

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2020

Andrej Krivulčík



# 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

## GENERÁTOR SLOW DOS ÚTOKŮ

SLOW DOS ATTACKS GENERATOR

### BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

### AUTOR PRÁCE

AUTHOR

Andrej Krivulčík

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Marek Sikora

BRNO 2020

# Bakalářská práce

bakalářský studijní program **Informační bezpečnost**

Ústav telekomunikací

**Student:** Andrej Krivulčík

**ID:** 203414

**Ročník:** 3

**Akademický rok:** 2019/20

**NÁZEV TÉMATU:**

## Generátor Slow DoS útoků

### POKYNY PRO VYPRACOVÁNÍ:

Slow DoS je specifická a poměrně nová skupina útoků s odepřením služby, způsobující nedostupnost síťových služeb, nejčastěji webových stránek. Slow DoS útoky se vyznačují velmi malým provozem a velkou podobností s legitimním provozem běžných uživatelů, díky čemuž jsou velmi efektivní a obtížně odhalitelné.

Jedním z cílů této práce je analyzovat vybrané typy slow DoS útoků (Slowloris, Slow POST, Slow Read), navrhnout modely těchto útoků s co nejvyšší možnou mírou abstrakce a vytvořit software pro generování těchto útoků včetně přehledného GUI umožňujícího detailní nastavení útoků. Dalším cílem práce je pomocí vytvořeného generátoru porovnat úroveň zabezpečení současných webových serverů a jejich bezpečnostních modulů a optimalizovat vedení útoků tak, aby byly útoky schopny tato zabezpečení přejít a způsobit odmítnutí služby legitimním uživatelům.

### DOPORUČENÁ LITERATURA:

[1] SIKORA, Marek a Petr BLAŽEK. Intrusion Prevention System of Slow HTTP DoS and DDoS attack. Elektrov revue [online]. 2017, 2017(4), 9 [cit. 2019-04-20]. Dostupné z: <https://bit.ly/2GUQsTL>

[2] CAMBIASO, Enrico, Gianluca PAPALEO, Giovanni CHIOLA a Maurizio AIELLO. Slow DoS attacks: definition and categorisation. International Journal of Trust Management in Computing and Communications [online]. 2013, 1(3/4) [cit. 2019-03-14]. DOI: 10.1504/IJTMCC.2013.056440. ISSN 2048-8378. Dostupné z: <https://bit.ly/2mcGdSZ>

**Termín zadání:** 3.2.2020

**Termín odevzdání:** 8.6.2020

**Vedoucí práce:** Ing. Marek Sikora

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

### UPOZORNĚNÍ:

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

## **ABSTRAKT**

Práca je zameraná na problematiku Slow DoS útokov a ich generovanie. Sú porovnané 3 najpopulárnejšie webové servery, ich ochranné moduly proti týmto útokom. Bližšie je popísaný sieťový model TCP/IP, protokol HTTP (Hypertext Transfer Protocol), jednotlivé DoS útoky, či už Slow GET, Slow POST alebo Slow Read ale aj záplavové. Následne je uvedený samotný generátor na tieto útoky, jeho popis a funkčnosť.

## **KĽÚČOVÉ SLOVÁ**

Pomalé DoS útoky, generátor, Slowloris, Slow GET, Slow POST, Slow Read, WPF, Apache2, Nginx, lighttpd, DDoS

## **ABSTRACT**

The work is focused on Slow DoS attacks and generating them. There are compared 3 most popular web servers and their defensive modules against this type of attacks. Closed are described network model TCP/IP, protocol HTTP (Hypertext Transfer Protocol), each DoS attack, Slow GET, Slow POST or Slow Read and also flood attacks. Afterwards the attack generator is described, with its functionality.

## **KEYWORDS**

Slow DoS útoky, generator, Slowloris, Slow GET, Slow POST, Slow Read, WPF, Apache2, Nginx, lighttpd, DDoS

KRIVULČÍK, Andrej. *Generátor Slow DoS útokov*. Brno, 2020, 52 s. Bakalárska práca. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedúci práce: Ing. Marek Sikora,

## VYHLÁSENIE

Vyhlasujem, že svoju bakalársku prácu na tému „Generátor Slow DoS útokov“ som vypracoval samostatne pod vedením vedúceho bakalárskej práce, s využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

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

Brno .....

.....

podpis autora

## POĎAKOVANIE

Rád by som poďakoval vedúcemu práce pánovi Ing. Marekovi Sikorovi za odborné vedenie, konzultácie, trpezlivosť a podnetné návrhy k práci.

# Obsah

Úvod	11
<b>1 Sieťový model TCP/IP</b>	<b>12</b>
1.1 Vrstva sieťového rozhrania	12
1.2 Sieťová vrstva	13
1.3 Transportná vrstva	13
1.4 Aplikačná vrstva	14
1.4.1 Hypertext Transfer Protocol	14
<b>2 Denial of Service</b>	<b>16</b>
2.1 Distributed Denial of Service	16
2.2 ARP spoofing	16
2.3 Záplavové DoS útoky	17
2.4 Pomalé DoS útoky	17
2.4.1 Slowloris	18
2.4.2 Slow POST	19
2.4.3 Slow Read	19
<b>3 Webové servery a ich zabezpečenie proti DoS útokom</b>	<b>21</b>
3.1 Zabezpečenie serveru Apache2	21
3.2 Zabezpečenie serveru Nginx	21
3.3 Zabezpečenie serveru lighttpd	22
3.4 Dodatočná ochrana	22
<b>4 Modely útokov</b>	<b>23</b>
4.1 Slowloris	23
4.2 Slow Post	24
4.3 Slow Read	26
4.4 DDoS	26
<b>5 Praktická implementácia - Packet cannon</b>	<b>29</b>
5.1 Vývojové prostredie	29
5.1.1 Vývoj generátora	29
5.1.2 Grafické užívateľské prostredie	31
5.2 DDoS	35

<b>6 Testovanie</b>	<b>36</b>
6.1 Apache2 . . . . .	36
6.1.1 Slowloris na Apache2 . . . . .	36
6.1.2 Slow Post na Apache2 . . . . .	37
6.1.3 Slow Read na Apache2 . . . . .	38
6.2 Nginx . . . . .	39
6.2.1 Slowloris na Nginx . . . . .	39
6.2.2 Slow Post na Nginx . . . . .	39
6.2.3 Slow Read na Nginx . . . . .	40
6.3 lighttpd . . . . .	40
6.3.1 Slowloris na lighttpd . . . . .	41
6.3.2 Slow Post na lighttpd . . . . .	41
6.3.3 Slow Read na lighttpd . . . . .	42
<b>7 Testovanie DDoS</b>	<b>43</b>
7.0.1 DDoS na serveri Apache2 . . . . .	43
7.0.2 DDoS na serveri Nginx . . . . .	45
7.0.3 DDoS na serveri lighttpd . . . . .	45
<b>Záver</b>	<b>48</b>
<b>Literatúra</b>	<b>49</b>
<b>Zoznam symbolov, veličín a skratiek</b>	<b>51</b>
<b>A Obsah priloženého média</b>	<b>52</b>



# Zoznam obrázkov

1.1	Skladanie dát na jednotlivých vrstvách podľa modelu TCP/IP . . . . .	12
1.2	Zloženie Ethernet rámcu štandardu IEEE 802.3 . . . . .	13
1.3	Zloženie paketu pri protokole IP . . . . .	13
1.4	Schéma TCP paketu . . . . .	14
2.1	Príklad výslednej hlavičky pre SlowLoris . . . . .	18
2.2	Príklad výslednej hlavičky pre Slow POST . . . . .	19
4.1	Diagram útoku Slowloris . . . . .	23
4.2	Príklad hlavičky útoku Slowloris (zvýraznené sú samotné dáta hlavičky)	24
4.3	Príklad hlavičky útoku Slowloris (zvýraznené sú udržiavacie dáta) . .	24
4.4	Diagram útoku Slow Post . . . . .	25
4.5	Príklad hlavičky útoku Slow Post (zvýraznené sú samotné dáta hlavičky) . . . . .	26
4.6	Príklad udržiavacieho paketu útoku Slow Post (zvýraznené je znak (dáta) udržiavacieho paketu) . . . . .	26
4.7	Diagram útoku Slow Read . . . . .	27
4.8	Príklad najmenšieho možného požiadavku na útok Slow Read . . . . .	28
4.9	Diagram útoku ARP spoofing . . . . .	28
4.10	Príklad výpisu ARP tabuľky po ARP spoofingu . . . . .	28
5.1	Sieťové zapojenie jednotlivých strojov . . . . .	29
5.2	Konfiguračné okno pre generátor . . . . .	32
5.3	Výber sieťového adaptéru . . . . .	32
5.4	Základné nastavenia pre útok Slowloris . . . . .	33
5.5	Základné nastavenia pre útok Slow Post . . . . .	33
5.6	Základné nastavenia pre útok Slow Read . . . . .	33
5.7	Základné nastavenia pre falošných klientov . . . . .	34
5.8	Nastavenia DDoS . . . . .	35
6.1	Graf priebehu útoku Slowloris . . . . .	37
6.2	Namerané hodnoty pri útoku Slow Post . . . . .	38
6.3	Priebeh útoku Slow Read . . . . .	39
6.4	Priebeh útoku Slow Read na serveri Nginx . . . . .	40
6.5	Priebeh útoku Slowloris na serveri lighttpd . . . . .	41
6.6	Priebeh útoku Slow Post na serveri lighttpd . . . . .	42
6.7	Priebeh útoku Slow Read na serveri lighttpd . . . . .	42
7.1	Priebeh útoku Slowloris na serveri Apache2 s modulom DDoS . . . . .	43
7.2	Priebeh útoku Slow Post na serveri Apache2 s modulom DDoS . . . . .	44
7.3	Priebeh útoku Slow Read na serveri Apache2 s modulom DDoS . . . . .	44
7.4	Priebeh útoku Slow Read na serveri Nginx s modulom DDoS . . . . .	45

7.5	Priebeh útoku Slowloris na serveri lighttpd s modulom DDoS . . . . .	46
7.6	Priebeh útoku Slow Post na serveri lighttpd s modulom DDoS . . . . .	46
7.7	Priebeh útoku Slow Read na serveri lighttpd s modulom DDoS . . . . .	47

# Zoznam výpisov

5.1	Príklad skladania jednotlivých vrstiev . . . . .	30
-----	--	----

# Úvod

Bakalárska práca sa zaoberá problematikou útokov DoS (z anglického „Denial of Service“), v preklade odopretie služby a ich vytvorením. DoS útoky patria k najčastejším útokom v počítačovej sieti. Vo veľkom množstve sa zneužívajú protokoly ARP (Address Resolution Protocol), TCP (Transmission Control Protocol), HTTP (Hypertext Transfer Protocol) a to kvôli ich nie ideálnej implementácii. Pri takýchto útokoch sa server javí ako nedostupný, poprípade veľmi pomaly odpovedá. Dôvod takejto veľkej popularity útokov spočíva v ľahkej dostupnosti veľkého počtu generátorov takýchto útokov. Útočníci vytvárajú nové spôsoby ako obísť ochranné mechanizmy autorov týchto serverov. Výsledkom je neustály boj útokov a obrán.

Táto práca je hlavne zameraná na celkovo nové odvetvie DoS útokov, tzv. pomalé DoS útoky, ktoré sa prejavujú malým množstvom prenesených informácií, čo dopomáha k horšiemu rozpoznaníu od normálnej komunikácie a relatívne veľkou účinnosťou. Sú zamerané prevažne na aplikačnú a transportnú vrstvu sieťového modelu TCP/IP.

Cieľom práce je vytvoriť generátor 3 Slow DoS útokov a to Slowloris (Slow Get), Slow Post a Slow Read a ich následné otestovanie na najpopulárnejších webových serveroch s ich ochrannými modulmi.

V prvej kapitole budú čitatelia oboznámení so samotným modelom TCP/IP, budú popísané jeho jednotlivé vrstvy, ich dôležité a zásadné časti, funkcionality a protokoly operujúce na danej vrstve.

V druhej kapitole sú podrobnejšie popísané Denial of Service útoky, či už aj Slow Denial of Service útoky, popísané jednotlivé druhy útokov či už záplavové útoky alebo aj pomalé DoS útoky aj s príkladmi takýchto útokov.

V tretej kapitole budú opísané samotné webové servery, na ktorých bude následne generátor otestovaný a ich moduly na ochranu proti týmto útokom.

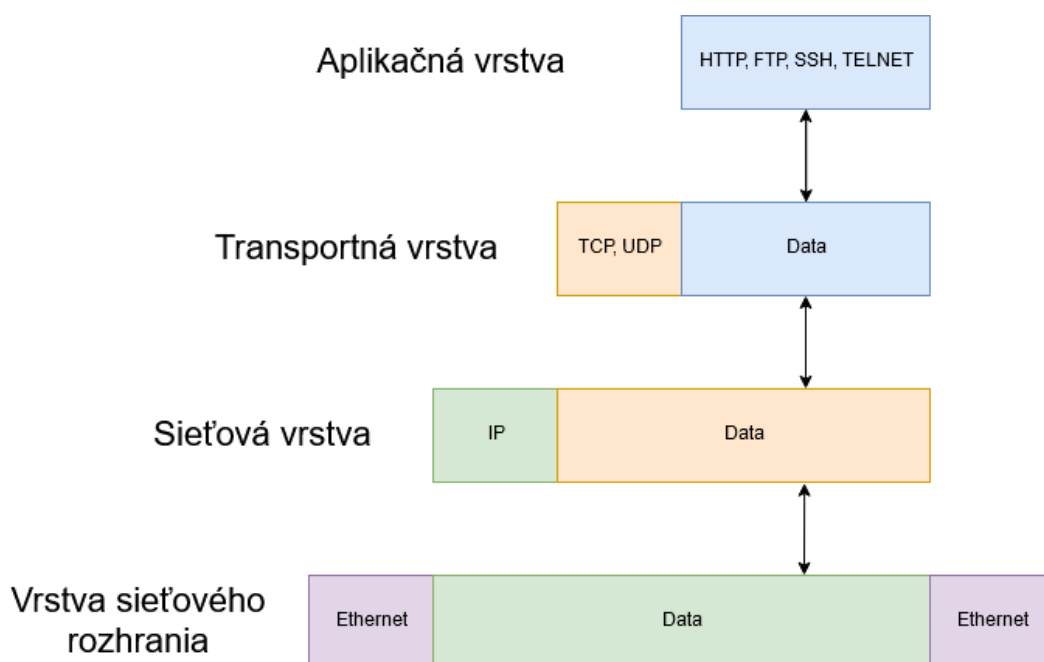
V štvrtej kapitole sú popísané modely útokov, ako fungujú a popísané jednotlivé pakety ako ich generuje falošný klient a ako na ne server odpovedá.

V piatej kapitole bude nasledovať vlastná implementácia generátora napísaná v C#, grafické užívateľské prostredie vo WPF (Windows Presentation Form), ktorý je modifikovateľný a konfigurovateľný podľa potrieb užívateľa.

Posledná, piata kapitola je orientovaná na samotné testovanie a výsledky samotného generátora na jednotlivých serveroch Apache2, Nginx a lighttpd aj s ich ochrannými modulmi aj s variantou DDoS útokov.

# 1 Sieťový model TCP/IP

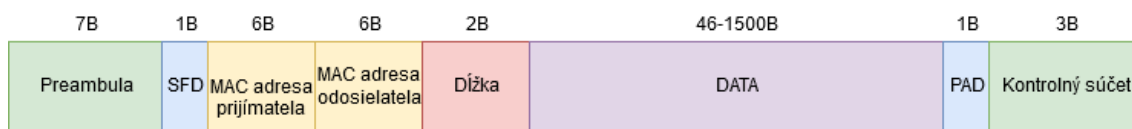
K jedným z najpoužívanějších sieťovým modelom sa radí model TCP/IP. Model vychádza z referenčného modelu ISO/OSI. TCP/IP model obsahuje niekoľko vrstiev. Na každej vrstve pracuje určitá sada protokolov, ktoré rozširujú pôsobnosť a možnosti využitia tohoto modelu. Pri sieťovej komunikácii sa z najvyššej vrstvy (aplikačnej) postupne spracovávajú dáta tzv. zapúzdrujú, ku ktorým sa podľa jednotlivých protokolov pridávajú hlavičky, poprípade aj zápätia. Pri prijatí takýchto dát sa spracovávajú dáta v opačnom smere ako boli vytvorené tzn. od vrstvy sieťových rozhraní až na aplikačnú vrstvu [1].



Obr. 1.1: Skladanie dát na jednotlivých vrstvách podľa modelu TCP/IP

## 1.1 Vrstva sieťového rozhrania

Najnižšia vrstva v modeli TCP/IP je vrstva sieťového rozhrania, využívajúca technológie protokolu Ethernet, ktorá smeruje rámce podľa ich fyzickej adresy (MAC adresy), využívajúci protokoly IP, ARP a pod. Ethernetová hlavička tiež obsahuje mechanizmus na opravu chýb, časť CRC (Cyklicky redundantný súčet), ktorý slúži na opravu chýb pri chybnom prenose cez sieť [1].



Obr. 1.2: Zloženie Ethernet rámcu štandardu IEEE 802.3

## 1.2 Sieťová vrstva

Najznámejší a najpoužívanější protokol na sieťovej vrstve je protokol IP, ktorý slúži k medzi-sieťovej adresácii zariadení. Hlavička môže mať dva formáty. Jeden pre adresáciu IPv4, ktorá sa používa už dlhšiu dobu. Adresa je 32 bitová, rozdelená na štyri osem bitové segmenty. Pri IPv6 je adresa globálne unikátna, o veľkosti 128 bitov, rozdelená do 8 segmentov, ktorá každá má svoj účel. V hlavičke sú informácie pre nasledujúcu vrstvu o použítom protokole, príznak TTL (Time-to-Live), pri IPv6 sa nazýva Hop Limit, ktorý určuje množstvo prechodom L3 zariadeniami, ktorý sa každým prechodom dekrementuje, keď sa TTL zníži na 0, bude paket zahodený [1].

### Hlavička IP paketu

20B+

Verzia 4b	Dĺžka záhlavia 4b	Typ služby 8b	Celková dĺžka 16b	
Identifikátor datagramu 16b		Priznaky 3b	Posunutie fragmentu od začiatku 13b	
Dĺžka živora paketu 1B		Protokol použitý vo vyššej vrstve 1B	Kontrolný súčet hlavičky paketu 2B	
Zdrojová IP adresa 4B				
Cieľová IP adresa 4B				
Voliteľné data 4B				

Obr. 1.3: Zloženie paketu pri protokole IP

## 1.3 Transportná vrstva

Transportná vrstva poskytuje prenos dát jednotlivým aplikáciám na vyššej vrstve pomocou tzv. portov. Každá aplikácia si môže rezervovať v operačnom systéme port, cez ktorý bude komunikovať v sieti. Pokiaľ aplikácia nevyžaduje čo najrýchlejšie doručenie dát, ale za to správne, využíva sa protokol TCP. Okrem mechanizmu na

kontrolu a prípadné opätovné odoslanie chybných paketov, obsahuje aj mechanizmus na úpravu rýchlosti prenosu alebo aj označovanie paketov, ktoré sú urgentné [1].

### Hlavička TCP paketu 20-60 B

Zdrojový port 2B		Cieľový port 2B	
Sekvenčné číslo (SEQ number) 4B			
Potvrdzovacie číslo (ACK number) 4B			
Data offset 4b	Rezervované 3b	Riadiace bity (ACK, RST, PSH...) 9b	Veľkosť okna 2B
Kontrolný súčet paketu 2B		Urgentnosť paketu 2B	
Doplnkové data paketu 0-40 B			

Obr. 1.4: Schéma TCP paketu

## 1.4 Aplikačná vrstva

Na aplikačnej vrstve pracujú protokoly samotných aplikácií, ktoré používajú koncoví užívatelia. Medzi ne patria protokoly ako SSH (Secure Shell), DNS (Domain Name System), FTP (File Transfer Protocol) a HTTP, ktorý je konkrétnejšie spomenutý nižšie 1.4.1 [1].

### 1.4.1 Hypertext Transfer Protocol

HTTP je internetový protokol, slúžiaci na prenos hypertextových dokumentov vo formáte HTML popřípade XML. Protokol funguje na princípe, že klient pošle výzvu na daný server, server mu odpovie. Medzi najčastejšie typy výzvy protokolu HTTP patria metódy POST, GET, DELETE, CONNECT a ďalšie.

Požiadavok štýlu GET spočíva vo vyžiadaní klienta určitých dokumentov zo serveru na základe URL (Uniform Resource Locator), HTTP verzie, popřípade ešte nepovinných parametrov ako sú napríklad verzia webového prehliadača, podporovaného jazyku, pričom každý jeden atribút je ukončený príznakmi `\r\n`. Samotné ukončenie požiadavky je označené `\r\n\r\n`.

Požiadavok štýlu POST spočíva v posielaní daného textu na webový server na spracovanie, poväčšine obsahu webových formulárov, so samotnou adresou serveru a cesty, kde dané informácie majú byť predané. Rovnako pri výzve štýlu GET, každý jeden atribút je ukončený `\r\n` a koniec označený `\r\n\r\n`. Server na tieto

požiadavky odpovedá správou, ktorá obsahuje aj návratový 3-ciferný kód. Kódy začínajúce číslom:

- 2XX: Úspešné prijatie správy
- 3XX: Presmerovanie klienta
- 4XX: Chyba na strane klienta
- 5XX: Chyba na strane serveru

Server taktiež môže vracat atribúty ako sú napríklad typ serveru alebo dĺžka správy. Ak výzva bola úspešná, nasleduje samotný vyžiadaný webový dokument [2].



## 2 Denial of Service

Denial of service, v preklade odopretie služby, je jedným z kybernetických útokov, ktorý je zacielený na informačný systém či už webová stránka, email alebo online účty ako napríklad internetové bankovníctvo pre legitímnych užívateľov, pre ktorých bude daná služba po určitú dobu nedostupná. Charakteristika útoku spočíva v posielaní špecifických požiadavkov na server, ktorý spôsobí, že samotný server nebude schopný obslúžiť priveľké množstvo požiadavkov naraz a stane sa nedostupným pre ostatných užívateľov. Takéto útoky využívajú zraniteľnosti internetových protokolov [4].

### 2.1 Distributed Denial of Service

Keďže samotné útoky na odopretie služby sú veľmi ľahko odchaliteľné, používajú sa distribuované útoky na dostupnosť. Princíp útokov je, že falošní klienti nepochádzajú z jedného zdroja ale z viacerých, či už z jednej siete alebo z viacerých sietí. Čím viac rôznych zdrojov, tým je útok horšie odhaliteľný, lebo situácia bude pôsobiť, ako iba priveľký počet rôznych legitímnych užívateľov pripájajúcich sa v jeden moment, čo sa v reálnom svete deje na bežnej báze [4].

### 2.2 ARP spoofing

Každý klient sa pri pripojení do lokálnej siete, oznámi v lokálnej sieti pomocou protokolu ARP (Address Resolution Protocol), že sa bude vyskytovať pod danou IP adresou a všetku komunikáciu majú smerovať na jeho MAC adresu. Protokol ARP funguje na vrstve sieťového rozhrania, viď 1.1. Z takýchto záznamov je zostavená smerovacia tabuľka. Každý jeden klient si po každom úspešnom ARP dotaze aktualizuje svoju ARP tabuľku, ktorú si udržuje. Takýto záznam v tabuľke je udržiavaný až 4 hodiny od poslednej komunikácie s danou stanicou, pred jeho vymazaním.

ARP spoofing je technika útoku, kde útočník pošle podvrhnuté pakety do lokálnej siete oznamujúce nových klientov pripojujúcich do siete, zmenu MAC adresy nejakého iného klienta na svoju, čiže všetka komunikácia na klienta bude presmerovaná na útočníka alebo aby sa vydával aj za východziu bránu siete, ktorý následne všetku komunikáciu posielala na východziu bránu. Čiže útočník dostáva všetku komunikáciu smerujúcu zo siete, ktorú môže či už odpočúvať alebo modifikovať. Radí sa medzi tzv. Man-in-the-middle útoky (z angličtiny „človek uprosted“). Proti takýmto útokom sú bežne používané ochranné opatrenia na zamedzenie podvrhovaniu [7].

## 2.3 Záplavové DoS útoky

Prvým a najznámejším druhom DoS útokov je zaplavenie serveru požiadavkami tzv. flood. Ide o posielanie čo najväčšieho množstva paketov v jeden moment, pokiaľ server nezačne pakety zahadzovať z dôvodu, že nebude mať dostatočné prostriedky na ich spracovanie. Takéto útoky sú aj veľmi náročné na šírku pásma útočníka a aj jeho výpočetné schopnosti. Môže sa jednať o zaplavenie pomocou ICMP-echo (ping), ktorý slúži na zistenie či daná adresa existuje a je schopná odpovedať.

Existujú útoky, ktoré zneužívajú protokol TCP, ktorý sa snaží o bezchybovú komunikáciu. Pri naviazaní komunikácie klient so serverom prevedú TCP 3-way handshake. Princíp funguje tak, že klient vyšle TCP paket s príznakom **SYN** (synchronizácia číslovania paketov), na ktorý mu server odpovie paketom príznakmi **SYN**, **ACK**. Následne klient odpovie príznakom **ACK** (potvrdenie o tom, že daný paket bol prijatý), na ktorý nadviaže HTTP GET výzvu, viď. 1.4.1. Akonáhle jedna zo strán neobdrží jeden z paketov, po určitom čase, pošle paket znova s požiadavkou o opätovné poslanie, keďže odpoveď na paket nedorazila. Syn flood zneužíva takéhoto potvrdzovania. Vysiela sa veľké množstvo paketov s príznakom **SYN**, ktoré server spracuje a odpovedá na ne. Lenže klient sa tvári, že odpoveď nedostal a posíla pakety znova. Tento proces sa opakuje až pokiaľ server vyčerpá množstvo poloopených spojení, nebude môcť naviazať ďalšie spojenia, pokiaľ nevyrieši aktuálne a neuvoľnia sa. Pre reálneho klienta sa server tvári ako nedostupný alebo mu odpoveď veľmi dlho trvá [3].

Takéto útoky sú veľmi ľahko vystopovateľné, keďže všetky pakety sú posielané z jednej stanice. Preto sa postupom času začali používať distribuované útoky. Princíp spočíva v rozdelení takýchto požiadaviek medzi veľké množstvo staníc, čiže útok nie je centralizovaný ale rozdistribuovaný medzi stovky klientov, viď 2.1.

Časom ale rástli aj zabezpečenia proti takýmto útokom. Ide v podstate o preteky v zbrojení. Na jednej strane sú zabezpečovacie systémy, na druhej sú útočníci, ktorí sa snažia využiť každú jednu možnosť protokolov [3].

## 2.4 Pomalé DoS útoky

Pomalé DoS útoky sú relatívne novým druhom útokov, ktoré sa vyznačuje malou vyťaženosťou siete. Tento útok simuluje užívateľa s pomalým internetovým pripojením, čiže každú jednu správu posíla s veľkým časovým rozstupom od predchádzajúcej, server čaká na jednotlivé správy až pokiaľ ich klient neukončí alebo spojenie neprehlási za ukončené. Útok je ťažko rozlíšiteľný od bežnej komunikácie, keďže prebieha rovnako ako by postupoval bežný užívateľ, takže ľahšie splynie s normálnou komunikáciou so serverom. Cieľom takýchto útokov je zahltenie webového serveru natolko,

že nebude môcť na ďalšie dotazy odpovedať. Server tieto požiadavky vyhodnocuje až vtedy, keď správa bude kompletne prijatá. Takto server necháva spojenie polo-otvorené. Veľmi dôležité je to, aby útočník neustále komunikoval so serverom do takej miery, aby nevypršali časovače na ukončenie spojenia. Takto útočník vytvorí toľko spojení, že server nebude môcť naviazať viac spojení s reálnymi užívateľmi, dokým neukončí polootvorené. Na rovnakom princípe fungujú aj záplavové DoS útoky, lenže takéto útoky sú oveľa ľahšie odraziteľnejšie, lebo sú od obvyčajnej komunikácie veľmi výrazné. Keďže pomalé DoS útoky sa vyznačujú malou mierou komunikácie, ktoré sú výpočetne nenáročné, môže útočník takéto správy posielat aj z iných zariadení ako len z počítača, napríklad z mobilných zariadení. V poslednej dobe útočníci využívajú napadnuté zariadenia malwarom, ktorý čaká na signál aby všetky stanice začali posielat pakety v jeden moment [3] [6].

### 2.4.1 Slowloris

Slowloris, Slow GET alebo Slow header je útok, ktorý zneužíva HTTP protokol. Konkrétne ukončovanie HTTP GET požiadavku znakmi `\r\n\r\n`. Takéto požiadavky po určitom čase nastavenom na serveri sú zahadzované ako nevalidné, lenže tesne pred ukončením spojenia a vyhodnotením spojenia za mŕtve, útočník pošle tzn. keep-alive paket (správa na udržanie spojenia) o veľmi malej veľkosti so správou o veľkosti niekoľko znakov, napríklad `X-a: b\r\n`, vid. obrázok 2.1, následne útočník opäť čaká čo najdlhšie a proces sa opakuje teoreticky donekonečna. Útočník týmto spôsobom vytvorí veľké množstvo spojení, pokiaľ nebudú vyčerpané všetky zdroje serveru, ktorý nebude schopný odpovedať na ďalšie požiadavky [5]. Na tento typ útoku bol najzraniteľnejší server Apache 1.X , 2.X, avšak od verzie 2.2.14 boli pridané ochranné prvky na obmedzenie tohoto útoku, kde sa znížil časovač na celkové prijatie hlavičky na niekoľko sekúnd.

```
GET /?89018286261135 HTTP/1.1
Host: bbc.com
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR
1.1.4322; .NET CLR 2.0.50313; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729; MSOffice 12)
Content-Length: 42
X-a: b
X-a: b
X-a: b
X-a: b
```

Obr. 2.1: Príklad výslednej hlavičky pre SlowLoris



a spracovať. Veľa krát je tento parameter určený dostupnosťou siete alebo umiestnením klienta od výstupného bodu zo siete. Útočník nastaví tento parameter na veľmi malú hodnotu, často do desiatok, čo reprezentuje maximálny počet bytov, ktoré môžu obsahovať dáta odpovede [8]. Napríklad pri požiadaní o obrázok ktorý bude mať 1 MB a `window-size` by bol 10, by musel server poslať minimálne 100 000 správ, aby klient obdržal celý obrázok. Pri odmlčaní na 3 sekundy medzi každou správou by prenos trval skoro 6 dní.

## 3 Webové servery a ich zabezpečenie proti DoS útokom

Pomalé DoS útoky neprešli bez povšimnutia samotným autorom webových serverov, ktorí vytvárajú ochranné mechanizmy na sťaženie, poprípade úplne znemožnenie útoku.

### 3.1 Zabezpečenie serveru Apache2

K jedným z najpoužívanejších webových serverov na svete patrí Apache, ktorý je open-source a podľa údajov z apríla 2019 tvorí Apache viac ako 26% všetkých serverov na svete [9]. Pri samotnej konfigurácii serveru sa ponúka políčko `RequestReadTimeout`. Jedná sa o časovač, ako dlho môže klient posilať požiadavok na server, dokým ho server nezruší z dôvodu príliš dlhého časového okna. Pôvodná hodnota je nastavená na 20, čo sa jedná o 20 sekúnd, kedy klient môže poslať hlavičku požiadavky na server. Toto nastavenie ovplyvňuje hlavne útok Slowloris, ktorý sa snaží vytvoriť čo najviac spojení a udržať ich po čo najdlhší čas. Ďalšie podstatné nastavenie je `KeepAliveTimeout`, ktoré ovplyvňuje čas, za ktorý musí klient poslať celé telo požiadavky, inak bude spojenie zamietnuté a zrušené. Nastavenia ako `LimitRequestBody`, `LimitRequestFields`, `LimitRequestFieldSize`, `LimitRequestLine` a `LimitXMLRequestBody` slúžia ako dodatočné nastavenia webového serveru, ktoré znemožnia iné druhy útokov, hlavne ich obmedzením. Tieto moduly na ochranu proti DoS útokom sú od verzie 2.3.14 predinštalované na webovom serveri [11]. Takéto nastavenia sú ale stále iba čisto na obmedzenie jednotlivých požiadaviek, čiže útočník môže dôsledným kontrolovaním zistiť tieto obmedzenia a aj napriek ich nastaveniu ich môže obísť, ako sa jedná u vlastnej implementácie, vid. kapitola 5.

### 3.2 Zabezpečenie serveru Nginx

Druhým najpopulárnejším webovým serverom používaným vo svete je Nginx. Na rozdiel od Apache2 je Nginx viac pamäťovo efektívnejší na úkor flexibility a konfigurovateľnosti. Webový server je opäť open-source. Spoločnosť Nginx tiež vytvorila dodatočné balíky a moduly na konfiguráciu, ktoré sú ale platené [12]. Taktiež ako Apache2, Nginx má v sebe predinštalovaný modul na zmiernenie DoS útokov. Taktiež poskytuje dodatočné moduly na rozloženie vyťaženia, ale tieto možnosti sú iba v platenej verzii. Podobne ako pri webovom serveri Apache2, existujú už priamo

vstavané moduly na spomalenie DoS útokov. Podobne ako pri Apache2, je možné nastaviť modul proti DoS útokom `client_body_timeout` a `client_header_timeout`, ktoré slúžia na ukončenie pomalých spojení. Prednastavená hodnota je 60 sekúnd na obe nastavenia. Tieto nastavenia sú cielené priamo proti pomalým útokom ako je Slowloris, Slow Post a Slow Read. Ďalším významným modulom je `limit_req_zone`, čo znamená, koľko požiadavkov môže urobiť jeden užívateľ behom jednej sekundy. Referenčná hodnota pre toto nastavenie je 30 požiadavkov za minútu, alebo jeden požiadavok za 2 sekundy [13]. Posledným významným modulom je `limit_conn_zone`, ktorý slúži na obmedzenie počtu spojení z jednej IP adresy. V prípade tohoto vytvoreného generátora, viď 5, sa jedná o efektívne a devastujúce opatrenie, pokiaľ sa útočník nenachádza v lokálnej sieti, kde môže použiť ARP spoofing, viď 2.2. Ale tento modul je absolútne neefektívny ak sa jedná o útok typu DDoS, viď 2.1, keďže spojenia sa naväzujú z rôznych IP adries.

Existujú aj nastavenia, ktoré sa ale musia nastavovať ručne ako napríklad zakazovanie komunikácie z určitých IP adries alebo ich povolenie.

### 3.3 Zabezpečenie serveru lighttpd

Posledným z populárnych serverov je lighttpd. Je určený prevažne na infraštruktúry, kde je vyžadovaná rýchlosť. Originálne bol lighttpd vytvorený ako dôkaz konceptu (z angličtiny „proof of concept“), že je možné pripojenie 10 000 klientov na jednom serveri paralelne [14]. Opäť je tento webový server open-source. Rovnako ako pri predošlých webových serveroch, tiež obsahuje predinštalovaný základný modul na limitovanie množstva toku dát na server pre všetky alebo iba pre jedno spojenie. Neobsahuje však žiadne zložitejšie nastavenia [15].

### 3.4 Dodatočná ochrana

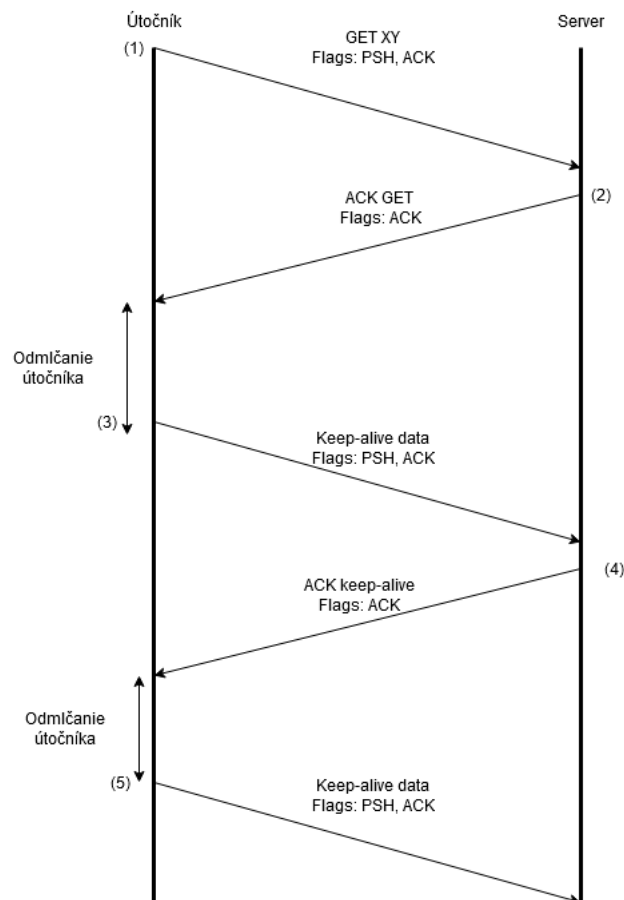
Takéto ochranné mechanizmy nie sú dostačujúce samé na ochranu serveru, preto sú v poslednej dobe populárne softwarové riešenia tretích strán (Third-party software), ktoré sú situované pred serverom, čiže všetka komunikácia ide cez ne. Vybavené veľkým množstvom pravidiel a umelou inteligenciou sa snažia zabrániť hocijakému pokusu o útok. Existuje veľa rôznych mechanizmov na odvrátenie alebo zťaženie útoku či už vyvažovanie záťaže medzi viac serverov, firewall alebo iné.

Odhaľovanie DDoS útokov (Distributed Denial of service) je oveľa zložitejšia záležitosť a to kvôli tomu, že útoky putujú na server z veľa rôznych zdrojov. Takéto útoky sú oveľa zložitejšie na odhalenie, keďže sa tvária ako normálni užívatelia z celého sveta žiadajúci server.

## 4 Modely útokov

### 4.1 Slowloris

Po úspešnom nadviazaní TCP spojenia na server, čaká na ďalšie pakety s informáciami o požiadavku, ako bolo spomenuté v kapitole 2.4.1. Útok bude prebiehať poslaním hlavného požiadavku v jednom pakete, ale nebude ukončený, čiže server bude čakať na jeho ukončenie, ktoré ale nikdy neskončí. Čiže útočník bude posielat náhodné udržiavacie informácie, ktoré budú dostatočné na to, aby server čakal na ďalšie [6].



Obr. 4.1: Diagram útoku Slowloris

Prvý paket, viď obr. 4.1 (1) obsahuje iba tie najnutnejšie informácie pre server, aby ho vyhodnotil ako validný, aby ho hneď pri prijatí nezahodil a spojenie nezrušil. Tieto informácie sú samotný typ požiadavku, v tomto prípade **GET** nasledujú náhodný reťazec znakov, automaticky generovaný reťazec 15 náhodných čísel, za ním verzia **HTTP** protokolu. Nasleduje odriadkovanie znakmi `\r\n`. Nasleduje políčko **Host**, opäť odriadkovanie, viď obr. 4.2. Server tento paket prijme (2), útočník sa



odmlčí na čo najdlhšiu dobu, tento parameter je veľmi subjektívny, keďže každý jeden webový server je nastavený na inú hodnotu, pre tento prípad je to iba 5 sekúnd, pošle udržiavací paket s reťazcom znakov **X-a: b (3)**, viď obr. 4.3, server tento paket prijme (4), čaká na ďalšie dáta. Opäť sa klient odmlčí na 5 sekúnd, následne opäť pošle udržiavací reťazec **X-a: b**. Keďže požiadavok nebol odriadkovaný dvakrát, server ho stále nevyhodnocuje až pokiaľ nebude ukončený. Takéto spojenia sú vytvorené podľa požiadavok útoku. Keďže všetci klienti v generátore sú poukladaní do veľkého zoznamu, oneskorenie medzi paketmi od rôznych klientov je ideálne nastavenie na 10 milisekúnd pre 500 klientov, pre 1 000 klientov 5 milisekúnd a pre 1500 a viac, 2 milisekundy, aby sa útok nejavil až tak moc záplavový.

```

00 00 00 01 00 06 e0 d5 5e c5 e7 48 00 00 08 00 ..... ^..H....
45 00 00 72 01 46 40 00 80 06 75 22 c0 a8 01 65 E..r.F@. .u"....e
c0 a8 01 68 13 88 00 50 20 01 92 fc 1a 4d 5a ac ...h...P ...MZ.
50 18 00 0f dc d3 00 00 47 45 54 20 2f 3f 36 35 P..... GET /?65
34 38 36 35 32 34 31 35 36 32 34 35 36 20 48 54 48652415 62456 HT
54 50 2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 31 39 TP/1.1.. Host: 19
32 2e 31 36 38 2e 31 2e 31 30 34 20 0d 0a 43 6f 2.168.1. 104 ..Co
6e 74 65 6e 74 2d 4c 65 6e 67 74 68 3a 20 34 32 ntent-Le ngth: 42
0d 0a ..

```

Obr. 4.2: Príklad hlavičky útoku Slowloris (zvýraznené sú samotné dáta hlavičky)

```

00 00 00 01 00 06 e0 d5 5e c5 e7 48 00 00 08 00 ..... ^..H....
45 00 00 30 01 47 40 00 80 06 75 63 c0 a8 01 65 E..0.G@. .uc....e
c0 a8 01 68 13 88 00 50 20 01 93 46 1a 4d 5a ac ...h...P ...F.MZ.
50 18 00 de 07 dc 00 00 58 2d 61 3a 20 62 0d 0a P..... X-a: b..

```

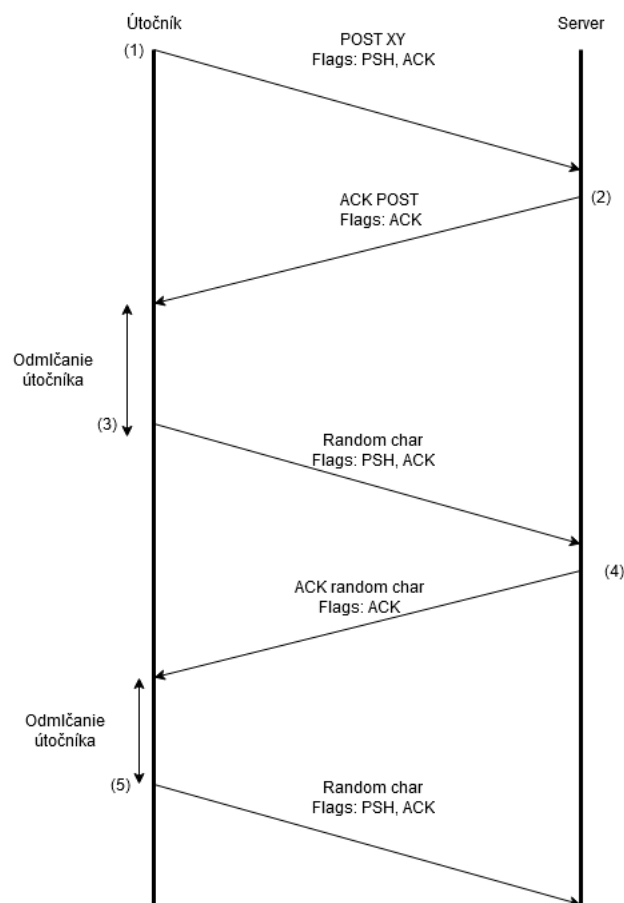
Obr. 4.3: Príklad hlavičky útoku Slowloris (zvýraznené sú udržiavacie dáta)

## 4.2 Slow Post

Slow Post útok je v základe dosť podobný útoku Slowloris, rozdiel medzi nimi je, že Slow Post posiela dáta po ukončení hlavičky s požiadavkom POST, narozdiel od útoku Slowloris, ktorý hlavičku ani nedokončí s požiadavkom GET. Slow POST sa používa oveľa častejšie a to preto, lebo požiadavok POST je viac flexibilný a veľa vecí má dovolenejších ako požiadavok GET [6].

Ako aj pri predchádzajúcom útoku, opäť je treba najskôr TCP spojenie so serverom. Pošle sa validná hlavička s požiadavkom POST, čo znamená že chceme vkladať údaje na server s cestou či už validnou alebo nie, napríklad **index.php (1)**. Podstatný je parameter **Content-Length**, ktorý je nastavený na hodnotu 1 000 000,

ktorá je veľká hodnota a nebude vyčerpaná, viď obr. 4.5. Keďže v paketoch patrí 1 byte jednému znaku, znamenalo by to, že by sa muselo preniesť 1 000 000 udržiavacích paketov, čo by pri dĺžke odmlčania 5 sekúnd znamenalo, že komunikácia by prebiehala cez 9 rokov. Pri zvolení hodnoty príliš nízkej môže dôjsť k stavu, kedy bude daná dĺžka vyčerpaná udržiavacími paketmi. Server by následne tento požiadavok vyhodnotil a spojenie ukončil. Toto okno by bolo šancou pre legitímneho klienta na pripojenie, čiže by sa jednalo o zneefektívnenie samotného útoku. Tento paket server potvrdí (2) a čaká na samotné dáta. Útočník opäť čaká 5 sekúnd. Útočník generuje náhodné dáta, ktoré sú iba jedno písmeno, či už malé alebo veľké, čiže od **a** po **Z** (3), viď obr. 4.6. Medzi správami sa server odmlčí na 5 sekúnd. Takto je server zaneprázdnený čakaním na útočníka, že nemôže reagovať na ostatných.



Obr. 4.4: Diagram útoku Slow Post

Najmenší paket je podobný tomu pri útoku Slowloris, keďže podstatné položky sú typ požiadavku, cesta k nemu, HTTP verzia, Host a Content-Length, viď 4.5.

```

00 00 00 01 00 06 e0 d5 5e c5 e7 48 00 00 08 00 ..... ^..H....
45 00 00 75 00 c9 40 00 80 06 75 9c c0 a8 01 65 E..u..@. .u...e
c0 a8 01 68 13 88 00 50 0e 54 c8 d5 b8 87 e8 5c ...h...P .T....\
50 18 00 0f 48 18 00 00 50 4f 53 54 20 2f 74 65 P..H... POST /te
78 74 66 6f 72 6d 2e 70 68 70 20 48 54 54 50 2f xtform.p hp HTTP/
31 2e 31 0d 0a 48 6f 73 74 3a 20 31 39 32 2e 31 1.1..Host: 192.1
36 38 2e 31 2e 31 30 34 0d 0a 43 6f 6e 74 65 6e 68.1.104 ..Conten
74 2d 4c 65 6e 67 74 68 3a 20 31 30 30 30 30 30 t-Length : 100000
30 0d 0a 0d 0a 0.....

```

Obr. 4.5: Príklad hlavičky útoku Slow Post (zvýraznené sú samotné dáta hlavičky)

```

00 00 00 01 00 06 e0 d5 5e c5 e7 48 00 00 08 00 ..... ^..H....
45 00 00 29 00 ca 40 00 80 06 75 e7 c0 a8 01 65 E..)..@. .u...e
c0 a8 01 68 13 88 00 50 0e 54 c9 23 b8 87 e8 5c ...h...P .T.#...\
50 18 00 de 37 9b 00 00 67 00 00 00 00 00 P...7... g.....

```

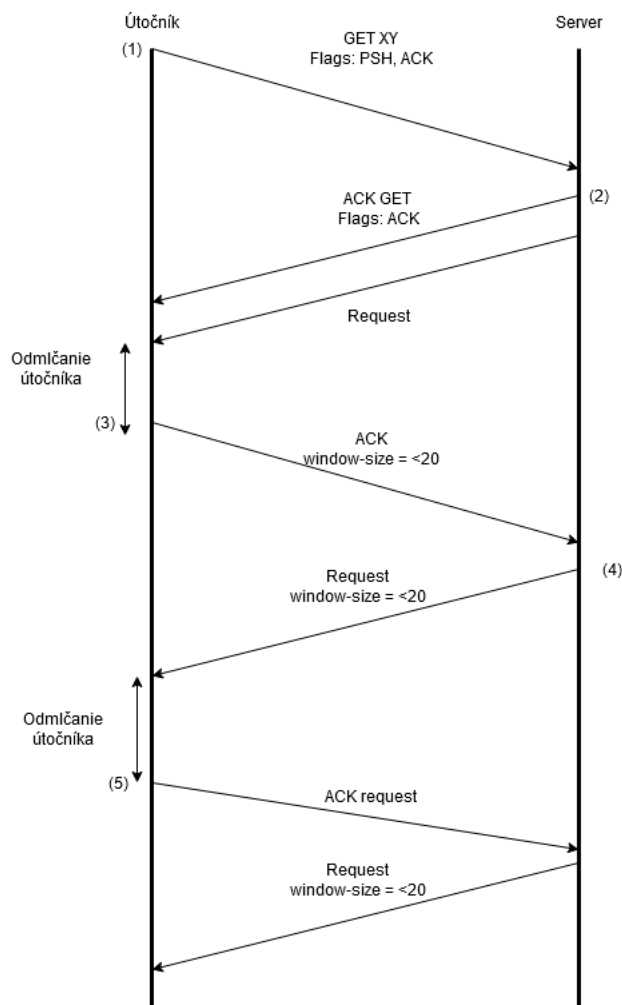
Obr. 4.6: Príklad udržiavacieho paketu útoku Slow Post (zvýraznené je znak (dáta) udržiavacieho paketu)

### 4.3 Slow Read

Posledným útokom je Slow Read. Opäť sa musí v prvom rade naviazať TCP spojenie so serverom, v ktorom sa nastaví parameter **window-size** na hodnotu 10, čo je dostatočne malá hodnota, aby spojenie pretrvalo dostatočne dlhú dobu. Požiadavok GET na obdržanie obsahu je odoslaný naraz, keďže je veľmi malý. Ako aj pri útoku Slowloris, viď obr. 4.2, základné parametre je samotný požiadavok s cestou k obsahu, ktorá je náhodne generovaná z 15 čísel, verzia HTTP 1.1, Host. Takýto požiadavok je ukončený, poslaný na server (1). Ten ho prijme a pošle prvý paket (2). Do paketu sa zmestí iba 10 bytov dát. Klient ich prijíma ešte s 5 sekundovým oneskorením (5). Takáto komunikácia je síce pomalá, ale javí sa ako validná, keďže takéto situácie sa stávajú pri veľmi pomalom pripojení. Takýto prenos dát je okolo 10 bytov za 5 sekúnd, čo sú 2 byty dát za sekundu. Menšie hodnoty môžu zapríčiniť, že server toto spojenie automaticky zamietne, z dôvodu príliš malého okna, čo by znamenalo priveľmi dlhú komunikáciu [6].

### 4.4 DDoS

Jednou z pridaných variant útokov je aj DDoS variata, ktorá simuluje oveľa viac klientov v lokálnej sieti. K takémuto účelu sa používa ARP spoofing, viď sekcia 2.2. V jednoduchosti pomocou ARP protokolu oznámi niekoľko IP adries, ktoré sa javia ako reálni klienti, ale ich MAC adresy smerujú na útočnickovu MAC adresu. Takže



Obr. 4.7: Diagram útoku Slow Read

v ARP tabuľke sú určité IP adresy smerujúce na tú istú MAC adresu. Jedna zo staníc sa opýta, že takú adresu nepozná, nech jej niekto povie akú MAC adresu má daná IP adresa. Útočník jej následne nato odpovie, keďže žiadna zo staníc nevie akú MAC adresu má. Teraz si každá stanica v danej sieti zapíše spoofnuté adresy. Celý proces ARP spoofing útoku je zobrazený na obrázku 4.9.

Modelová situácia:

- Útočníkova IP adresa: 1.1.1.1
- Klientska IP adresa: 1.1.1.2

V prvom kroku útočník vyvolá nejakú akciu, ktorá bude smerovať na neexistujúcu IP adresu 1.1.1.3 (1). Klient zistí, že danú IP adresu nemá vo svojej tabuľke a dotazuje sa pomocou ARP, nech mu niekto prezradí MAC adresu (2). Útočník mu odpovie na dotaz so svojou MAC adresou (3). Klient si aktualizuje svoju ARP tabuľku. Tento proces sa opakuje toľkokrát, koľko bude falošných IP adries. Bohužiaľ, ARP spoofing je efektívny iba v lokálnej sieti a to iba za podmienok, že v sieti nie sú

```

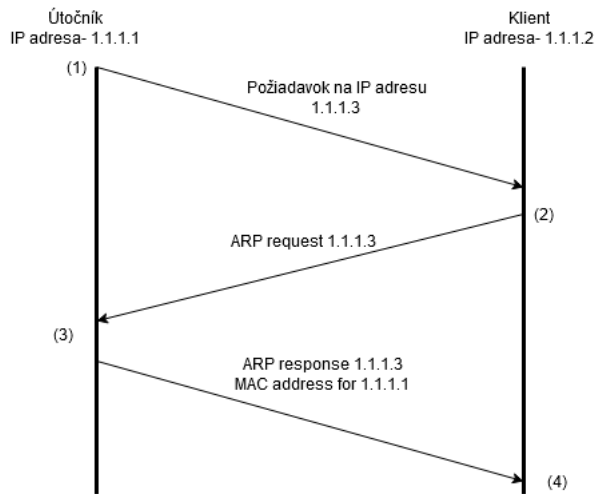
GET /index.html HTTP/1.1
Host: 192.168.1.104

HTTP/1.1 200 OK
Date: Tue, 05

```

Obr. 4.8: Príklad najmenšieho možného požiadavku na útok Slow Read

ochranné mechanizmy na obranu proti ARP spoofing-u. V lokálnej sieti kvôli tomu, že akonáhle paket opustí lokálnu sieť, východzia brána siete prepisuje MAC adresy na svoje.



Obr. 4.9: Diagram útoku ARP spoofing

Takto vyzerá ARP tabuľka, viď obr. 4.10, kde klient s IP adresou **192.168.0.254** je jeden z pravých klientov, ostatné IP adresy sú smerované na tú istú MAC adresu.

192.168.0.254	ether	e4:8d:8c:ba:96:ff	C	enp0s3
192.168.0.205	ether	e0:d5:5e:c5:e7:48	C	enp0s3
192.168.0.234	ether	e0:d5:5e:c5:e7:48	C	enp0s3
192.168.0.210	ether	e0:d5:5e:c5:e7:48	C	enp0s3
192.168.0.106	ether	e0:d5:5e:c5:e7:48	C	enp0s3
192.168.0.202	ether	e0:d5:5e:c5:e7:48	C	enp0s3
192.168.0.235	ether	e0:d5:5e:c5:e7:48	C	enp0s3

Obr. 4.10: Príklad výpisu ARP tabuľky po ARP spoofingu

## 5 Praktická implementácia - Packet cannon

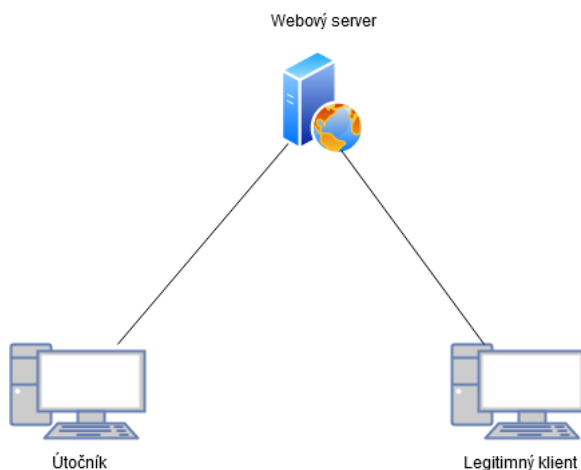
Podľa zadania bakalárskej práce bol vytvorený generátor na Denial of service útoky, konkrétne na Slowloris, Slow Post, Slow Read, ktorý by dokázal obísť ochranné mechanizmy a znemožniť, či poprípade sťažiť prístup na webový server. Pre použitie generátora je potrebné, aby bol nainštalovaný **WinPcap**, normálne je súčasťou software-u **Wireshark**. Slúži na prijímanie a analýzu paketov a ich odosielanie. V tejto kapitole sa bude rozoberať samotný model, či už implementácia, poprípade príklad samotného využitia.

### 5.1 Vývojové prostredie

Vývojové prostredie sa skladalo z 3 virtuálnych staníc:

- **Webový server:** Ubuntu – Spustený webový server na testovanie
- **Útočník:** Windows 10 – taktiež aj hositeľ jednotlivých virtuálnych staníc
- **Klient:** Ubuntu – Nič netušiaci klient, snažiaci sa pripojiť na server

Všetky virtuálne stroje majú sieťové adaptéry nastavené na mód "Bridge adapter", ktorý vytvorí most medzi sieťovými adaptéromi hostiteľa a jednotlivých virtuálnych strojov, čiže jednotlivé virtuálne stanice sa tvária ako fyzické stanice v sieti oddelené od seba. Nasledujúci obrázok popisuje zapojenie virtuálnych strojov.



Obr. 5.1: Sieťové zapojenie jednotlivých strojov

#### 5.1.1 Vývoj generátora

Vývoj generátora prebiehal v prostredí Visual Studio 2017, v jazyku C# s využitím Nuggetu NPcap.net, čo sa jedná o Wrapper, ktorý využíva C++ knižnicu libPcap,

aby bola použiteľná aj v prostredí C#. Na grafické užívateľské prostredie bol použitý grafický subsystém od Microsoft-u **WPF**. Navrhnutý generátor využíva všetky 3 druhy DoS útokov, vid. kapitola 2, podľa toho čo si útočník vyberie. Základom všetkých útokov je simulácia klienta, o čo sa stará trieda `DosSender`, ktorá obsahuje metódy, z ktorých sa skladajú samotné vrstvy sieťovej komunikácie a posielajú priamo na cieľ, vid. kapitola 1.

Výpis 5.1: Príklad skladania jednotlivých vrstiev

```
1 // Ethernet vrstva
2     EthernetLayer ethernetLayer = new EthernetLayer
3     {
4         Source = SourceMac,
5         Destination = DestinationMac,
6     };
7     // IPv4 vrstva
8     IPv4Layer ipv4Layer = new IPv4Layer
9     {
10        Source = SourceIPv4,
11        CurrentDestination = DestinationIPv4,
12        Ttl = 128,
13        Fragmentation = new IPv4Fragmentation
14            (IPv4FragmentationOptions.DoNotFragment, 0),
15        Identification = _identificationNumber,
16    };
17
18    // TCP vrstva
19    TcpLayer tcpLayer = new TcpLayer
20    {
21        SourcePort = SourcePort,
22        DestinationPort = DestinationPort,
23        SequenceNumber = SeqNumber,
24        ControlBits = TcpControlBits.Synchronize,
25        Window = WindowSize,
26    };
27    communicator.SendPacket(PacketBuilder.Build(DateTime.Now,
28        ethernetLayer, ipv4Layer, tcpLayer));
```

Po zložení jednotlivých vrstiev je paket odoslaný vybraným sieťovým rozhraním ku cieľu. Takto sa realizuje pre každé jedno spojenie spomenutý TCP handshake v sekcii 2.3 zvolený útok.

Ako je vidno na výpise 5.1, pre jednotlivé pakety je potrebné zistiť MAC adresu daného zariadenia, ktoré sa nachádza v sieti. MAC adresa je unikátny identifikátor daného zariadenia v sieti, je pevne stanovená výrobcom. Je zapísaná ako kombinácia 12 hexadecimálnych čísiel, rozdelených po pároch a od seba oddelený buď '-' alebo ':'. Príklad MAC adresy jedného zo zariadení: E0:D5:5E:C5:E7:48. Keďže MAC adresa sa používa iba v rámci siete, mimo siete sa používa IP adresa. Do políčka sa zadáva IP adresa daného webového serveru, následne pomocou knižnice `iphlpapi.dll` sa pomocou protokolu ARP, ktorým si stanice v sieti vymieňajú sieťové tabuľky so všetkými známymi MAC adresami, preloží daná IP adresa na MAC adresu stanice v sieti. Ak sa stane, že dané zariadenie neexistuje v danej sieti, nastaví sa MAC adresa východzej brány zo siete, aby sa vedelo, kde má byť paket smerovaný, tzn. von zo siete.

Nad jednotlivými falošnými klientmi je Controller, ktorý ovláda jednotlivých klientov v tom význame, ktorý paket majú poslať ako ďalší. Osobitné vlákno sa stará o prichodzie pakety, podľa toho, ktorému klientovi patria, podľa cieľového portu.

Falošní klienti sú posielaní po vlnách, medzi ktorými je čas, ktorý sa tvária ako nedostupní. Avšak ešte je oneskorenie pri posielaní medzi jednotlivými klientmi, aby sa útok tváril viac náhodný a nie záplavový. Pri konfigurácii dodatočných nastavení, viď. sekcia 5.1.2, keďže sa môže stať, že jednotlivé oneskorenia budú tak veľké, že server začne zahadzovať spojenia, ku ktorým sa útočník nemohol ani len dostať, čiže útok bude neefektívny.

## 5.1.2 Grafické užívateľské prostredie

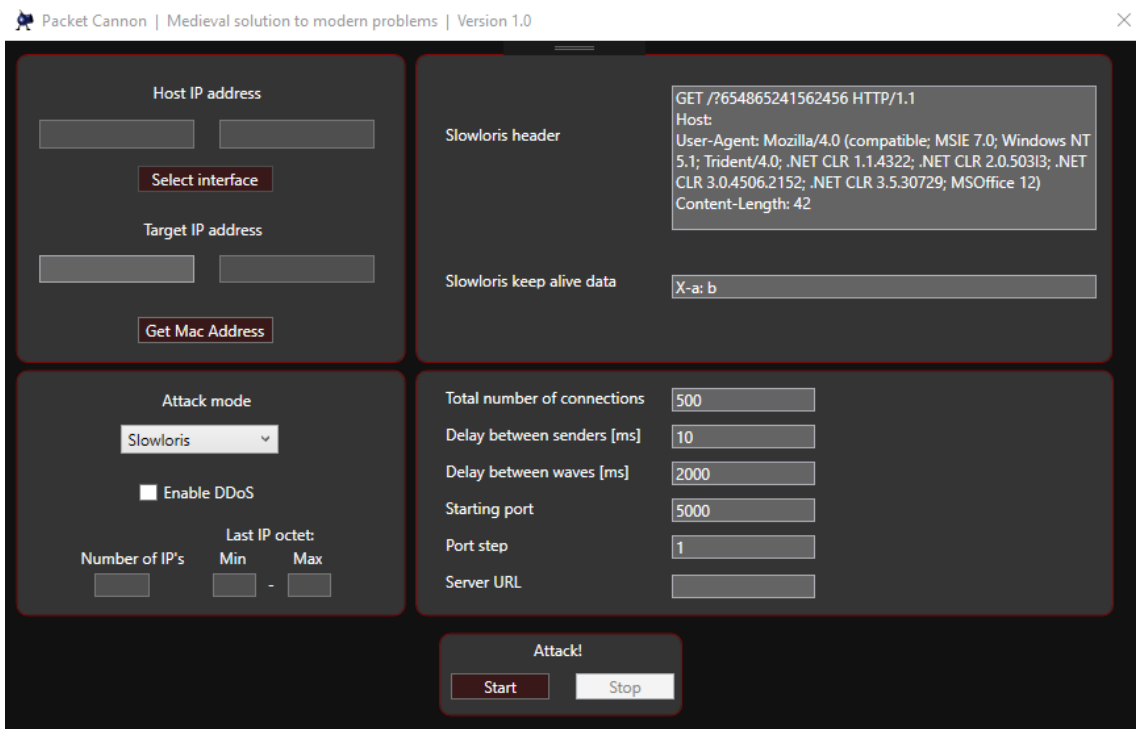
Grafické užívateľské prostredie je vytvorené na subsystéme WPF, ktorý je priamo vytvorený pre programovací jazyk C#. Pri spustení vyskočí okno základnej konfigurácie, viď obr. 5.2.

Do pravého políčka **Target IP address** sa zadáva IP adresa cieľa, následne sa pomocou tlačítka **Get Mac Address** preloží IP adresa na MAC adresu daného zariadenia, pokiaľ sa nachádza v lokálnej sieti pomocou protokolu ARP. Pokiaľ sa dané zariadenie nenachádza v lokálnej sieti, MAC adresa cieľa sa nastaví ako východzia brána zo siete von. Pomocou rolovacej rolety **Attack mode** je možné si vybrať z 3 útokov:

1. Slowloris
2. Slow Port
3. Slow Read

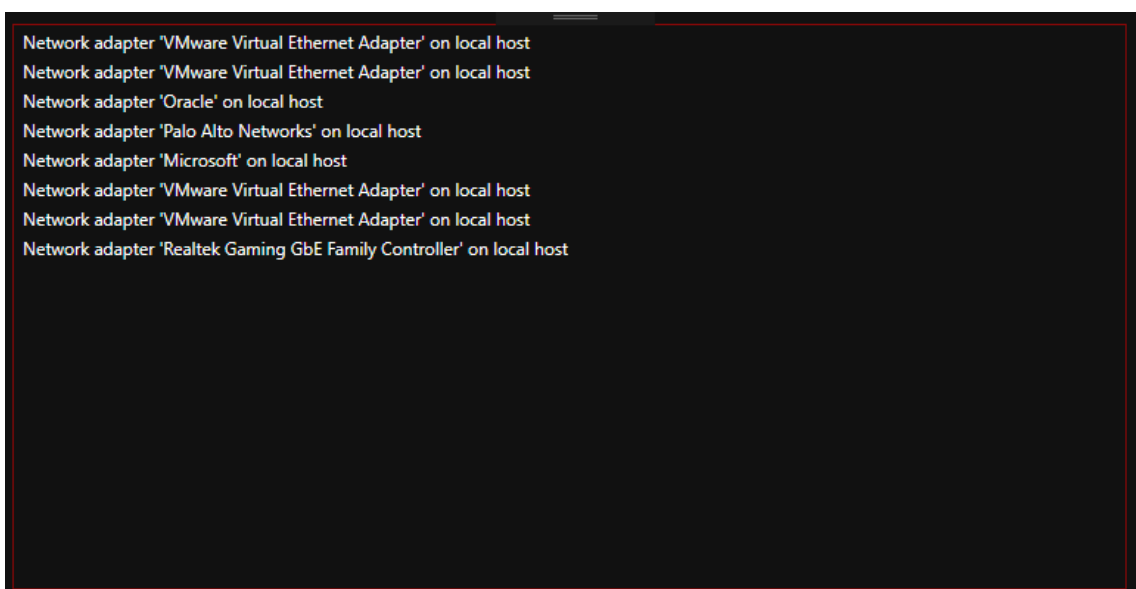
Tlačítka **Start** a **Stop** kompletne resetujú celý generátor, takže zakaždým, keď sa stlačí tlačítko **Start**, pôjde útok od začiatku.





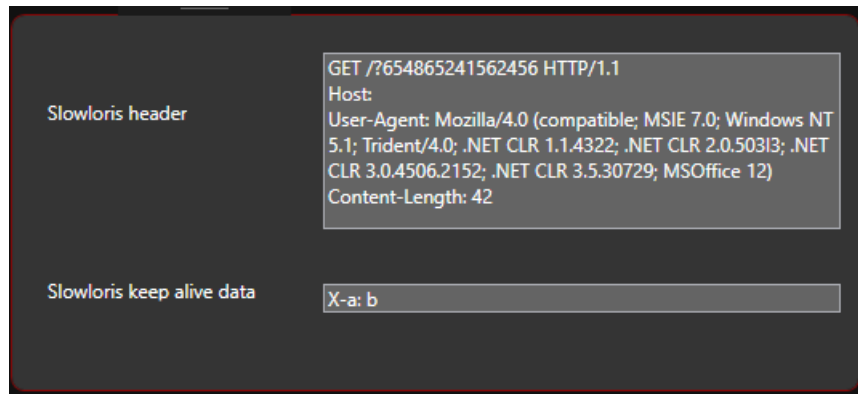
Obr. 5.2: Konfiguračné okno pre generátor

Pri kliknutí na tlačítko **Select interface**, sa otvorí dodatočné okno s vybraním sieťového adaptéru, ktorý bude použitý pri útoku, vid. obrázok 5.3. Treba si dávať obzvlášť pozor na to, ktorý adaptér je fyzický a ktorý je virtuálny, keďže cez virtuálny adaptér sa nedá dostať mimo virtuálnej siete.



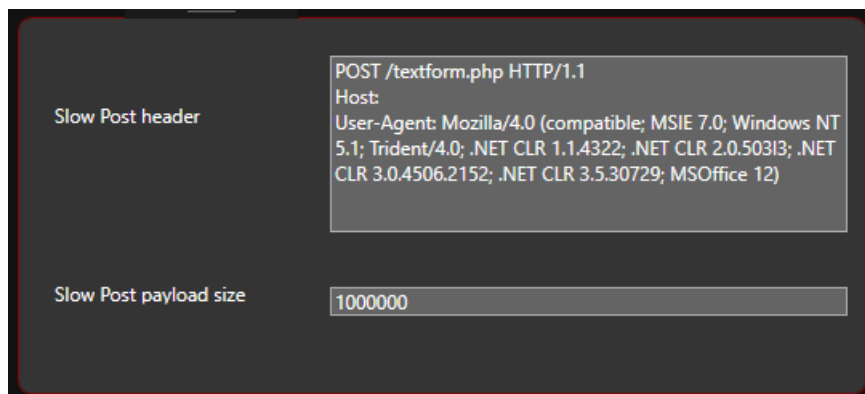
Obr. 5.3: Výber sieťového adaptéru

Na pravej strane užívateľského prostredia sa nachádzajú parametre útoku, ktorý je aktuálne zvolený a už predvyplnené parametre na základnú hodnotu.



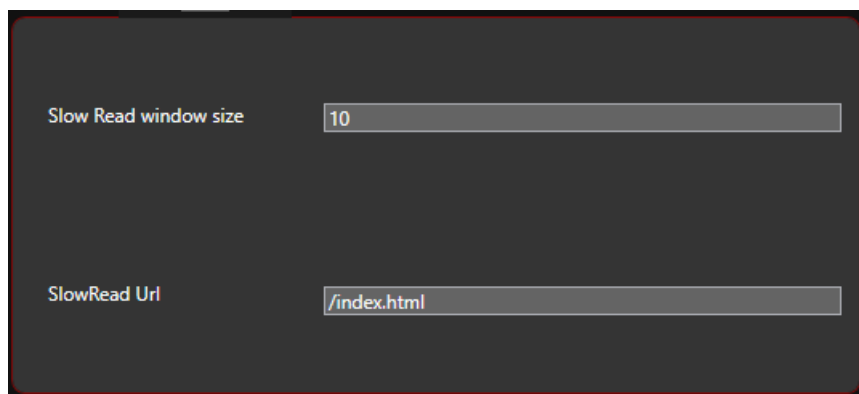
The screenshot shows the configuration interface for a Slowloris attack. It features two main sections: 'Slowloris header' and 'Slowloris keep alive data'. The 'Slowloris header' section contains a text box with the following content: 'GET /?654865241562456 HTTP/1.1', 'Host:', 'User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 1.1.4322; .NET CLR 2.0.50313; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729; MSOffice 12)', and 'Content-Length: 42'. The 'Slowloris keep alive data' section contains a text box with the value 'X-a: b'.

Obr. 5.4: Základné nastavenia pre útok Slowloris



The screenshot shows the configuration interface for a Slow Post attack. It features two main sections: 'Slow Post header' and 'Slow Post payload size'. The 'Slow Post header' section contains a text box with the following content: 'POST /textform.php HTTP/1.1', 'Host:', 'User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 1.1.4322; .NET CLR 2.0.50313; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729; MSOffice 12)'. The 'Slow Post payload size' section contains a text box with the value '1000000'.

Obr. 5.5: Základné nastavenia pre útok Slow Post



The screenshot shows the configuration interface for a Slow Read attack. It features two main sections: 'Slow Read window size' and 'SlowRead Url'. The 'Slow Read window size' section contains a text box with the value '10'. The 'SlowRead Url' section contains a text box with the value '/index.html'.

Obr. 5.6: Základné nastavenia pre útok Slow Read

Total number of connections	<input type="text" value="500"/>
Delay between senders [ms]	<input type="text" value="10"/>
Delay between waves [ms]	<input type="text" value="2000"/>
Starting port	<input type="text" value="5000"/>
Port step	<input type="text" value="1"/>
Server URL	<input type="text"/>

Obr. 5.7: Základné nastavenia pre falošných klientov

Popis ku jednotlivým možnostiam:

- **Slowloris header** – samotná hlavička Slowloris útoku (predvolená hodnota je vid. obrázok 5.4).
- **Slowloris keep alive data** – Vlastné dáta pre udržiavacie pakety (predvolená hodnota je X-a: b).
- **Slow Post header** – hlavička Slow Post útoku (predvolená hodnota je vid. obrázok 5.5).
- **Slow Post payload size** – celková veľkosť dát, ktoré budú posielané, mala by byť čo najvyššia, aby bol útok efektívny (predvolená hodnota je 1 000 000).
- **Slow Read window size** – veľkosť okna pri čítaní (predvolená hodnota je 10 bytov).
- **Slow Read url** – keďže Slow Read útok je efektívnejší na webových stránkach, ktoré sú väčšie ako 1 MB, slúži na vybranie cesty k napríklad obrázku (predvolená hodnota je vid. obrázok 5.6).
- **Total number of connections** – Počet falošných klientov snažiaci sa pripojiť na webový server (predvolená hodnota je 500 klientov).
- **Delay between Senders** – čas medzi odpoveďami od falošných klientov (predvolená hodnota je 10 milisekúnd).
- **Delay between waves** – keďže klienti odpovedajú takmer v jeden moment, tento rozstup slúži na simulovanie nedostupnosti klientov, keby sa všetci klienti tvárili ako nedostupní (predvolená hodnota sú 2 sekundy).
- **Starting port** – začiatkový port, cez ktoré budú komunikovať falošný klient, keďže niektoré dôležité aplikácie na systéme, môžu používať port viac ako je predvolená hodnota, v tomto prípade je predvolená hodnota na 5000.
- **Port step** – hodnota, o ktorú sa inkrementuje port predchádzajúceho falošného klienta, čiže ak prvý klient komunikuje cez port 5000 a Port step je nastavený na hodnotu 5, nasledujúci klient bude komunikovať cez port 5005,

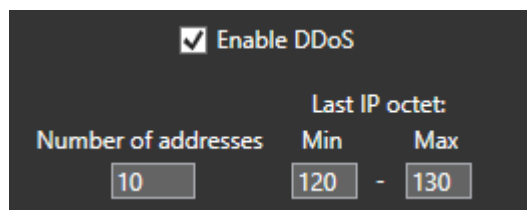
další 5010 atď. (predvolená hodnota je 1).

- **Server URL** – slúži k tomu, aby bolo možné sa dostať aj na stránky, ktoré nie sú definované iba IP adresou ale majú aj tzv. Alias, čiže ich IP adresa je preložená do názvu stránky, ktoré sú uložené v záznamoch DNS serveru, keď sa budú jednotlivé stanice na ne dotazovať, napríklad pri doméne `www.google.com` sa jeho adresa prekladá na `172.217.23.238` a na túto adresu budú smerované jednotlivé pakety.

## 5.2 DDoS

Poslednou z pridaných funkcií, je možnosť DDoS útoku, ktorá je ale efektívna iba v lokálnej sieti, ktorá nemá ARP spoofing ochranu. ARP spoofing je vytvorený poslaním požiadavku **ICMP-echo**, ľudovo povedané "ping", na dané neexistujúce adresy. Celý proces je popísaný v sekcii 4.4. Takéto adresy sú následne použité ako falošní užívatelia.

Po zaškrtnutí políčka **Enable DDoS**, viď obr. 5.8, sa aktivujú políčka vedľa neho. Prvé políčko **Number of addresses** určuje, koľko falošných adries sa má vytvoriť. Ďalšie dve políčka slúžia na zadanie minimálnej a maximálnej koncovú posledného oktetu adresy. Adresy sú vytvorené z IP adresy vybraného sieťového rozhrania. Napríklad, keby máme adresu sieťového rozhrania `1.1.1.1` a chceme vytvoriť 5 adries medzi 10 a 20, vytvorené adresy budú od adresy **1.1.1.10 až po 1.1.1.20** vybrané náhodne. Klienti na nich sú tiež vyberaní náhodne, tzn. z jednej adresy môže byť viac klientov ako z druhej, poprípade určité adresy nebudú použité vôbec.



Obr. 5.8: Nastavenia DDoS

## 6 Testovanie

Ako hostiteľské zariadenie sa využíva stolný PC s operačným systémom Windows 10, vo virtualizačnom programe Oracle VirtualBox. Hostiteľský počítač má parametre:

- **Operačný systém:** Windows 10, 64-bitová verzia
- **Procesor:** Intel Core i5-8400
- **Operačná pamäť:** 16 GB

Virtuálna stanica webového serveru s parametrami:

- **Operačný systém:** Ubuntu 18.04.3
- **Procesor:** 1 jadro z hostiteľského zariadenia
- **Operačná pamäť:** 4 GB z hostiteľského zariadenia

Ako klient sa používa naklonovaná virtuálna stanica, s rovnakými parametrami ako pre webový server. Jednotlivé zariadenia sú prepojené pomocou virtuálneho sieťového mostu, tzn. že v sieti sa tvária ako separátne fyzické stanice.

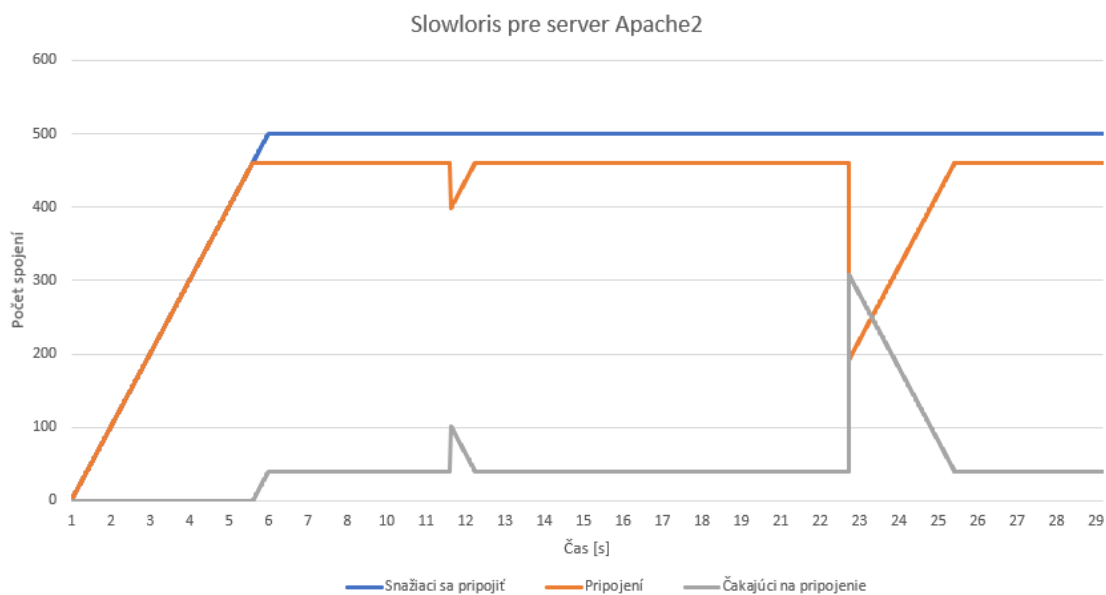
### 6.1 Apache2

Prvý webový server bol vybraný Apache 2 na verzii 2.4.29, hlavne kvôli jeho vysokej popularite vo svete. Pre toto testovanie sú nastavené parametre na prednastavenú hodnotu, tzn. `RequestReadTimeout` je nastavený na 20 sekúnd, `TimeOut` na 60 sekúnd a `LimitRequestBody` na 0, čo znamená neobmedzená veľkosť tela požiadavku.

#### 6.1.1 Slowloris na Apache2

Test prebiehal pomocou generátora s predvolenými hodnotami na útok. tzn. 500 falošných klientov, udržiavacie hlavičky `X-a: b`, odmlčanie klientov bolo nastavené na prednastavené hodnoty, tzn. 10 milisekúnd medzi jednotlivými klientmi a 2 sekundy medzi vlnami.

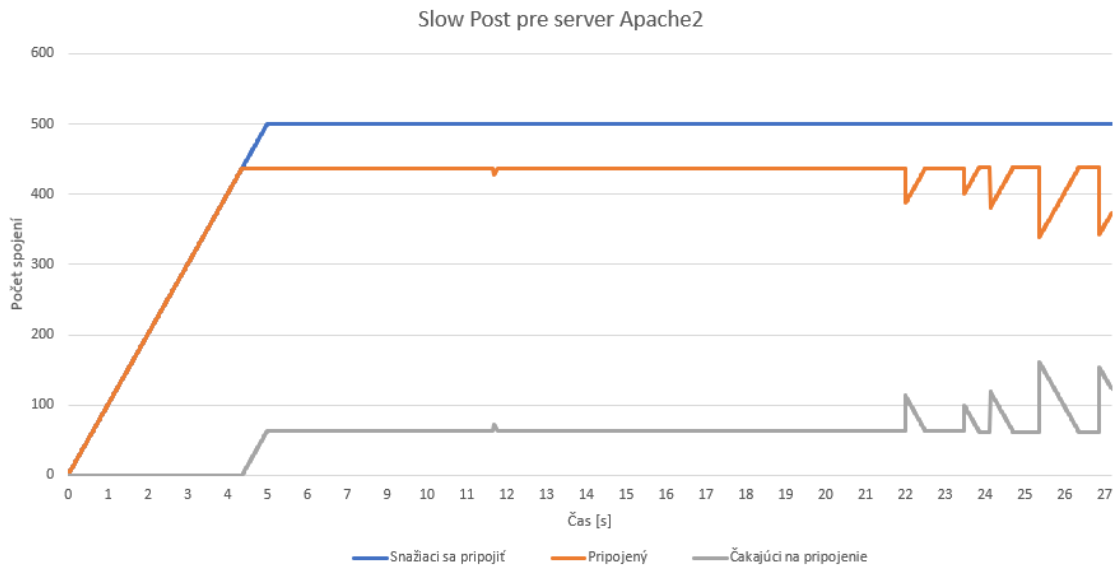
Akonáhle sa všetci klienti začnú pripájať, behom pár sekúnd server neodpovedá na akékoľvek dotazy. Útok prebiehal 30 sekúnd. Počas tejto doby bol neustále server nedostupný. Na grafe je vidieť ako pracoval samotný modul na ochranu proti DoS útokom. Približne v čase 22 sekúnd je viditeľné ako server zrušil prvotné spojenia, ktoré trvali priveľmi dlho, tým obnovil čiastočne službu ale nie na dlhú dobu. Priebeh útoku znázorňuje graf 6.1.



Obr. 6.1: Graf priebehu útoku Slowloris

### 6.1.2 Slow Post na Apache2

Rovnako ako pri útoku Slowloris v sekcii 6.1.1, sa pracuje s rovnakým prostredím. Nastavenia sú na predvolených hodnotách, čiže opäť 500 falošných užívateľov, dĺžka dát je nastavená na 1 000 000 bytov, ktoré sú náhodne generované. Výsledky by mali byť podobné s útokom Slowloris, keďže server drží polootevorené spojenia aktívne, pokiaľ ich klient posiela v dostatočne veľkej rýchlosti. Útok bežal opäť 30 sekúnd, pričom sa opakoval scenár ako pri útoku Slowloris. Pri dosiahnutí dostatočného množstva spojení, serverové služby sa stanú nedostupné. Opäť pri približne 22 sekundách zareagoval modul na ochranu a priveľmi staré spojenia zastaví. Priebeh je možné vidieť na obrázku 6.2



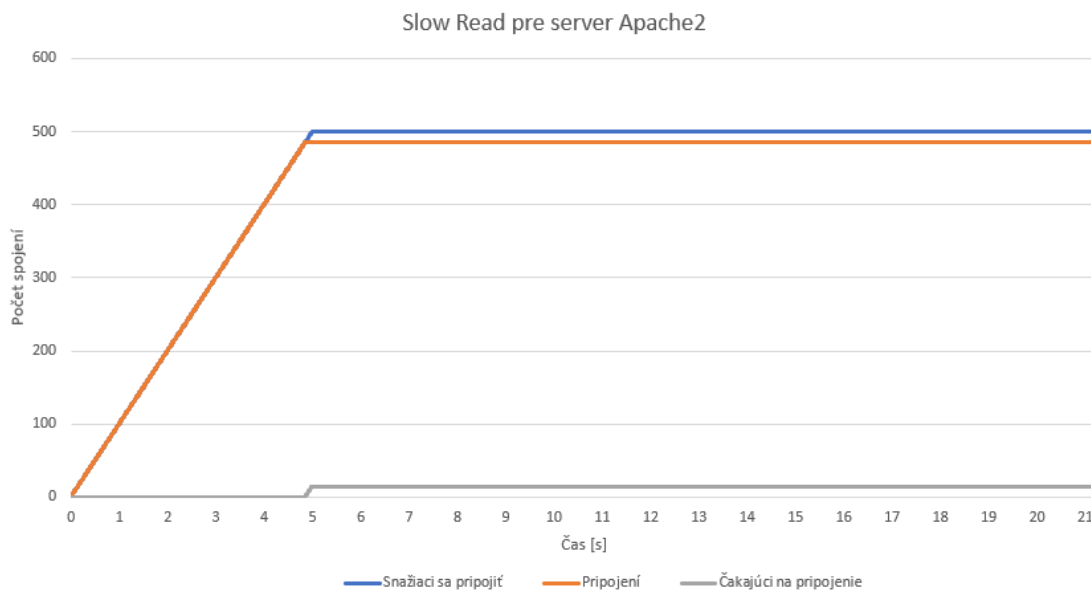
Obr. 6.2: Namerané hodnoty pri útoku Slow Post

### 6.1.3 Slow Read na Apache2

Pri finálnom testovaní serveru Apache2 sa opäť pracovalo s rovnakým prostredím. Predvolených 500 falošných klientov by malo stačiť na zneprístupnenie webového serveru. Predvolená cesta požiadavky na stránku je základná stránka `cat.html`. Jedná sa o obrázok mačky, ktorý bude reprezentovať veľký objekt na webovej stránke.

Všetky hodnoty sú nastavené na predvolené, čiže veľkosť okna na čítanie jednotlivých dát zo serveru je nastavená na 10 bytov. Jeden z možných problémov, ktoré môžu nastať pri takomto útoku je, že server nebude akceptovať priveľmi malé okno od užívateľa a takéto spojenie okamžite zastaví. Prvé inicializácie by sa mali sledovať v software-y **Wireshark** na odpovede od serveru, keďže generátor neobsahuje spätnú väzbu od serveru. Priebeh útoku je možné vidieť na grafe 6.3.

Z grafu je vidieť, že tentokrát ochranný modul nezareagoval, čiže server bol počas celej doby nedostupný. Jediný problém by nastal v momente, keby stránka, na ktorú sa útok dotazuje bola priveľmi malá (napr. textová stránka s 2 vetami). Vtedy by bol útok veľmi zneefektívnený. Preto by sa malo ako cieľ dávať obrázok, poprípade väčšia stránka.



Obr. 6.3: Priebeh útoku Slow Read

## 6.2 Nginx

Ďalším webovým serverom je Nginx, viac o ňom viď 3.2, na verzii 1.17.10. Nie sú zmenené žiadne nastavenia, všetko je vo východných nastaveniach. Medzi podstatné nastavenia patrí `client_body_timeout` nastavený na 5 sekúnd, `client_header_timeout` nastavený tiež na 5 sekúnd, oboje slúžiace na ukončenie pomalých spojení po 5 sekundách a `limit_req_zone` nastavený na 30 požiadavkov za minútu alebo požiadavok každé 2 sekundy.

### 6.2.1 Slowloris na Nginx

Vďaka architektúre, na akej je postavený webový server Nginx, útok Slowloris by mal byť neefektívny, kvôli tomu, že Nginx ako jeden z viacerých serverov používajú cache na takéto požiadavky.

Základné testovacie nastavenia boli rovnaké ako pri teste u serveru Apache2, viď 6.1.1, až na počet spojení, ktoré bolo zvýšené na 1500. Útok prebiehal 40 sekúnd. Počas celého priebehu server odpovedal normálne, bez nejakého výrazného oneskorenia, všetky spojenia si udržiaval. Čiže útok Slowloris je neefektívny na serveri Nginx.

### 6.2.2 Slow Post na Nginx

Ďalší z útokov je Slow Post. Jednou z veľkých výhod serveru Nginx je, že kontroluje URL cesty ešte pred tým ako je vôbec požiadavok dokončený, takže akonáhle príjme



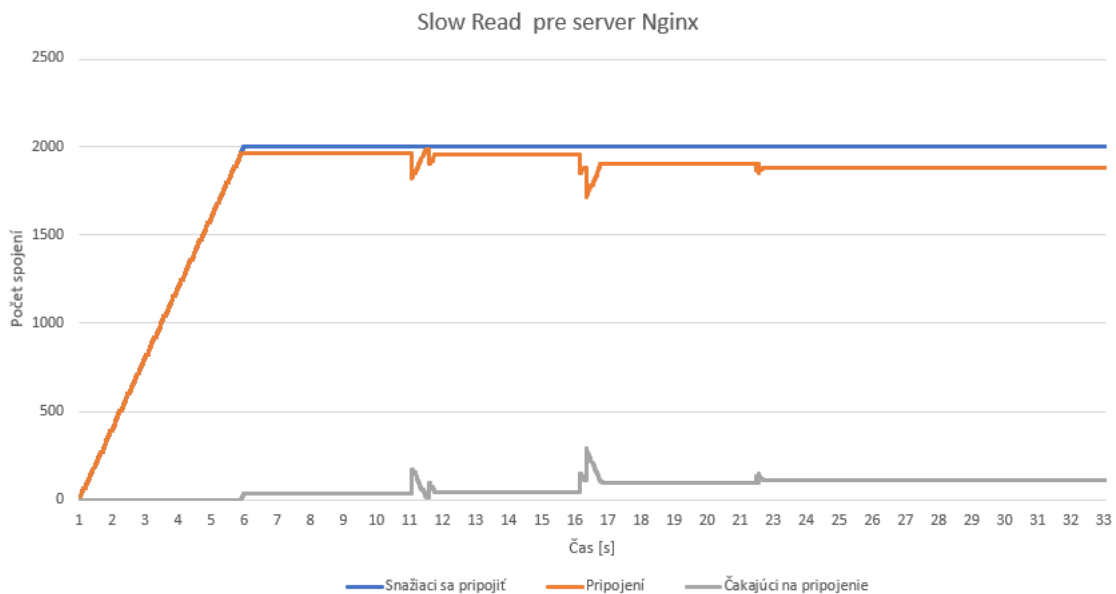
požiadavok na cestu ktorá neexistuje, server ju automaticky odmietne s kódom 404, ktorý znázorňuje, že taká URL cesta neexistuje a požiadavok odmietne.

To je hlavnou príčinou prečo útok Slow Post nefunguje na serveri Nginx. Bolo by potreba doinštalovať PHP časť serveru, ktorá manipuluje s takýmito požiadavkami, lenže takéto útoky by boli smerované priamo na PHP časť a nie na server ako taký. Takže útok Slow Post je neefektívny na webový server Nginx.

### 6.2.3 Slow Read na Nginx

Posledný útok na server Nginx je útok Slow Read. Parametre útoku sú rovnaké ako aj pri 6.1.3, len s rozdielom, že je zvýšený počet klientov na 2000.

Na grafe 6.4 je možné vidieť že server dosiahol maximálny počet klientov približne okolo 1900. Každých 5 sekúnd zareagoval modul na ničenie pomalých spojení. Lenže generátor stíhal obnovovať rýchlejšie spojenia ako ich server dokázal rušiť. Akonáhle bol dosiahnutý maximálny počet pripojených klientov, server nedokázal odpovedať na ďalšie požiadavky a javil sa ako nedostupný.



Obr. 6.4: Priebeh útoku Slow Read na serveri Nginx

## 6.3 lighttpd

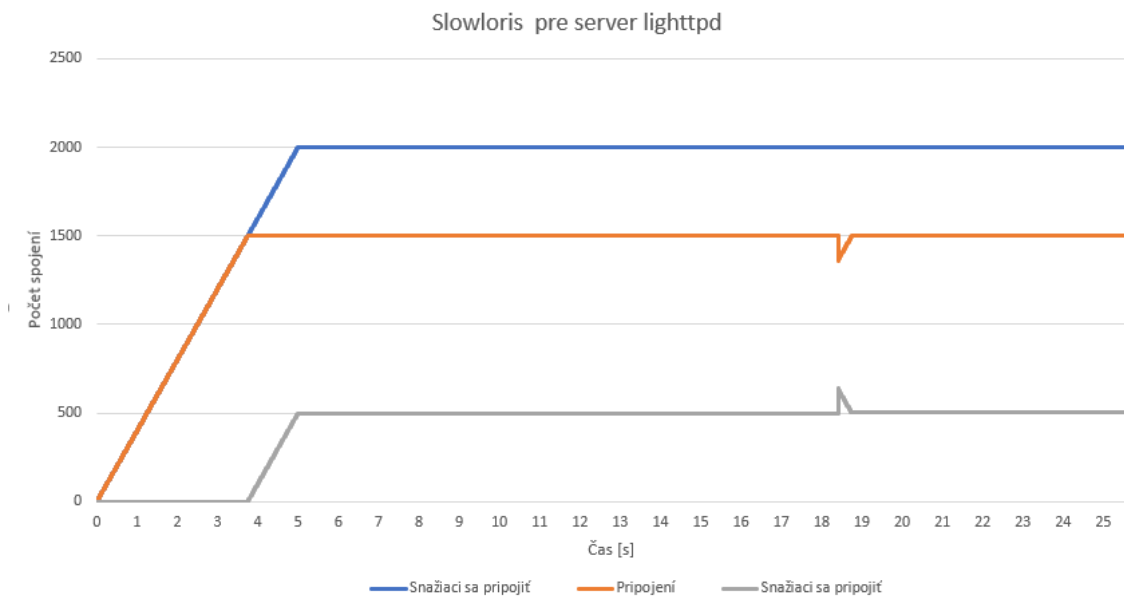
Posledný testovaný server je lighttpd, pre viac informácií, viď 3.3, na verzii 1.4.55. Keďže server lighttpd nemá priveľa nastavení čo sa týka DoS útokov, jedine nastavenie, ktoré je na serveri je `server.kbytes-per-second` nastavený na 0, čo indikuje neobmedzená rýchlosť požiadavkov na server. Ani jedno z obmedzení, ktoré bolo

spomenuté v sekcii 3.3, nemá žiaden vplyv na pomalé DoS útoky, keďže v oboch nastaveniach ide o rýchlosť posielania požiadavkov a pomalé DoS útoky sa vyznačujú veľmi malými prenosmi dát.

### 6.3.1 Slowloris na lighttpd

Rovnako ako pri predchádzajúcich útokoch Slowloris, parametre útoku sú rovnaké, až na počet klientov zvýšených na 2000, kvôli lepšej optimalizácii prijímania a manipulácie s požiadavkami. Server je viac optimalizovanejší, aby dokázal vydržať viac požiadavkov v jeden moment.

Ako je možné vidieť na grafe priebehu útoku, viď obr. 6.5, server bol schopný udržať maximálne okolo 1500 spojení naraz. Akonáhle server dosiahol maximum spojení, server bol nedostupný po celú dobu.

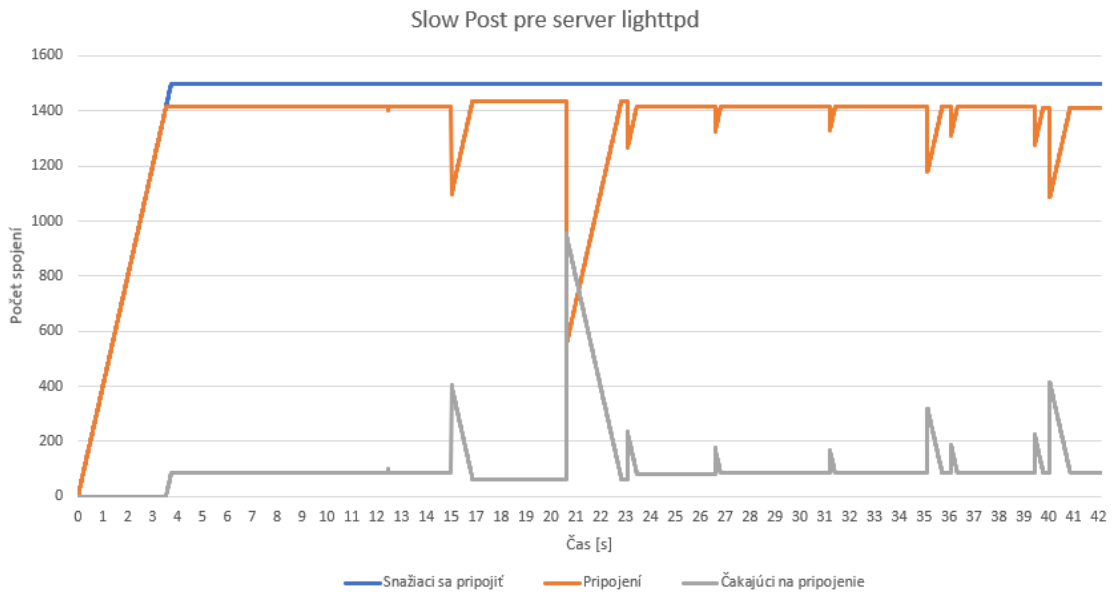


Obr. 6.5: Priebeh útoku Slowloris na serveri lighttpd

### 6.3.2 Slow Post na lighttpd

Opäť, rovnako ako pri Slow Post útoku pri serveri Apache2, viď 6.1.2. Je zvýšený počet klientov na 1500, z rovnakého dôvodu ako pri útoku Slowloris, viď 6.3.1. Ostatné parametre zostávajú stále rovnaké ako pri útoku na webový server Apache2, viď 6.1.1, tzn. Content-length nastavený na 1 000 000.

Z grafu 6.6 je možné vidieť, že server sa snažil odmietat a ukončovať spojenia, lenže generátor ich stíhal obnovovať, že server bol nedostupný počas celého útoku, akonáhle bol dosiahnutý maximálny počet spojení.

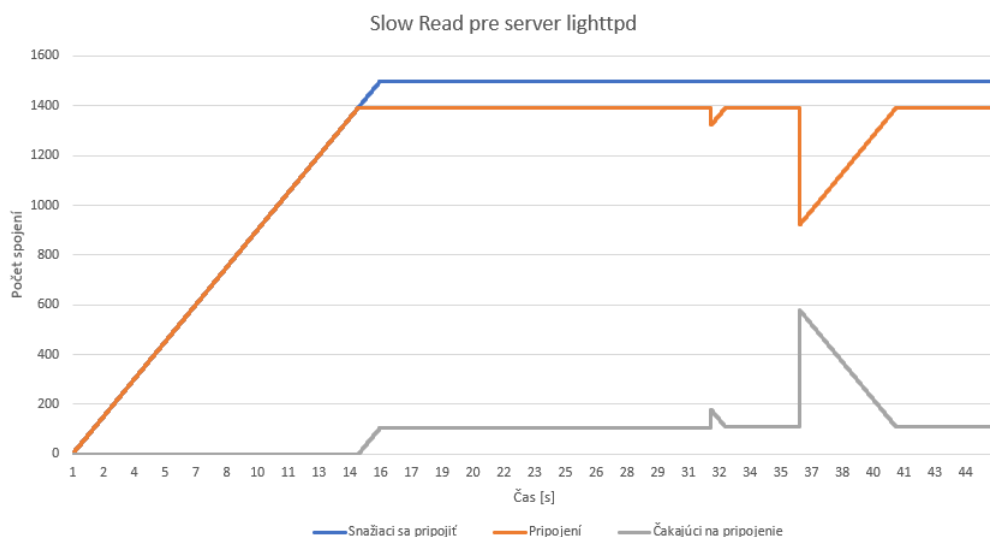


Obr. 6.6: Priebeh útoku Slow Post na serveri lighttpd

### 6.3.3 Slow Read na lighttpd

Posledným testom je útok Slow Read. Rovnako ako pri predchádzajúcich útokoch je referenčná cesta /cat.png. Jedná sa o obrázok mačky, ktorý predstavuje náročnú stránku. Window-size jednotlivých útočníkov je nastavený na 10 bytov.

Z grafu 6.7 je možné vidieť, že server približne po 20 sekundách ukončil spojenia s veľkým množstvom klientov. Zapríčinené to môže byť napríklad veľmi dlhým spojením alebo nejakým vnútorným ochranným mechanizmom v serveri.



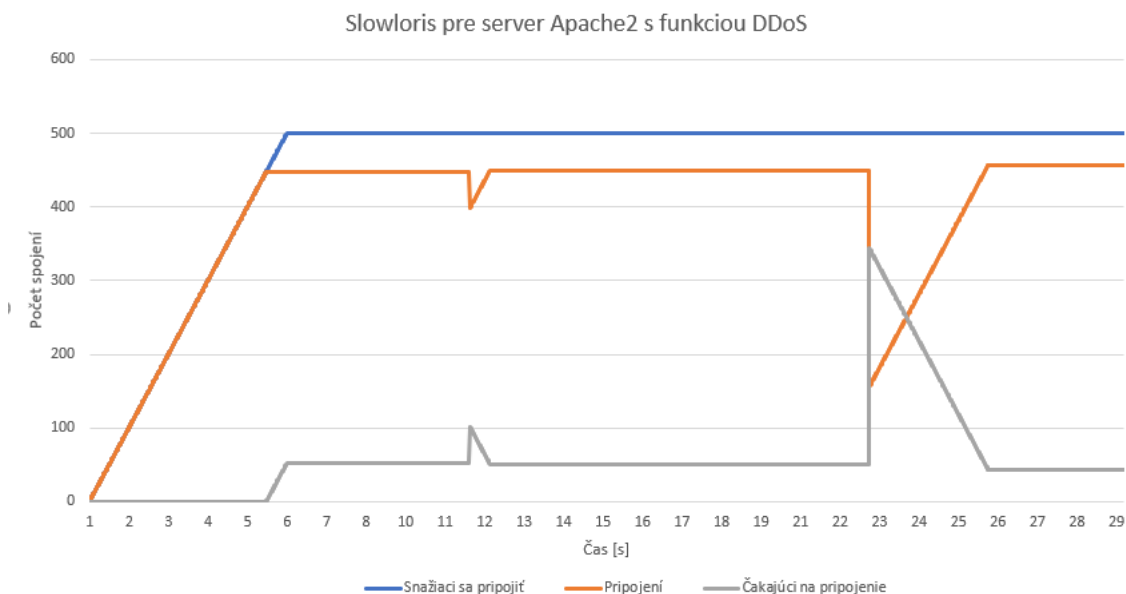
Obr. 6.7: Priebeh útoku Slow Read na serveri lighttpd

## 7 Testovanie DDoS

Na posledné testanie boli vybrané jednotlivé útoky na webové servery v režime DDoS, viac informácií viď 5.2.

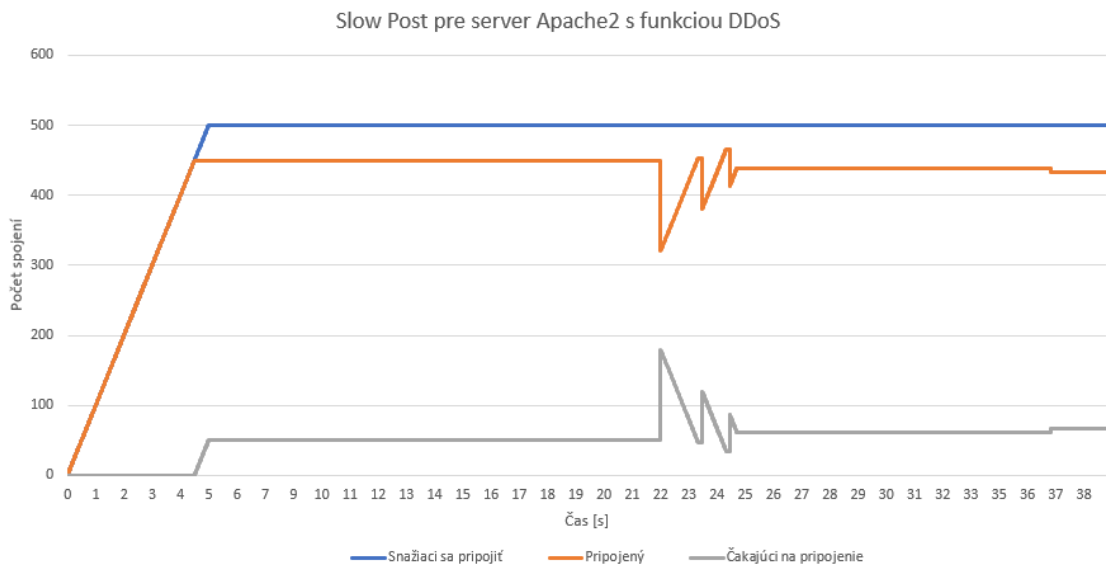
### 7.0.1 DDoS na serveri Apache2

Útoky na webový server Apache2 boli parametrami útoku identické ako pri normálnych útokoch bez použitia režimu DDoS, viď 6.1. Jediné dodatočné nastavenie na generátore bolo 100 falošných lokálnych IP adries, ktoré mali rozmedzie od xxx.xxx.xxx.120 až xxx.xxx.xxx.220.

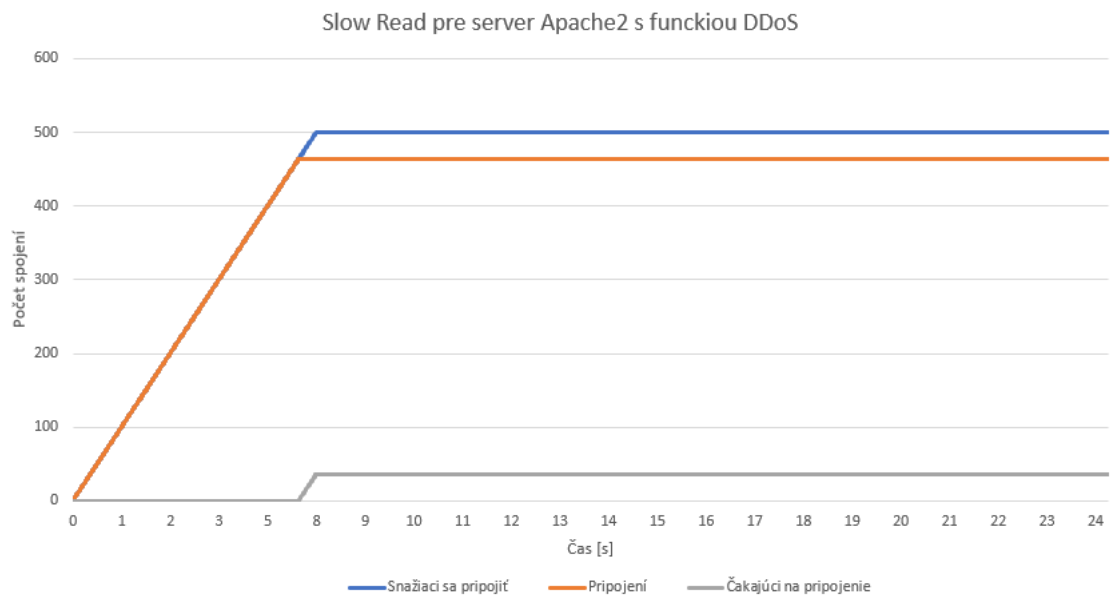


Obr. 7.1: Priebeh útoku Slowloris na serveri Apache2 s modulom DDoS

Ako je možné z grafov vidieť, priebeh útoku bol oveľa vyhladenejší a vo väčšine prípadov bolo zapotreby aj menej falošných klientov.



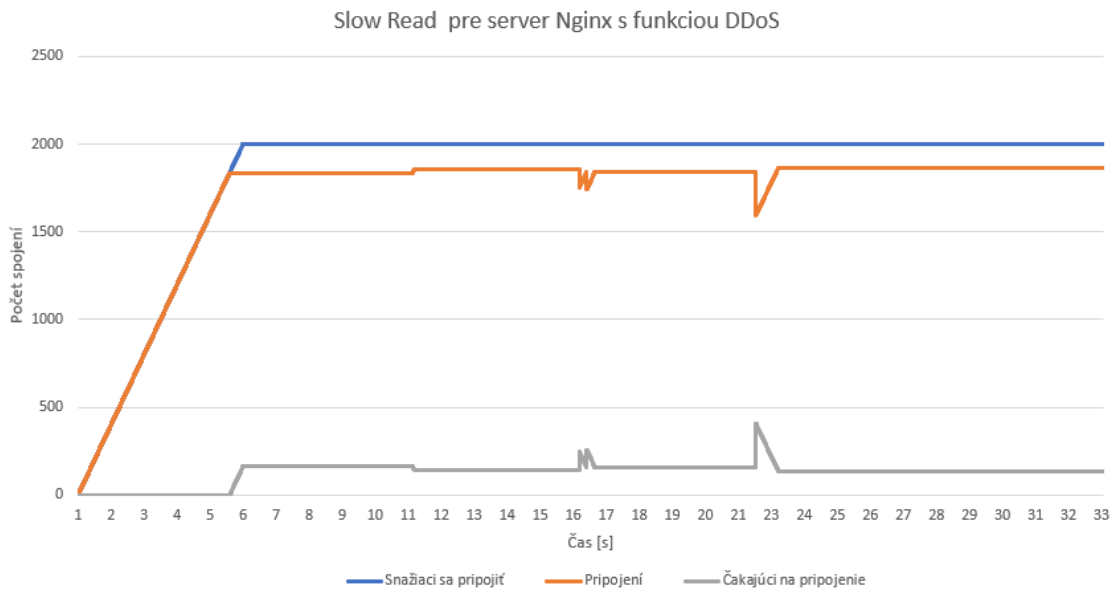
Obr. 7.2: Priebeh útoku Slow Post na serveri Apache2 s modulom DDoS



Obr. 7.3: Priebeh útoku Slow Read na serveri Apache2 s modulom DDoS

## 7.0.2 DDoS na serveri Nginx

Rovnako ako pri DDoS útoku na webový server Apache2, parametre útoku zostávajú rovnaké ako pri normálnom útoku bez využitia DDoS modulu, viď 6.2. Opäť, je nastavených 100 falošných IP adries s rovnakým rozsahom **.120** až **.220**.



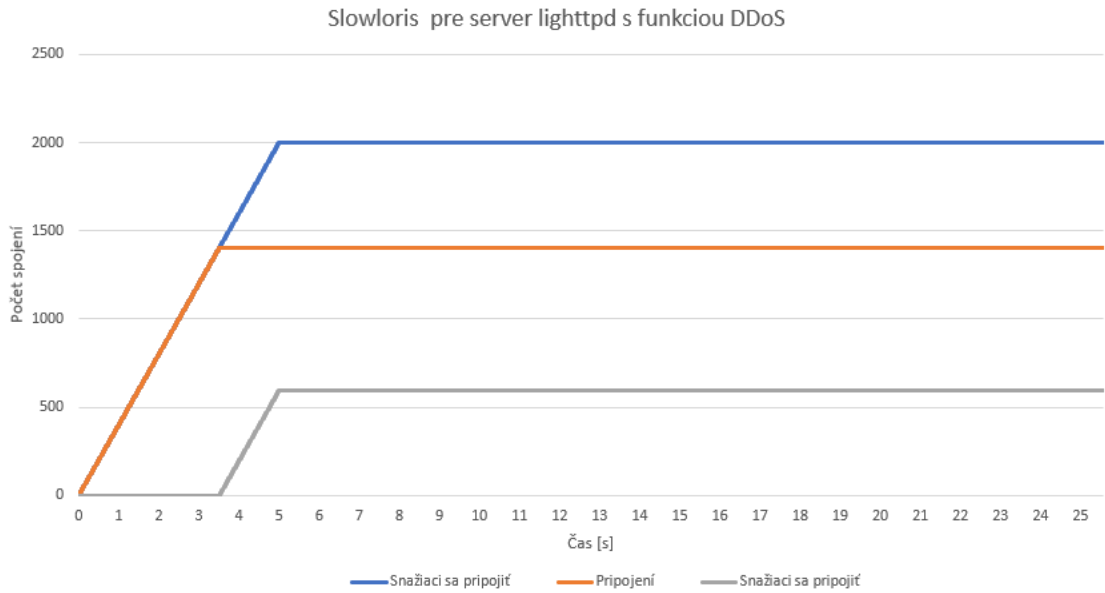
Obr. 7.4: Priebeh útoku Slow Read na serveri Nginx s modulom DDoS

Bohužiaľ, útoky Slowloris a Slow Post neboli opätovne efektívne a boli kompletne odrazené, z rovnakých príčin ako pri normálnych útokoch, viď 6.2.

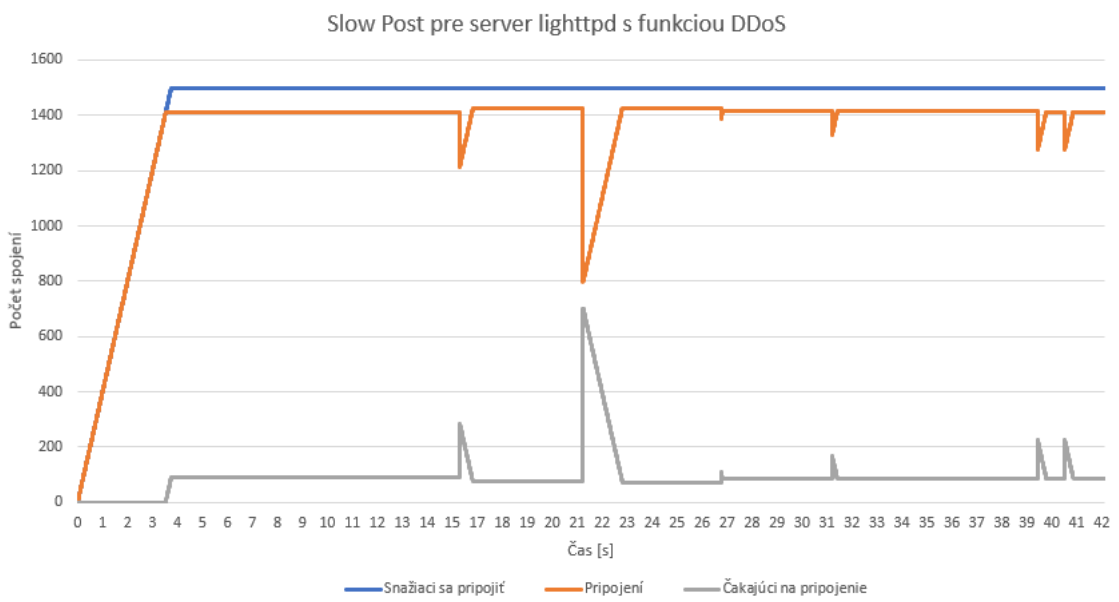
## 7.0.3 DDoS na serveri lighttpd

Ako posledný webový server lighttpd, parametre útoku sú opätovne rovnaké, viď 6.3. Je vytvorených 100 falošných IP adries, v rovnakom rozmedzí **.120** až **.220**.

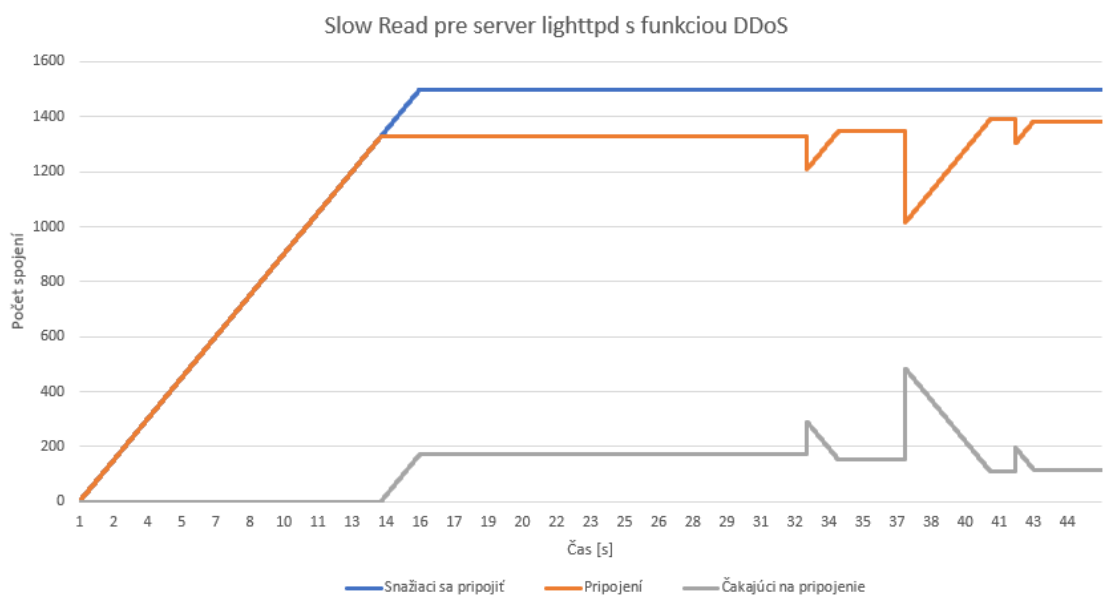
Taktiež ako pri normálnych útokoch, všetky útoky prebehli bez problémov, server bol nedostupný celú dobu útoku. Priebehy sú oveľa vyhladenejšie, čo javí vyššiu efektivitu útoku a nižšie požiadavky na množstvo vytvorených falošných klientov.



Obr. 7.5: Priebeh útoku Slowloris na serveri lighttpd s modulom DDoS



Obr. 7.6: Priebeh útoku Slow Post na serveri lighttpd s modulom DDoS



Obr. 7.7: Priebek útoku Slow Read na serveri lighttpd s modulom DDoS



# Záver

Práca sa zaoberala analýzou vybraných pomalých DoS útokov (Slowloris, Slow Post, Slow Read), navrhnutím modelu generátora a jeho následnou tvorbou a testovaním.

V prvej kapitole bol popísaný samotný model TCP/IP, jeho jednotlivé vrstvy a ich dôležité časti pri komunikácii. Pri aplikačnej vrstve aj samotný HTTP protokol, ktorý sa používa pri Slow DoS útokoch.

V druhej kapitole boli rozobrané jednotlivé kategórie DoS útokov, ako už záplavové ale aj samotné Slow DoS útoky, vybrané útoky, popísaná ich funkčnosť.

V tretej kapitole boli následne popísané zabezpečovacie moduly jedných z najpoužívanejších webových serverov, Apache, Nginx a lighttpd.

Vo štvrtej kapitole boli popísané modely útokov, ich priebeh a chovanie pri komunikácii so serverom. Bola popísaná aj DDoS varianta útokov.

V piatej kapitole bol popísaný generátor, ktorý bol napísaný v jazyku C#, navrhnutý podľa popísaných útokov. Grafické užívateľské prostredie funguje na platforme **WPF**, jeho funkčnosť a parametre nastavení jednotlivých útokov a variant.

V šiestej a siedmej kapitole bol následne generátor otestovaný v testovacom prostredí, otestované jednotlivé útoky proti webovým serverom, či už DoS alebo DDoS varianta, graficky spracované priebehy útokov.

V závere je možné konštatovať, že generátor pracuje správne, je jednoducho konfigurovateľný, GUI je intuitívne. Dokázal obísť ochranné moduly na ochranu proti DoS útokom serveru Apache2 a lighttpd, pre každý jeden útok, po čom servery boli nedostupné pre bežných užívateľov. Server Nginx bol lepšie zabezpečený, pri ktorom bolo možné stav nedostupnosti navodiť iba pri jednom z 3 útokov a to pre útok Slow Read. Ostatné útoky boli zmarené samotným návrhom a manipuláciou s požiadavkami. Taktiež bol otestovaný modul na DDoS útoky, ktoré dopadli o trochu lepšie čo sa týka výkonnosti a efektivity. Ale stále nedokázali prelomiť ochranu webového serveru Nginx a to pri útokoch Slowloris a Slow Post.

# Literatúra

- [1] BOUŠKA, P. TCP/IP– model, encapsulace, paket vs. rámeček. *SAMURAJ-cz* [online]. 16.08.2007 [cit. 2019-12-12] Dostupné z URL: <<https://www.samuraj-cz.com/clanek/tcpip-model-encapsulace-paketu-vs-ramec/>>.
- [2] HTTP and everything you need to know about it. *Medium* [online]. 7.12.2018 [cit. 2019-12-12]. Dostupné z URL: <<https://medium.com/faun/http-and-everything-you-need-to-know-about-it-8273bc224491/>>.
- [3] C. L. Schuba, I. V. Krsul, M. G. Kuhn, E. H. Spafford, A. Sundaram and D. Zamboni, *Analysis of a denial of service attack on TCP*. Proceedings. 1997 IEEE Symposium on Security and Privacy (Cat. No.97CB36097), Oakland, CA, USA, 1997, strany 208- 223. doi: 10.1109/SECPRI.1997.601338. Dostupné z URL: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=601338&isnumber=13107>>.
- [4] Understanding Denial-of-Service Attacks. *US-CERT* [online]. 6.2.2013 [cit. 2020-5-5].
- [5] Y. G. Dantas, V. Nigam and I. E. Fonseca, *A Selective Defense for Application Layer DDoS Attacks* [online]. In: . The Hague, Netherlands: IEEE, 2014, 08 December 2014, s. 75-82 [cit. 2019-12-12]. DOI: 10.1109/JISIC.2014.21. ISBN 978-1-4799-6364-5. Dostupné z URL: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6975557&isnumber=6975536/>>.
- [6] CAMBIASO, Enrico, Gianluca PAPALEO a Maurizio AIELLO. Taxonomy of Slow DoS Attacks to Web Applications. *Recent Trends in Computer Networks and Distributed Systems Security* [online]. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, strany 195–204 [cit. 2019-12-12]. Communications in Computer and Information Science. DOI: 10.1007/978-3-642-34135-920. ISBN 978-3-642-34134-2. Dostupné z URL: <[http://link.springer.com/10.1007/978-3-642-34135-9\\_20](http://link.springer.com/10.1007/978-3-642-34135-9_20)>.
- [7] RAMACHANDRAN, Vivek a Sukumar NANDI. *Detecting ARP Spoofing: An Active Technique* Kolkata, India, 2005, strana 239 [cit. 2020-5-5]. ISBN 978-3-540-30706-8. Dostupné z URL: <[https://books.google.sk/books?id=4LmERFxBzSUC&pg=PA239&redir\\_esc=y#v=onepage&q&f=false/](https://books.google.sk/books?id=4LmERFxBzSUC&pg=PA239&redir_esc=y#v=onepage&q&f=false/)>.
- [8] J. Park, K. Iwai, H. Tanaka and T. Kurokawa, *Analysis of Slow Read DoS attack* [online]. Melbourne, VIC, Australia: IEEE, 2014 [cit. 2019-12-12]. ISBN 978-4-8855-2292-5. Dostupné z URL: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6979803&isnumber=6979787/>>

- [9] April 2019 Web Server Survey. *Netcraft* [online]. Netcraft, ©1995-2019 [cit. 2019-12-01]. Dostupné z URL: <<https://news.netcraft.com/archives/2019/04/22/april-2019-web-server-survey.html/>>.
- [10] NetData. *GitHub* [online]. ©2016-2019 [cit. 2019-12-10]. Dostupné z URL: <<https://github.com/netdata/netdata/>>.
- [11] Apache Security Tips. *Apache* [online]. The Apache, ©1994-2019 [cit. 2019-12-01]. Dostupné z URL: <[https://httpd.apache.org/docs/trunk/misc/security\\_tips.html/](https://httpd.apache.org/docs/trunk/misc/security_tips.html/)>.
- [12] NGINX vs. Apache: Our View of a Decade-Old Question. *NGINX* [online]. Október 2015 [cit. 2020-5-5]. Dostupné z URL: <<https://www.nginx.com/blog/nginx-vs-apache-our-view/>>.
- [13] Mitigating DDoS Attacks with NGINX and NGINX Plus. *Nginx* [online]. Júl 2015 [cit. 2020-5-5]. Dostupné z URL: <<https://www.nginx.com/blog/mitigating-ddos-attacks-with-nginx-and-nginx-plus/>>.
- [14] lighttpd: Story. *lighttpd.net* [online]. Január 2007 [cit. 2020-5-5]. Dostupné z URL: <<http://www.lighttpd.net/story/>>.
- [15] Lighttpd Traffic Shaping: Throttle Connections Per Single IP (Rate Limit). *NixCraft* [online]. Jún 2009 [cit. 2020-5-5]. Dostupné z URL: <<https://www.cyberciti.biz/tips/lighttpd-set-throughput-connections-per-ip.html>>.

## Zoznam symbolov, veličín a skratiek

<b>DoS</b>	Denial of Service
<b>DDoS</b>	Distributed Denial of Service
<b>TCP</b>	Transmission Control Protocol
<b>IP</b>	Internet Protocol
<b>HTTP</b>	Hypertext Transfer Protocol
<b>WPF</b>	Windows Presentation Foundation
<b>DNS</b>	Domain Name System
<b>ARP</b>	Address resolution protocol
<b>MAC</b>	Media access control
<b>URL</b>	Uniform Resource Locator

## A Obsah priloženého média

Priložené médium obsahuje manuál na obsluhu, zložku so zdrojovými kódmi **PacketCannon\_projekt.zip**, pracovnú zložku **PacketCannon**, ktorý obsahuje súbory generátoru potrebného na spustenie vrátane aj samotného spustiteľného súboru. Spustiteľný súbor **PacketCannon.exe**, otvorí grafické užívateľské prostredie, pričom ostatné súbory slúžia ako zdrojové knižnice pre samotný chod generátoru.

```
/.....Koreňový adresár priloženého média
├─ Bakalarska_praca.pdf ..... Elektorinická kópia tejto práce
├─ Manual.pdf ..... Manuál na obsluhu generátoru
├─ PacketCannon_projekt.zip.....Projekt so zdrojovými kódmi
├─ PacketCannon ..... Pracovná zložka s generátorom
│   └─ PacketCannon.exe.....Spustiteľný súbor generátora
│       └─ PacketCannon.exe.config.....Konfiguračný súbor generátora
│           └─ PacketCannon.pdb ..... Programová databáza na odladovanie
│               └─ PcapDotNet.Base.dll.....Pracovná knižnica Pcap.Net
│                   └─ PcapDotNet.Base.pdb.....Pracovná knižnica Pcap.Net
│                       └─ PcapDotNet.Base.xml.....Pracovná knižnica Pcap.Net
│                           └─ PcapDotNet.Core.dll.....Pracovná knižnica Pcap.Net
│                               └─ PcapDotNet.Core.Extensions.dll ..... Pracovná knižnica Pcap.Net
│                                   └─ PcapDotNet.Core.Extensions.pdb ..... Pracovná knižnica Pcap.Net
│                                       └─ PcapDotNet.Core.Extensions.xml ..... Pracovná knižnica Pcap.Net
│                                           └─ PcapDotNet.Core.pdb.....Pracovná knižnica Pcap.Net
│                                               └─ PcapDotNet.Core.xml.....Pracovná knižnica Pcap.Net
│                                                   └─ PcapDotNet.Packets.dll ..... Pracovná knižnica Pcap.Net
│                                                       └─ PcapDotNet.Packets.pdb ..... Pracovná knižnica Pcap.Net
│                                                           └─ PcapDotNet.Packets.xml ..... Pracovná knižnica Pcap.Net
│                                                               └─ WpfAnimatedGif.pdb.....Pracovná knižnica pre WPF
│                                                                   └─ WpfAnimatedGif.dll.....Pracovná knižnica pre WPF
│                                                                       └─ WpfAnimatedGif.xml.....Pracovná knižnica pre WPF
```