

MASARYKOVA UNIVERZITA  
FAKULTA INFORMATIKY



# **Rozpoznanie uživateľa na základe informácií o HTTP komunikácií**

DIPLOMOVÁ PRÁCA

**Matej Majdiš**

Brno, jeseň 2016



*Namiesto tejto stránky vložte kópiu oficiálneho podpísaného zadania práce a  
prehlásenie autora školského diela.*



## **Prehlásenie**

Prehlasujem, že táto diplomová práca je mojím pôvodným autorským dielom, ktoré som vypracoval samostatne. Všetky zdroje, pramene a literatúru, ktoré som pri vypracovaní používal alebo z nich čerpal, v práci riadne citujem s uvedením úplného odkazu na príslušný zdroj.

Matej Majdiš

**Vedúci práce:** doc. RNDr. Vlastislav Dohnal Ph.D. title



## **Podakovanie**

TODO

# Zhrnutie

TODO



## **Klíčové slová**

keyword1, keyword2, ...



# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
1.1	<i>Aplikácie typu Klient-Server</i>	2
1.1.1	Klient-Server model	2
1.1.2	Architektúra	3
<b>2</b>	<b>Analýza sieťových vrstiev a protokolov</b>	<b>5</b>
2.1	<i>Aplikačná vrstva</i>	5
2.2	<i>Transportná vrstva</i>	5
2.3	<i>Sieťová vrstva</i>	5
2.4	<i>Vrstva sieťového rozhrania</i>	5
<b>3</b>	<b>Útoky typu <i>Denial of Service</i></b>	<b>7</b>
3.1	<i>Základne typy a techniky</i>	8
3.1.1	Distribuované DoS útoky	8
3.1.2	Sémantické DoS útoky	9
3.1.3	Brute-force útoky	10
3.1.4	Reflexia a zosilnenie	10
3.1.5	HTTP POST DoS útoky	10
3.2	<i>DoS nástroje a DoS as a Service</i>	10
3.3	<i>DoS - Zhrnutie</i>	12
<b>4</b>	<b>Existujúce prístupy k identifikácií</b>	<b>13</b>
4.1	<i>Využitie sieťovej vrstvy</i>	13
4.1.1	Internet Protocol (IP)	13
4.1.2	Nedostatky	13
4.2	<i>Aplikačné identifikátory</i>	13
<b>5</b>	<b>Tvorba unikátneho identifikátoru</b>	<b>15</b>
5.1	<i>Možnosti protokolu HTTP</i>	15
5.2	<i>Možnosti TCP</i>	15
5.3	<i>Popis Algoritmu</i>	15
5.4	<i>Záver</i>	15
	<b>Register</b>	<b>17</b>
<b>A</b>	<b>Príloha</b>	<b>17</b>



## **Zoznam tabuliek**



## Zoznam obrázkov

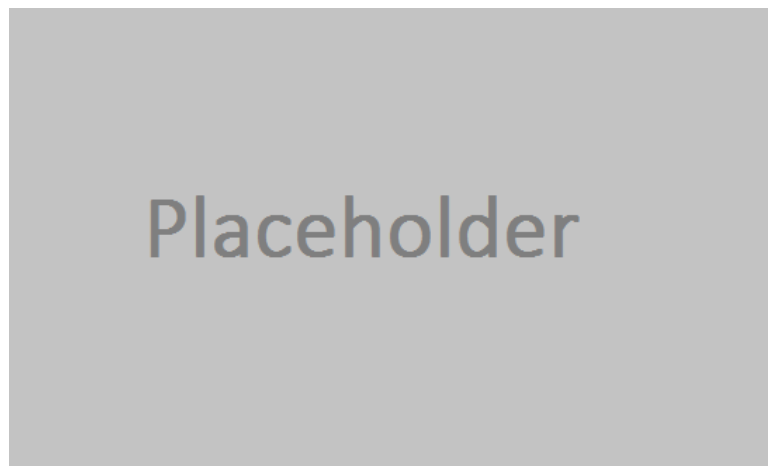
- 1.1 Vizualizácia pomeru počtu užívateľov webových a desktopových aplikácií v čase, zdroj: vlastné spracovanie 1
- 1.2 Schéma znázorňuje základnú architektúru modelu Klient-Server, zdroj: vlastné spracovanie 2
- 1.3 Grafické znázornenie a popis priebehu komunikácie 2-tier architektúry, zdroj: vlastné spracovanie 3
- 1.4 Grafické znázornenie a popis priebehu komunikácie 3-tier architektúry, zdroj: vlastné spracovanie 4
- 3.1 Schéma DoS útoku, zdroj: vlastné spracovanie 7
- 3.2 Schéma rozloženia DDoS útoku, zdroj: vlastné spracovanie 9
- 3.3 Schéma rozloženia DRDoS útoku, zdroj: vlastné spracovanie 11





# 1 Úvod

Problematika jednoznačnej identifikácie používateľa je dnes veľmi dôležitou a riešenou témou. Jedným z hlavných dôvodov je fakt, že väčšina dnešných existujúcich, prípadne novo vznikajúcich systémov a aplikácií je nejakým spôsobom zapojená do Internetu. Zároveň znamená nárast aplikácií, ktoré poskytujú užívateľom webové rozhranie a ústup takzvaných desktopových aplikácií.



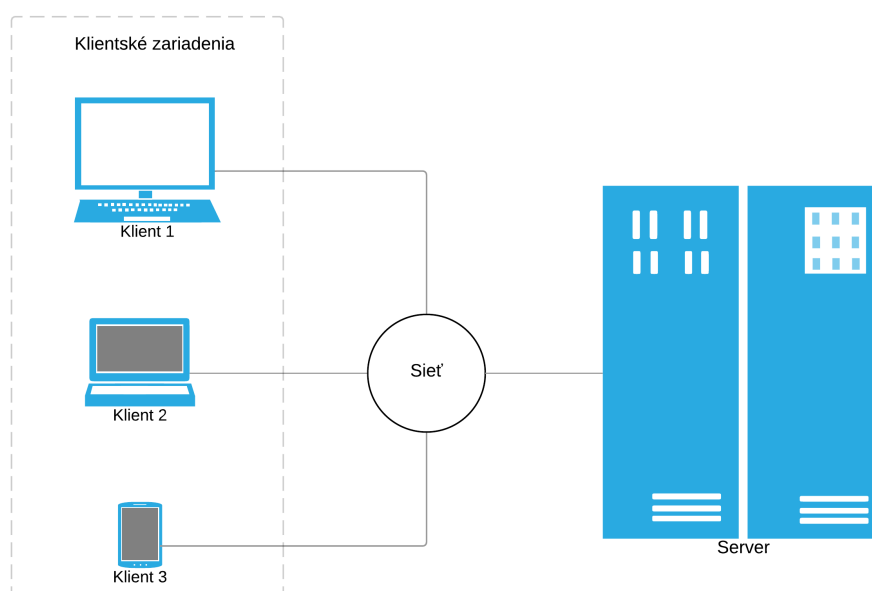
Obr. 1.1: Vizualizácia pomeru počtu užívateľov webových a desktopových aplikácií v čase, zdroj: vlastné spracovanie

Z tohto vyplýva potreba rozoznania a identifikácie používateľov, ktorý s danou aplikáciou interagujú. Existuje niekoľko rôznych prístupov k identifikácii, od mapovania IP adres sieťovej vrstvy až po aplikačnú správu užívateľských účtov. Podrobne sa nimi zaoberá kapitola 4. Cieľom tejto práce je vytvoriť unikátny identifikátor na základe informácií dostupných z *HTTP* protokolu. Pred zostavením samotného algoritmu je preto dôležité popísať niektoré kľúčové oblasti a postupy.

Nasledujúce kapitoly sa preto budú stručne zaoberať fungovaním aplikácií typu klient-server, modelom sieťových vrstiev či útokmi typu *Denial of Service*. Ďalej v práci popíšem spomínané existujúce prístupy a vlastný návrh algoritmu identifikácie užívateľa.

### 1.1 Aplikácie typu Klient-Server

S pokračujúcim vývojom nových technológií sa Web stáva stále väčšou súčasťou našich životov. Web taktiež už nie je limitovaný prehliadaním na počítačoch. Musí sa prispôbovať rôznym novým technológiám, ako sú napríklad mobilné, či iné zariadenia. Najčastejšie používaným modelom komunikácie pre architektúru webových aplikácií je tzv. Klient-Server model. Základnou myšlienkou tohto modelu je zaslanie požiadavku (*requestu*) klientom na server, ktorý vystupuje ako poskytovateľ služby.



Obr. 1.2: Schéma znázorňuje základnú architektúru modelu Klient-Server, zdroj: vlastné spracovanie

#### 1.1.1 Klient-Server model

Pretože Klient-Server model je používaný rôznymi typmi aplikácií bolo nutné použiť štandardizované protokoly, na základe ktorých bude možné komunikovať. Základné používané protokoly sú: *FTP*

(*File Transfer Protocol*), *Simple Mail Transfer Protocol (SMTP)* a *Hypertext Transfer Protocol (HTTP)*. Bližšie sieťové vrstvy a jednotlivé protokoly popisuje kapitola 2.

### 1.1.2 Architektúra

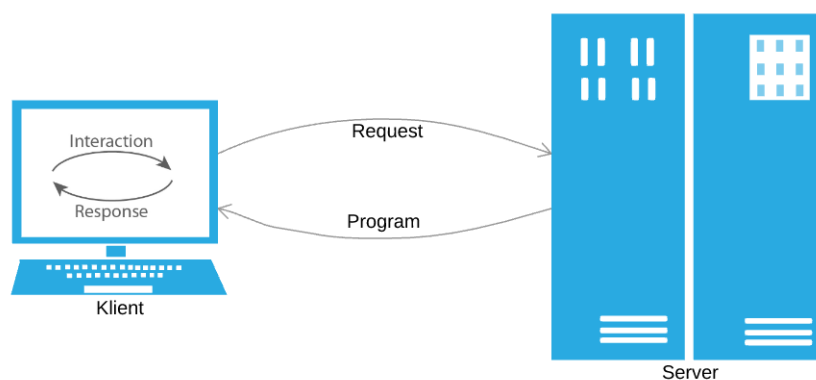
Architektúra modelu Klient-Server sa vo všeobecnosti typicky skladá z troch častí:

- Aplikačný server
- Databázový server
- Zariadenie klienta

Zároveň Existujú dva základné typy architektúr:

- 2-stupňová (2-tier)
- 3-stupňová (3-tier)

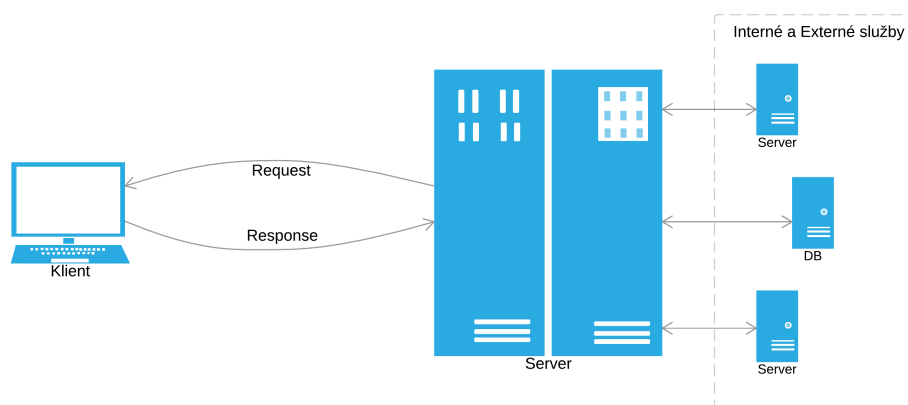
2-tier architektúra zahrna len zariadenie klienta a databázový server. U tohoto typu architektúry je aplikácia spustená na zariadení klienta, ktoré sa následne pripája priamo na server. Zariadenie tak obsluhuje zároveň *business* logiku aj zobrazovanie aplikácie. Inak tento typ architektúry nazývame aj tučný klient (*thick client*).



Obr. 1.3: Grafické znázornenie a popis priebehu komunikácie 2-tier architektúry, zdroj: vlastné spracovanie

## 1. Úvod

3-tier architektúra, ktorou sa budem zaoberať v tejto práci sa od 2-tier líši najmä tým, že okrem zariadenia klienta a databázového servera zahŕňa aj aplikačný server. Tento je následne používaný na obsluhu *business* logiky aplikácie a komunikáciu s databázou, pričom zariadenie klienta slúži len na zobrazovanie. Iný názov pre takýto typ architektúry je tenký klient (*thincient*).



Obr. 1.4: Grafické znázornenie a popis priebehu komunikácie 3-tier architektúry, zdroj: vlastné spracovanie

Nasledujúce kapitoly tejto práce sa budú zaoberať jedným z najdôležitejších problémov Webových aplikácií, ktorým je identifikácia užívateľa. Najskôr kapitola 2 analyzuje jednotlivé sieťové vrstvy a protokoly, ktorých informácie je možné použiť na následnej identifikácii. Ďalšou časťou je zhrnutie existujúcich prístupov k rozoznaniu užívateľov v kapitole 4 a popis útokov typu DOS v kapitole 3. Najdôležitejšou časťou je však samozrejme kapitola 5, ktorá popisuje návrh samotného algoritmu identifikátoru.

## **2 Analýza sieťových vrstiev a protokolov**

//TODO - Úvod

### **2.1 Aplikačná vrstva**

### **2.2 Transportná vrstva**

### **2.3 Sieťová vrstva**

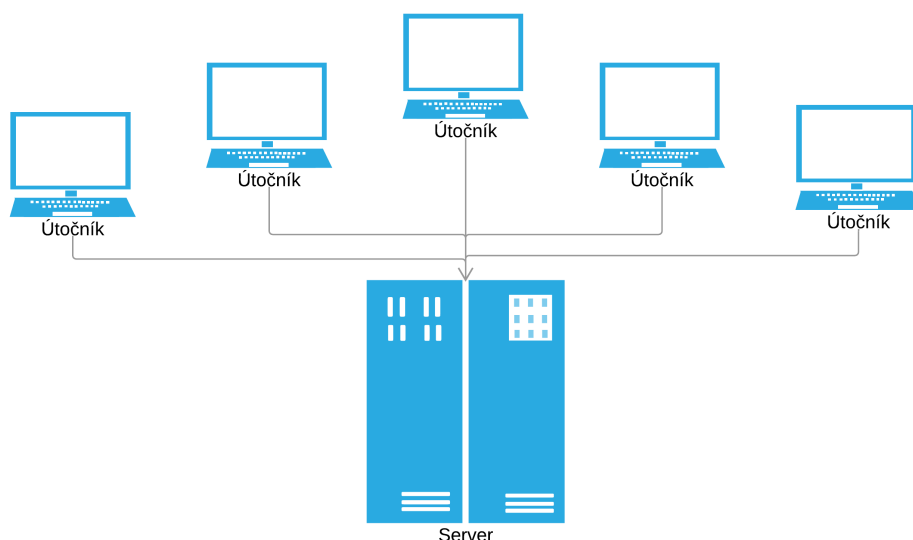
### **2.4 Vrstva sieťového rozhrania**



### 3 Útoky typu *Denial of Service*

Jedným z hlavných dôvodov identifikácie užívateľov je prevencia proti útokom. Medzi najznámejšie z útokov patrí tzv. *Denial of Service*, ďalej len *DoS*.

Vo všeobecnosti je za *DoS* útok považovaná snaha útočníka zabrániť oprávneným užívateľom v prístupe k informáciám, prípadne službám poskytovateľa. Snahou útočníka je znefunkčniť pripojenie neustálym narúšaním služby serveru, prípadne sieťovej infraštruktúry, v dôsledku čoho môže dôjsť k čiastočnej, či úplnej strate internetového pripojenia hostiteľa.



Obr. 3.1: Schéma DoS útoku, zdroj: vlastné spracovanie

U distribuovaného *DoS* útoku môže útočník použiť k útoku na server počítače klientov. Nad týmito zariadeniami je možné prevziať kontrolu využitím bezpečnostných chýb alebo nedostatkov. Takto je následne možné donútiť počítač poslať obrovské množstvo dát na webové servery, prípadne odosielanie nevyžiadanej pošty na konkrétne e-mailové adresy. Útok sa nazýva "distribúovaný", pretože útočník používa viac zariadení na začatie útoku *denial-of-service*.

### 3. ÚTOKY TYPU *Denial of Service*

---

*DoS* útok je podobný veľkej skupine ľudí, ktorá sa zhromažďuje pri vstupe do obchodu a bráni vo vstupe skutočným zákazníkom, ktorých záujmom sú reálne služby. Útočníci vykonávajúci tieto útoky sa často zameriavajú na webové služby a servery, ktoré sú poskytované vysoko profitujúcimi inštitúciami, ako sú napríklad banky, prípadne platobné brány.

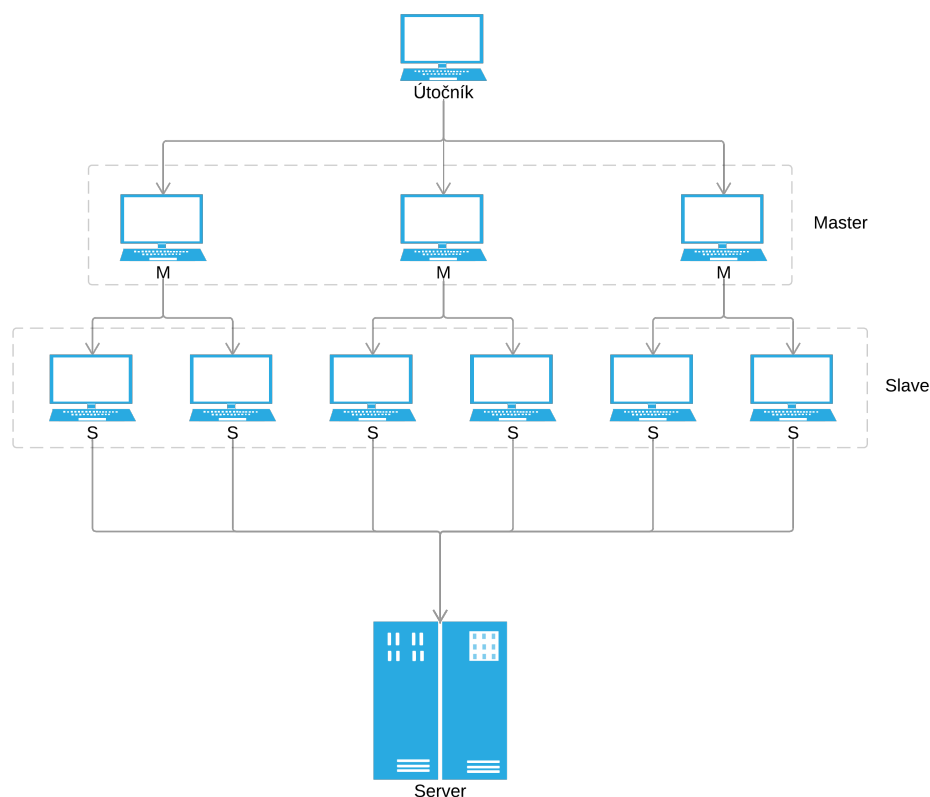
#### 3.1 Základne typy a techniky

Nasledujúce odseky popisujú najpoužívannejšie typy a techniky vykonávania *DoS* útokov a identifikujú prostriedky, ktorými je im možné zabrániť.

##### 3.1.1 Distribuované *DoS* útoky

O distribuovaných *DoS* útokoch hovoríme v prípade, že viaceré zariadenia zaplavia celú šírku pásma prípadne zdrojov cieľového systému, ktorým je zvyčajne jeden, alebo viacero serverov. Takýto útok je často dôsledkom použitia rôznych systémov a zariadení (napríklad *botnetu*), ktoré sa snažia vyťažiť cieľový systém. *Botnet* je rozsiahla virtuálna sieť umelých (*zombie*) počítačov, ktorých cieľom je prijímať príkazy bez vedomia majiteľa. Keď cieľový systém spotrebuje všetky voľné spojenia, ďalšie (nové) už nie je možné nadviazať. Hlavné výhody útočníka pri využití Distribuovaného *DoS* útoku spočívajú v skutočnostiach, že viaceré zariadenia dokážu generovať väčšiu záťaž ako jedno, pričom použitie množstva systémov zabezpečuje omnoho ťažšiu detekciu útočníka a správanie sa každého z týchto zariadení je menej pozorovateľné čo sťažuje obranu voči útočníkovi. Tieto výhody taktiež spôsobujú vývoj obranných mechanizmov. Na strane cieľového serveru už nebude stačiť jednoduché zvýšenie šírky pásma nad hranicu momentálnej veľkosti útoku, pretože útočník môže napríklad zvýšiť počet zapojených zariadení čím by taktiež spôsobil zaťaženie a výpadok systému.





Obr. 3.2: Schéma rozloženia DDoS útoku, zdroj: vlastné spracovanie

#### 3.1.2 Sémantické DoS útoky

Sémantické útoky využívajú špecifickú funkcionality, alebo implementačnú chybu aplikácie, prípadne protokolu zariadenia obete na zneužitie určitého množstva jeho zdrojov. Napríklad v prípade *TCP SYN* útoku je touto zneužitou funkcionalitou alokácia značného množstva priestoru v zozname pripojení ihneď po potvrdení *TCP SYN requestu*. Útočník otvorí viaceré spojenia, ktoré nikdy neuzavrie, čím zahlcuje server.

Pri CGI útoku je cieľom útočníka takýmto spôsobom zahltiť procesor viacerými *CGI requestami*.

### 3. ÚTOKY TYPU *Denial of Service*

---

Jedným z obzvlášť nebezpečných útokov je *NAPTHA* útok, ktorý sa zameriava na *TCP* protokol. Inicializuje mnoho *TCP* spojení, ktoré zaplnia zdroje serveru.

#### 3.1.3 Brute-force útoky

TODO

Source ref: <https://www.eecis.udel.edu/sunshine/publications/ccr.pdf>

#### 3.1.4 Reflexia a zosilnenie

TODO

Source ref (reflection): Vern Paxson. An Analysis of Using Reflectors for Distributed Denial-of-service Attacks. *SIGCOMM Comput. Commun. Rev.*, 31(3):38–47, July 2001.

Source ref (amplification): <http://crt.io/DNS-Amplification-Attacks.pdf>

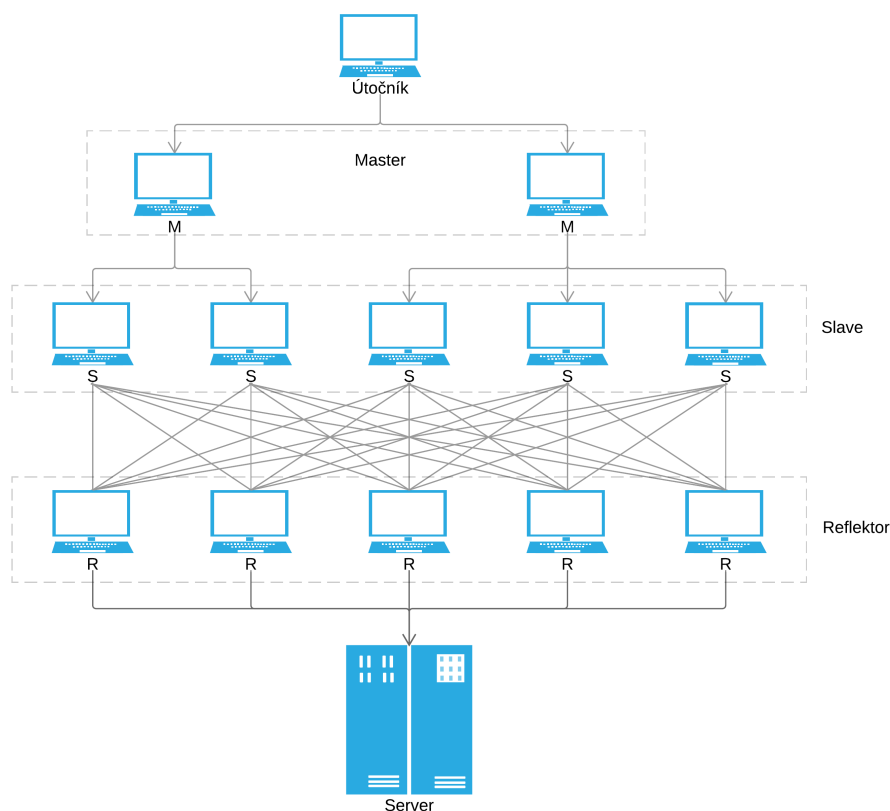
#### 3.1.5 HTTP POST DoS útoky

TODO

### 3.2 DoS nástroje a DoS as a Service

Typickou metódou prenosu mechanizmov *DDoS* útokov je malvér. Jedným z príkladov bol takzvaný *MyDoom*. Ide o *DoS* mechanizmus, ktorý sa spúšťal vo predom naplánovanom čase. Tento útok zahŕňal nastavenie hodnoty *IP* adresy cieľového systému pre nasadenie malvéru, pričom pre spustenie útoku nebola potrebná žiadna interakcia s používateľom.

Ďalším spôsobom zneužitia systému pre *DDoS* útok je použitie skrytej časti softvéru tretej strany, ktorý umožní útočníkovi stiahnutie *zombie* agenta, prípadne ho už softvér sám obsahuje. Útočník môže preniknúť do systému taktiež pomocou automatizovaných nástrojov, ktoré zneužívajú chyby v programoch počívajúcich vzdialené pripojenia. Tento scenár zasahuje primárne systémy, ktoré sa správajú ako webové servery. Typickým príkladom *DDoS* nástroja z tejto oblasti je takzvaný *Stacheldraht*. *Stacheldraht* využíva vrstevnatú štruktúru, v ktorej útočník používa program klienta na pripojenie sa k *handlerom*,



Obr. 3.3: Schéma rozloženia DRDoS útoku, zdroj: vlastné spracovanie

ktoré sú zneužívané na prenos príkazov k *zombie* agentom. Agenti následne vykonávajú samotný *DDoS* útok. Agenti sú cez *handlers* zneužívané útočníkom za pomoci použitia automatizovaných algoritmov na vyhľadávanie zraniteľností v programoch, ktoré prijímajú vzdialené pripojenia. Každý *handler* je schopný kontrolovať až tisíc agentov.

*DDoS* nástroje ako *Stacheldraht* stále používajú klasické *DoS* metódy zamerané na zosilnenie a podvrhovanie IP adries ako napríklad útok vyťaženia šírky pásma. Ďalšou z možností je zahltenie zdrojov - *SYN Flood* útok. Novšie nástroje používajú na *DoS* útoky taktiež *DNS* servery.

Nástroje ako *MyDoom* môžu byť použité voči ľubovoľnej IP adrese. Menej skúsení útočníci ich používajú k znemožneniu dostupnosti

### 3. ÚTOKY TYPU *Denial of Service*

---

populárných a známych webových serverov. Naopak sofistikovanejší útočníci používajú tieto nástroje na vydieranie, napríklad voči svojim obchodným protivníkom.

V niektorých prípadoch sa však môže zariadenie stať časťou *DDoS* útoku zámerne - so súhlasom majiteľa. Príkladom je distribuovaný útok *Operation Payback* organizovaný skupinu *Anonymous*.

—  
TODO DoSaaS

Source ref: <http://krebsonsecurity.com/2015/08/stress-testing-the-booter-services-financially/>

### 3.3 DoS - Zhrnutie

TODO

## **4 Existujúce prístupy k identifikácií**

//TODO - Úvod

### **4.1 Využitie sieťovej vrstvy**

//TODO - Úvod

#### **4.1.1 Internet Protocol (IP)**

#### **4.1.2 Nedostatky**

### **4.2 Aplikačné identifikátory**



## **5 Tvorba unikátneho identifikátoru**

//TODO - Úvod

### **5.1 Možnosti protokolu HTTP**

### **5.2 Možnosti TCP**

### **5.3 Popis Algoritmu**

### **5.4 Záver**





## **A Príloha**

Appendices of thesis.