365 -sha256
-nodes
$ openssl pkcs12 -export -in cert.pem -inkey key.pem
-out cert.p12
```

Zde však byl zjištěn problém s verzí OpenSSL. V operačním systému Ubuntu 22.04 LTS je již ve výchozím nastavení OpenSSL ve verzi 3.0.2. V této variantě nebylo možné certifikát s příponou .p12 úspěšně vygenerovat. V předchozí verzi s dlouhodobou podporou, tedy Ubuntu 20.04 LTS byla OpenSSL ve verzi 1.1.1. V tomto případě bylo možné certifikát vygenerovat. Zde stojí za zmínku, že se takto jedná pouze o vlastní certifikát, který není u žádné certifikační autority.

Jako další možnost je například žádost u CESNETu, kde je možné se přihlásit skrze VUT.



Obr. 2.22: Digitální podpis s grafickým zobrazením bez zvolení obrázku

Obr. 2.23: Digitální podpis s grafickým zobrazením se zvolením obrázku

Podepsaný PDF dokument bude mít stejný název jako ten konvertovaný, ale bude doplněn o „_signed“, uložen bude opět do stejné složky.

2.6 Grafické uživatelské rozhraní

Pro možnost implementace konverze do PDF a digitálního podpisu bylo potřeba upravit stávající grafické uživatelské rozhraní modulu *Report generator*, které je vytvořeno ve třídě *ReportGeneratorGUI*.

Zde byla snaha o zachování přehlednosti a grafického stylu již realizovaného řešení. Nově vytvořené komponenty byly tedy pojmenovávány a tvořeny ve stejném stylu.

Byl vytvořen JPanel s názvem „Signed PDF“, který byl přidán na hlavní panel třídy pod další již vytvořené panely. Na ten byly následně vkládány všechny potřebné prvky pro výsledné řešení.

Mezi ty patří několik prvků JLabel, které slouží k zobrazení popisu. Dále byla použita textová pole. První slouží k vybrání HTML souboru, který bude následně konvertován do PDF dokumentu a poté podepsán. Toto pole je nastaveno tak, že do něj nelze psát. Slouží pouze k výpisu cesty k vybranému HTML souboru. Hned vedle textového pole je tlačítko sloužící k otevření dialogového okna, ve kterém je možné požadovaný soubor vybrat. Je zde nastaveno filtrování zobrazovaných souborů pro usnadnění výběru. Cesta k němu se následně zobrazí. Poté je zde ještě tlačítko pro smazání cesty, to smaže cestu z textového pole a také z proměnné, ve které byla uložena.

Další část je velice podobná. Zde je popisek vyzývající uživatele k vybrání certifikátu. Následuje opět textové pole, u nějž je zakázaná možnost vepisování. Vedle textového pole je tlačítko ke spuštění dialogového okna ve kterém může uživatel vybrat certifikát. V tom je opět vyřešeno filtrování, aby se nezobrazovaly soubory, které nejsou pro tento výběr relevantní. Po výběru je cesta k certifikátu uložena do proměnné a je zobrazena v textovém poli. Textové pole a proměnnou je opět možné vyprázdnit pomocí tlačítka ke smazání.

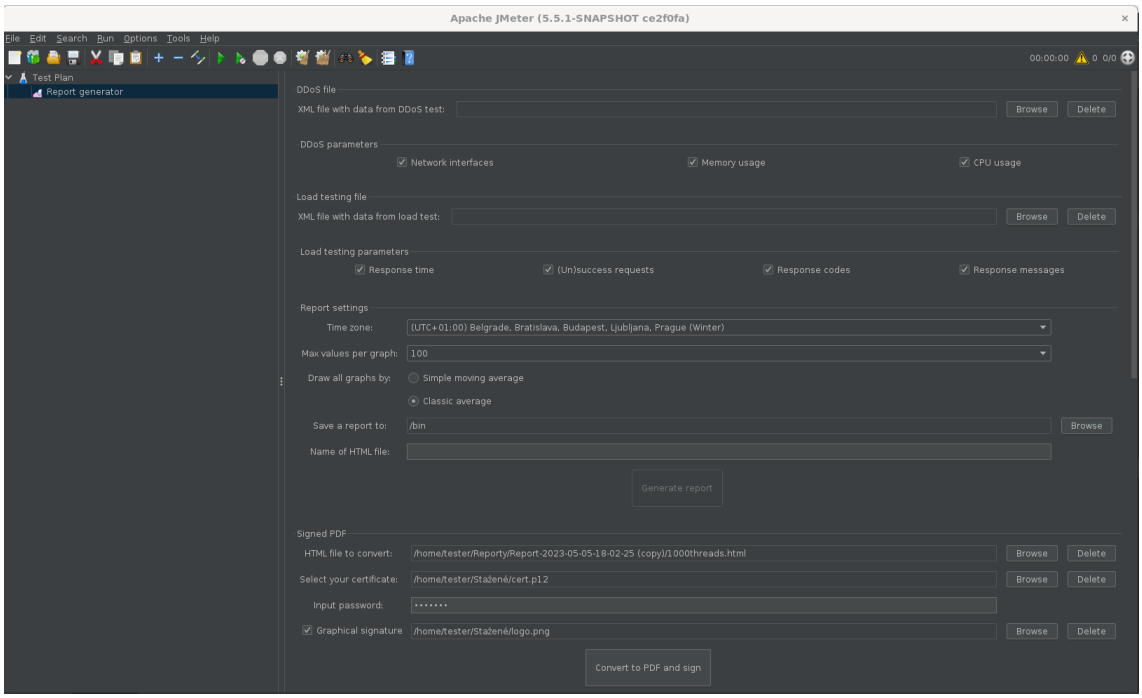
Následuje heslo k certifikátu, to je řešeno pomocí prvku *JPasswordField*. To je použito z bezpečnostních důvodu. Do tohoto pole zadává uživatel heslo, to je však vypisováno zástupnými znaky.

Následně bylo vytvořeno zaškrtávací tlačítko, kterým je možné volit, zda bude podpis s grafickým zobrazením, nebo bez něj. Vedle něj je další textové pole k zobrazení cesty k obrázku, pokud si uživatel přeje zvolit svůj vlastní. Po zaškrtnutí políčka se také aktivuje tlačítko pro vyvolání dialogového okna, ve kterém je možné obrázek vybrat a tlačítko pro smazání cesty a proměnné do které se cesta načetla.

Poté bylo vytvořeno tlačítko, které zavolá metodu pro konverzi a digitální podepsání. To je ve výchozím nastavení vypnuté a aktivuje se až poté, co uživatel zvolí cestu k HTML souboru a k certifikátu. Pokud následně jednu z nich smaže, tlačítko se opět deaktivuje. Výběr obrázku na něj nemá vliv, jelikož je možné provést

digitální podpis s grafickým zobrazením i bez vlastního obrázku.

Nová podoba grafického uživatelského rozhraní modulu *Report generator* je zobrazena na obrázku 2.24



Obr. 2.24: Výsledné grafické uživatelské rozhraní

2.7 Úprava nastavení v Gradle

Jelikož byly v tomto modulu použity další knihovny je také potřeba přidat jejich závislosti do konfigurace nástroje Gradle.

Do závislostí byly přidány **org.jsoup** knihovny JSoup ve verzi 1.14.3, Flying Saucer **flying-saucer-core** a **flying-saucer-pdf**, obě ve verzi 9.1.20, knihovna Apache PDFBox **org.apache.pdfbox**, ve verzi 2.0.26 a knihovny Bouncy Castle **org.bouncycastle** bcpkix-jdk15on, bcutil-jdk15on a bcprov-jdk15on, vše ve verzi 1.70.

V příloze budou také jednotlivé JAR soubory, které se vkládají do složky JMeteru v umístění /lib/ext.

2.8 Odevzdání příloh

Všechny potřebné přílohy k práci byly odevzdány vedoucímu práce na samostatném médiu.

Závěr

V práci s názvem Tester ICT byla probrána problematika DoS útoků. Nejprve bylo popsáno co DoS útok je a v čem se od něj liší DDoS útok. Dále byly rozděleny typy DoS útoků, podle toho jak se snaží dosáhnout nedostupnosti služeb, kde nejběžnějším případem je snaha o vyčerpání omezených zdrojů.

Následně byly popsány vybrané DoS útoky - SYN Flood, UDP Flood, HTTP Flood, DNS Amplification, NTP Amplification, ICMP Flood, Slowloris a R.U.D.Y. U těchto útoků bylo podrobněji popsáno jak fungují, na co se zaměřují a jejich charakteristika.

V další části bylo rozebráno jak se proti DoS útokům bránit a spolu s tím bylo přiblíženo i několik metod, například blackholing, filtrování provozu pomocí scrubbing center, nebo SYN cookies.

Aby byl blíž přiblížen reálný rozsah DoS útoků, ke kterým na internetu dochází neustále, tak byly vybrány příklady skutečně provedených útoků, spolu s popisem a jejich rozsahem. Tímto bylo ukázáno, že v dnešní době tyto útoky dosahují intenzity až v řádu Tb/s, přestože jsou to zatím ojedinělé případy.

Poté byl představen nástroj Apache JMeter, který slouží k funkčnímu i zátěžovému testování a měření výkonnosti například webových aplikací. Tento nástroj lze rozšiřovat dalšími vytvořenými moduly. Mezi ty patří například právě DoS útoky.

Další kapitola se věnuje koncepci rozšíření modulu o možnost digitálního podepisování generovaných výstupů, který byl následně implementován. Nejprve byl popsán celosvětově rozšířený formát PDF. Dále knihovny se kterými je možné digitální podpis realizovat. K tomu byl popsán certifikát X.509, jakožto rozšířená varianta k realizaci digitálního podpisu.

Dále se práce věnovala tvorbě vybraných testovacích scénářů. Mimo konkrétní testovací scénáře byl vytvořen uživatelský manuál pro jednotlivé typy útoků.

Poté byly vytvořené scénáře testovány. Pro toto byly popsány další zásuvné moduly, konkrétně Network Analyzer a Server Emulator. Ty slouží k ukládání statistik z probíhajícího testu a k emulaci serveru, na který testované scénáře cílily. Pro měření odezvy serveru byl využit externí program, jehož využití bylo popsáno.

Výsledky jednotlivých testů byly na základě vybraných statistik popsány. Mimo to byl vytvořen podrobný report obsahující grafy z modulu Report generator.

Jelikož byly vytvořené testovací scénáře spíše slabší intenzity, byl vytvořen další, tak aby bylo dosaženo odepření služby. Vzhledem ke specifikacím použitých strojů k testování muselo však nejprve být upraveno pokročilé nastavení emulovaného serveru, aby mu byl omezen výkon. Následně se podařilo úspěšně simulovat odepření služby.

Dále se práce již věnuje rozšíření modulu o možnost digitálního podpisu. Je zde

popsáno řešení konverze HTML stránky s reportem útoku do formátu PDF a jeho následném podepsání. Popsáno je také doplnění grafického uživatelského rozhraní a úprava nastavení nástroje Gradle.

Do modulu Report generator tak byla úspěšně doplněna funkcionalita pro digitální podpis.

Literatura

[1] *CISA: Understanding Denial-of-Service Attacks* [online]. 2009. Dostupné také z: https://www.cisa.gov/uscert/ncas/tips/ST04-015

[2] *Paloalto Networks: What is a denial of service attack (DoS) ?* [online]. Dostupné také z: https://www.paloaltonetworks.com/cyberpedia/what-is-a-denial-of-service-attack-dos

[3] *Cloudflare: What is a denial-of-service (DoS) attack?* [online]. Dostupné také z: https://www.cloudflare.com/learning/ddos/glossary/denial-of-service/

[4] *CERT - Denial of Service Attacks* [online]. Pittsburgh, 1997. Dostupné také z: https://resources.sei.cmu.edu/asset_files/WhitePaper/1997_019_001_496601.pdf

[5] *Kaspersky: What is a DDoS Attack? - DDoS Meaning* [online]. Dostupné také z: https://www.kaspersky.com/resource-center/threats/ddos-attacks

[6] *Comptia: What Is a DDoS Attack and How Does It Work?* [online]. Dostupné také z: https://www.comptia.org/content/guides/what-is-a-ddos-attack-how-it-works

[7] *Cloudflare: SYN flood attack* [online]. Dostupné také z: https://www.cloudflare.com/learning/ddos/syn-flood-ddos-attack/

[8] *TechTarget: SYN flood attack* [online]. 2022. Dostupné také z: https://www.techtarget.com/searchsecurity/definition/SYN-flooding

[9] *Imperva: UDP Flood* [online]. Dostupné také z: https://www.imperva.com/learn/ddos/udp-flood/

[10] *Cloudflare: UDP flood attack* [online]. Dostupné také z: https://www.cloudflare.com/learning/ddos/udp-flood-ddos-attack/

[11] *Netscout: UDP Flood DDoS Attacks* [online]. Dostupné také z: https://www.netscout.com/what-is-ddos/udp-flood

[12] *Myra: What is an HTTP flood attack?* [online]. Dostupné také z: https://www.myrasecurity.com/en/http-flood-attack/

[13] *Cloudflare: HTTP flood attack* [online]. Dostupné také z: https://www.cloudflare.com/learning/ddos/http-flood-ddos-attack/

[14] *Radware: HTTP Flood* [online]. Dostupné také z: https://www.radware.com/cyberpedia/application-security/http-flood/

[15] *Cloudflare: DNS amplification attack* [online]. Dostupné také z: https://www.cloudflare.com/learning/ddos/dns-amplification-ddos-attack/

[16] *CISA: DNS Amplification Attacks* [online]. 2014. Dostupné také z: https://www.cisa.gov/uscert/ncas/alerts/TA13-088A

[17] *Cloudflare: NTP amplification DDoS attack* [online]. Dostupné také z: https://www.cloudflare.com/learning/ddos/ntp-amplification-ddos-attack/

[18] *CISA: NTP Amplification Attacks Using CVE-2013-5211* [online]. 2014. Dostupné také z: https://www.cisa.gov/uscert/ncas/alerts/TA14-013A

[19] *Netscout: ICMP Flood DDoS Attacks* [online]. Dostupné také z: https://www.netscout.com/what-is-ddos/icmp-flood

[20] *Cloudflare: Ping (ICMP) flood DDoS attack* [online]. Dostupné také z: https://www.cloudflare.com/learning/ddos/ping-icmp-flood-ddos-attack/

[21] *Myra: What is Slowloris?* [online]. Dostupné také z: https://www.myrasecurity.com/en/what-is-slowloris/

[22] *Cloudflare: Slowloris DDoS attack* [online]. Dostupné také z: https://www.cloudflare.com/learning/ddos/ddos-attack-tools/slowloris/

[23] *Imperva: R.U.D.Y. (R-U-Dead-Yet?)* [online]. Dostupné také z: https://www.imperva.com/learn/ddos/rudy-r-u-dead-yet/

[24] *Cloudflare: R U Dead Yet? (R.U.D.Y.) attack* [online]. Dostupné také z: https://www.cloudflare.com/learning/ddos/ddos-attack-tools/r-u-dead-yet-rudy/

[25] *Radware: R.U.D.Y. Attack (R-U-Dead-Yet?)* [online]. Dostupné také z: https://www.radware.com/security/ddos-knowledge-center/ddospedia/rudy-r-u-dead-yet/

[26] *Cloudflare: What is DDoS mitigation?* [online]. Dostupné také z: https://www.cloudflare.com/learning/ddos/ddos-mitigation/

[27] *Computerworld: How to defend against DDoS attacks* [online]. 2004. Dostupné také z: https://www.computerworld.com/article/2564424/how-to-defend-against-ddos-attacks.html

[28] *Radware: Scrubbing Center* [online]. Dostupné také z: https://www.radware.com/security/ddos-knowledge-center/ddospedia/scrubbing-center/

[29] *IPQUALITYSCORE: IP Reputation Check* [online]. Dostupné také z: https://www.ipqualityscore.com/ip-reputation-check

[30] *IBM: IP reputation* [online]. 2021. Dostupné také z: https://www.ibm.com/docs/en/sva/8.0.0.4?topic=matchers-ip-reputation

[31] *Cloudflare: Famous DDoS attacks | The largest DDoS attacks of all time* [online]. Dostupné také z: https://www.cloudflare.com/learning/ddos/famous-ddos-attacks/

[32] *Root: Největší DDoS v historii: 300 Gbps směrem na Spamhaus (aktualizováno)* [online]. 2013. Dostupné také z: https://www.root.cz/clanky/nejvetsi-ddos-v-historii-300-gbps-smerem-na-spamhaus/

[33] *Matthew Prince: Technical Details Behind a 400Gbps NTP Amplification DDoS Attack* [online]. 2014. Dostupné také z: https://blog.cloudflare.com/technical-details-behind-a-400gbps-ntp-amplification-ddos-attack/

[34] *A10: Five Most Famous DDoS Attacks and Then Some* [online]. 2022. Dostupné také z: https://www.a10networks.com/blog/5-most-famous-ddos-attacks/

[35] *Amazon: AWS Shield Threat Landscape Report — Q1 2020* [online]. 2020. Dostupné také z: https://aws-shield-tlr.s3.amazonaws.com/2020-Q1_AWS_Shield_TLR.pdf

[36] *Kaspersky Securelist: The cost of launching a DDoS attack* [online]. 2017. Dostupné také z: https://securelist.com/the-cost-of-launching-a-ddos-attack/77784/

[37] *Radware: DDoS Attacks via DDoS as a Service Tools* [online]. 2016. Dostupné také z: https://www.radware.com/security/ddos-threats-attacks/threat-advisories-attack-reports/ddos-as-a-service/

[38] *The Apache Sofware Foundation: Apache JMeter* [online]. Dostupné také z: https://jmeter.apache.org/

[39] ŠÍPEK, Martin. *Zátěžové testování internetové telefonie*. Brno, 2022. Diplomová práce. Vysoké učení technické v Brně. Vedoucí práce doc. Ing. Petr Číka, Ph.D.

[40] *PDF association: About the Portable Document Format* [online]. Dostupné také z: https://www.pdfa.org/about-us/the-portable-document-format/

[41] *Adobe: Digital Signatures in a PDF* [online]. Dostupné také z: https://www.adobe.com/devnet-docs/acrobatetk/tools/DigSig/Acrobat_DigitalSignatures_in_PDF.pdf

[42] *Keyfactor: What is PKI and How Does it Work?* [online]. Dostupné také z: https://www.keyfactor.com/resources/what-is-pki/

[43] *Microsoft: Kurz: Principy certifikátů veřejných klíčů X.509* [online]. 2022. Dostupné také z: https://learn.microsoft.com/cs-cz/azure/iot-hub/tutorial-x509-certificates

[44] *The Apache Sofware Foundation: PDFBox* [online]. Dostupné také z: https://pdfbox.apache.org/

[45] *The Legion of the Bouncy Castle: Bouncy Castle* [online]. Dostupné také z: https://www.bouncycastle.org/

[46] *The Apache Sofware Foundation: pdfbox/examples/signature* [online]. Dostupné také z: https://svn.apache.org/viewvc/pdfbox/trunk/examples/src/main/java/org/apache/pdfbox

[47] SKOČDOPOLE, Jan. *Síťová sonda*. Brno, 2022. Diplomová práce. Vysoké učení technické v Brně. Vedoucí práce Doc. Ing. Václav Zeman, Ph.D.

Seznam symbolů a zkratek

DoS	Denial-of-service
DDoS	Distributed-denial-of-service
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
ICMP	Internet Control Message Protocol
HTTP	Hypertext Transfer Protocol
WWW	World Wide Web
HTTPS	Hypertext Transfer Protocol Secure
URL	Uniform Resource Locator
TLS	Transport Layer Security
DNS	Domain Name System
NTP	Network Time Protocol
ISP	Internet Service Provider
BGP	Border Gateway Protocol
AWS	Amazon Web Services
CLDAP	Connection-less Lightweight Directory Access Protocol
PDF	Portable Document Format
ISO	International Organization for Standardization
PKI	Public Key Infrastructure
CA	Certification Authority
ITU	International Telecommunication Union
HLS	HTTP Live Streaming
SIP	Session Initiation Protocol
MAC	Media Access Control

TTL	Time To Live
CPU	Central Processing Unit
GPU	Graphics Processing Unit
RAM	Random Access Memory
CSV	Comma Separated Values
XML	Extensible Markup Language
HTML	Hypertext Markup Language
SSL	Secure Socket Layer
CSS	Cascading Style Sheets
PNG	Portable Network Graphics
JAR	Java Archive