VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2016 Bc. Jan Henek

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

PROXY SERVERY V SÍTI INTERNET
PROXY SERVERS IN INTERNET

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. Jan Henek**
AUTHOR

VEDOUCÍ PRÁCE — **doc. Ing. Dan Komosný, Ph.D.**
SUPERVISOR

BRNO 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**
Ústav telekomunikací

Student: Bc. Jan Henek ***ID:*** 134487
Ročník: 2 ***Akademický rok:*** 2015/16

NÁZEV TÉMATU:

Proxy servery v síti Internet

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte problematiku proxy serverů v Internetu. Pomocí zhotoveného programu proveďte odhad, kolik stanic v síti Internet komunikuje prostřednictvím proxy serverů. K tomuto odhadu použijte seznamy volně dostupných podvodných IP adres, které se používají například pro zasílání nevyžádané elektronické pošty (spamu) a pro realizaci útoků typu phising.

DOPORUČENÁ LITERATURA:

[1] PUŽMANOVÁ, R. TCP/IP v kostce. 2. vyd. Kopp, 2009. 620 s. ISBN: 978-80-7232-388-3.

[2] Linux Dokumentační projekt. 4. vyd. Computer Press, 2008. 1336 s. ISBN: 978-80-251-1525-1.

[3] COOPER, M. Advanced Bash-Scripting Guide. Steve Glines, 2010. 518 s. ISBN: 978-14-357-5218-4.

Termín zadání: 1.2.2016 ***Termín odevzdání:*** 25.5.2016

Vedoucí práce: doc. Ing. Dan Komosný, Ph.D.
Konzultant diplomové práce:

doc. Ing. Jiří Mišurec, CSc., *předseda oborové rady*

ABSTRAKT

Cílem této práce je analyzovat zastoupení proxy serverů v útocích vedených prostřednictvím internetu. K tomu byla použita metoda srovnání testované IP adresy s databází veřejných proxy serverů. Sestavil jsem seznam IP adres z černých listin, které zahrnovaly škodlivé IP adresy z útoků za rok 2015 a pomocí vytvořeného programu Proxy checker jsem srovnal tento seznam s databází open proxy serverů. Měřením jsem zjistil neúčinnost této metody pro zpětnou detekci proxy serverů v seznamu IP adres z již proběhlých útoků.

KLÍČOVÁ SLOVA

DDoS, detekce, černá listina, Internet, phishing, proxy server, spam

ABSTRACT

The goal of this paper is to analyze the representation of proxy servers in cyber attacks conducted by Internet. For this purpose I used method which compares tested IP address with database of open proxy servers. I assembled a list of IP address taken from the blacklist of cyber attacks committed in 2015. Then I checked this list with the created program Proxy checker and compared them with a database of open proxy servers. By measurement I demonstrate the inefficacy of this method for reverse detection of proxy servers in the IP list of past attacks.

KEYWORDS

blacklist, DDoS, detection, Internet, phishing, proxy server, spam

HENEK, J. *Proxy servery v síti Internet*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2016. 60 s. Vedoucí práce byl doc. Ing. Dan Komosný, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Proxy servery v síti Internet“ jsem vypracoval(a) samostatně pod vedením vedoucího diplomové 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(ka) uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) 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(-ky)

PODĚKOVÁNÍ

Rád bych poděkoval především vedoucímu diplomové práce panu doc. Ing. Danu Komosnému Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno ....

podpis autora(-ky)

OBSAH

Úvod **11**

1 Proxy servery **12**
1.1 Typy proxy serverů 12
1.2 Komunikace pomocí proxy serveru 14
1.3 Praktické využití proxy serveru 15
1.4 Proxy server a anonymita 17
1.4.1 Transparentní proxy server 18
1.4.2 Anonymní proxy server 18
1.4.3 Elitní proxy server 19
1.5 Internetová kriminalita a proxy servery 19
1.5.1 DDoS 19
1.5.2 Botnet 20
1.5.3 Spam 22
1.6 Problematika open proxy serverů 24

2 Detekce proxy serverů **27**
2.1 Detekce ze strany webového serveru 27
2.1.1 Detekce pomocí rozboru HTTP hlavičky 27
2.1.2 Detekce pomocí vedlejších znaků 28
2.2 Detekce pomocí seznamu známých stanic 29
2.3 Honeypot 29

3 Program pro detekci proxy serveru **31**
3.1 Zvolená metoda pro sestavení programu Proxy checker 32
3.2 Nástroj proxy checker API 33
3.3 Struktura programu 34
3.4 Komunikace se serverem metodou POST 35
3.4.1 Metoda POST v kódu programu 38
3.5 Metoda arg.parse 39
3.6 Zpracování výstupu 42

4 Vyhodnocení práce programu Proxy checker **43**
4.1 Základní měření funkčnosti programu 43
4.2 Pokročilé měření funkčnosti programu a parametrů proxy serveru . . 45
4.3 Srovnání efektivity s jinými nástroji 47

5 Výsledky měření **50**

6 Závěr **53**

Literatura **54**

Seznam symbolů, veličin a zkratek **57**

Seznam příloh **59**

A Obsah přiloženého CD **60**

SEZNAM OBRÁZKŮ

1.1 Topologické umístění forward a reverse proxy serveru. 12
1.2 Topologické umístění forward a reverse proxy serveru. 13
1.3 Princip komunikace pomocí proxy serveru. 15
3.1 Vývojový diagram prochy checker skriptu. 36
3.2 Struktura HTTP dotazu. 37
4.1 Poměr zjištěných a nezjištěných stanic. 44
4.2 Rozložení transparentnosti nalezených serverů. 44
4.3 Poměr aktivních a neaktivních stanic z obou měření. 46
4.4 Počet nalezených proxy serverů vůči aktivním stanicím. 47
4.5 Průběh ověřování proxy serverů pro různé nástroje. 49
5.1 Poměr aktivních a identifiovaných stanic vůči portům. 51

SEZNAM TABULEK

1.1 Přehled zneužitých portů open proxy serveru, převzato z [17]. 23
1.2 Dotazy na HTTP a SOCKS moduly, převzato z [17]. 24
1.3 Výsledek testování škodlivosti open proxy serverů, převzato z [22]. . . 26
4.1 Výsledek třetího měření funkčnosti programu. 45
4.2 Výsledek druhého měření funkčnosti programu. 46
4.3 Seznam použitých webů s nabídkou proxy serverů. 48
4.4 Počet detekovaných stanic pro jednotlivé měřící sady. 48
5.1 Přehled ověřovaných portů 50
5.2 Zjištěné počty stanic pro jednotlivé porty. 51

SEZNAM VÝPISŮ

1.1 Záznam HTTP GET dotazu o zprostředkování otevření webové stránky. 25
1.2 Záznam HTTP GET skenu z botnet stanice. 25
1.3 Záznam skenování open proxy serveru od stanice zařazené do blacklistu. 26
2.1 Příklad PHP skriptu pro detekci proxy serveru. 28
3.1 Příklad HTTP POST dotazu. 37
3.2 Odpověď serveru. 38
3.3 POST request. 38
3.4 Argument parser. 40
3.5 Příklad nastavení argumentu pro parser. 41
3.6 Zpracování odpovědi ze serveru. 42

ÚVOD

Historický význam proxy serveru je čistě praktický. S rostoucím počtem stanic, které se zapojovaly do komunikace, zároveň rostly nároky na rychlost sítě a zpracování požadavků. Bylo třeba najít způsob, jak zrychlit a zjednodušit komunikaci. Proxy server slučuje více typů služeb (FTP, HTTP, ...) na jednom serveru a slouží jako prostředník pro komunikaci, odděluje vnitřní síť od vnější, umožňuje filtrovat komunikaci nebo využitím Cache paměti zkrátit čas pro doručení často dotazovaných dat. Urychluje tedy internetové spojení, zobrazení stránek a může fungovat jako aplikační firewall.

S rozvojem internetu se změnil i přístup k proxy serverům. Krom původního využití se dostala do pozornosti schopnost anonymizovat své klienty. Z tohoto důvodu bývá proxy server také spojován s internetovou kriminalitou či neetickým chováním v internetu. Možnosti anonymního pohybu v internetu využívají např. hackerské a hacktivistické skupiny, aktivisté a novináři ze zemí, kde jim hrozí stíhání, nebo uživatelé sdílející nelegální data.

Cílem práce je zmapovat výskyt proxy serverů v již proběhlých kybernetických útocích, včetně spamu a další podvodné činnosti. Protože útočníci často využívají pro svoji anonymizaci veřejné proxy servery, byl sestaven program Proxy checker, který vyhledá z vloženého seznamu IP adres veřejné proxy servery a vypíše jejich parametry. Pomocí tohoto programu byl analyzován rozsáhlý seznam škodlivých IP adres.

Text práce je rozdělen na tři hlavní části – teoretickou, praktickou a analytickou. V první kapitole je rozebrána teorie proxy serverů, princip jejich funkce a praktického využití. Dále se v textu věnuji problematice bezpečnosti v internetu. V sekci *Internetová kriminalita a proxy servery* jsou popsány typické útoky, v kterých figuruje proxy server ve větší, či menší míře. Z černých listin, které vznikly právě na základě některého z popsaných útoků, je sestavena výsledná měřící sada. V druhé teoretické kapitole *Detekce proxy serverů* se zaměřuji na metody, pomocí kterých je možné odhalit proxy server. Praktická část práce je shrnuta v kapitole *Program pro detekci proxy serveru*. Tato kapitola slouží především k popisu vytvořeného programu Proxy checker. V úvodu je uveden postup, který vedl k finální podobě programu, včetně slepých uliček. Dále kapitola obsahuje výňatky z kódu, jejich popis a technický rozbor souvisejících funkcí. Analytickou část práce začíná kapitola *Vyhodnocení práce programu Proxy checker*, v které jsou komentovány výsledky testů efektivity programu a vlastnosti open proxy serverů. Slouží jako teoretická nadstavba k poslední kapitole *Výsledky měření*, která shrnuje měření škodlivých IP adres a detekci proxy serverů v těchto seznamech.

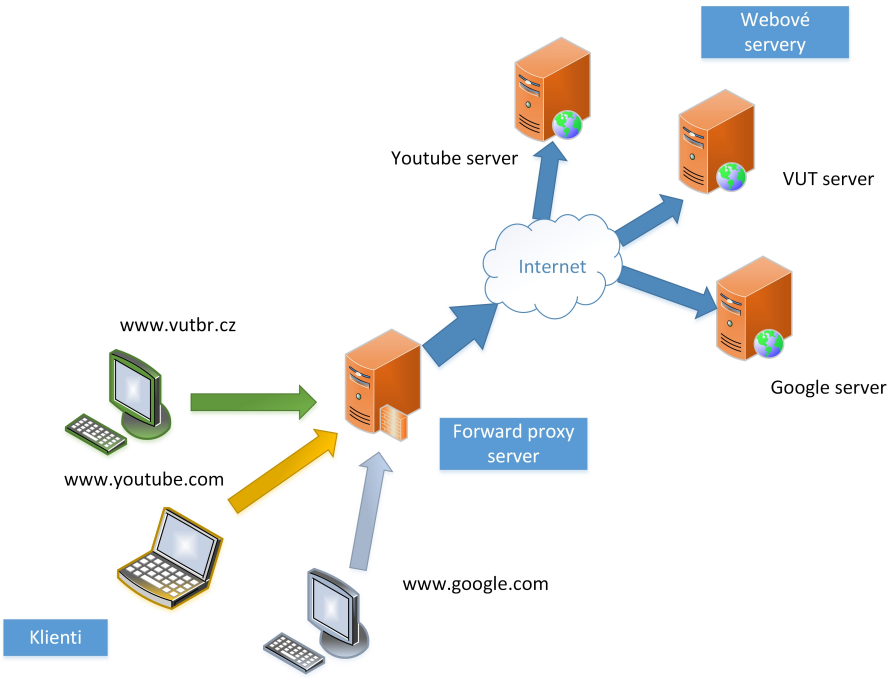
1 PROXY SERVERY

Proxy server slouží jako prostředník v komunikaci mezi klientem a serverem. Přebírá komunikaci za své klienty a vůči cílovému serveru vystupuje jako klient, který zasílá dotaz. Stává se tak příjemcem odpovědi. V komunikační síti může figurovat jako firewall, brána, nebo třeba server. Proxy servery se vyskytují buď v hardware podobě, nebo jako aplikace zaměřené na určitý druh přenosu [1].

1.1 Typy proxy serverů

Tato kapitola shrnuje druhy proxy serverů, dělené podle umístění v síti a podle přístupu ke komunikaci.

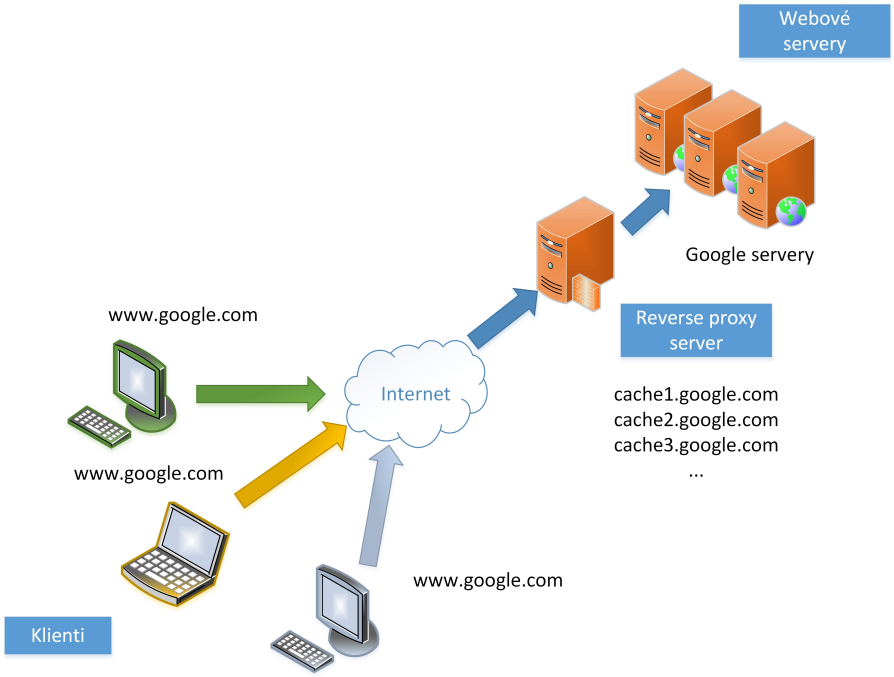
- **Forward proxy:** dopředná proxy je představitelem klasického proxy serveru, umístěného blíže ke klientovi. Klient na dopřednou proxy zasílá dotaz s cílovou adresou a proxy vyřizuje komunikaci za něj. V tomto případě musí být každý klient nakonfigurován tak, aby využíval dopřednou proxy pro komunikaci v internetu. Příkladem je proxy server nastavený přímo v internetovém prohlížeči, nebo již zmíněná caching proxy.



Obr. 1.1: Topologické umístění forward a reverse proxy serveru.

- **Reverse proxy:** je proxy využívaná na straně serveru. Z pohledu klienta přistupujícího ke službě vystupuje zpětná proxy jako server, který může definovat obsah dostupný pro klienta. Zpětná proxy může zastupovat více serverů,

mezi které pak dokáže rozložit zatížení. Slouží také k zprostředkování přístupu zvenčí k serverům, které jsou umístěny za firewallem v bezpečné vnitřní síti. Na straně klienta není třeba nic nastavovat [2].



Obr. 1.2: Topologické umístění forward a reverse proxy serveru.

- **Open proxy:** je speciálním typem dopředné proxy. Je dostupná komukoliv na internetu, jakýkoliv klient ji tedy může využít ke zprostředkování komunikace. Nejznámější využití této služby jsou anonymizační open proxy servery, které maskují klientovu veřejnou IP adresu v další komunikaci. Právě tato vlastnost, společně se snadnou dostupností proxy serveru, svádí ke zneužívání open proxy k nemorálnímu nebo nelegálnímu chování na internetu.
- **Transparentní proxy:** je zvláštním druhem aplikační proxy. Stejně jako open proxy může být zneužita k nelegální nebo amorální činnosti, tentokrát ale vůči klientovi, nebo vnitřní síti. Transparentní proxy totiž nepotřebuje z pohledu klienta žádné nastavení, takže uživatel ani nemusí postřehnout, že jeho komunikace prochází přes prostředníka. V tu chvíli mluvíme o útoku typu man in the midle[1]. Transparentní proxy se chová jako filtr paketů ze strany klienta a jako proxy ze strany serveru. Příkladem z praxe může být populární cachující proxy Squid nastavená v transparentním módu [3].

[1]Muž uprostřed je útočník, který zachytává komunikaci mezi dvěma stanicemi. Krom odposlechu hrozí u zkušenějšího útočníka i podvrhnutí komunikace, zachycení a podvržení veřejného klíče při započetí šifrovaného přenosu apod.

- **klasické aplikační proxy servery:** jsou určeny pro vybrané druhy síťových služeb, obvykle HTTP, HTTPS, FTP a jde spíše o úzce zaměřené programy na daný protokol. Nastavení u klienta je jednoduché, proxy pak využívá některý z výše popsaných principů.
- **SOCKS proxy servery:** nebo také Socket Secure, není pouze aplikace, ale také protokol využívající proxy server ke spojení dvou uživatelů. SOCKS proxy server zde figuruje jako zprostředkovatel spojení. Je třeba mít nainstalovaného SOCKS klienta, případně program podporující SOCKS komunikaci, na obou komunikujících zařízení. Klient posílá žádost o spojení na SOCKS proxy server, který zároveň provádí autentizaci účastníků, následně server zprostředkuje spojení a přenos soketů s druhým SOCKS klientem [4].

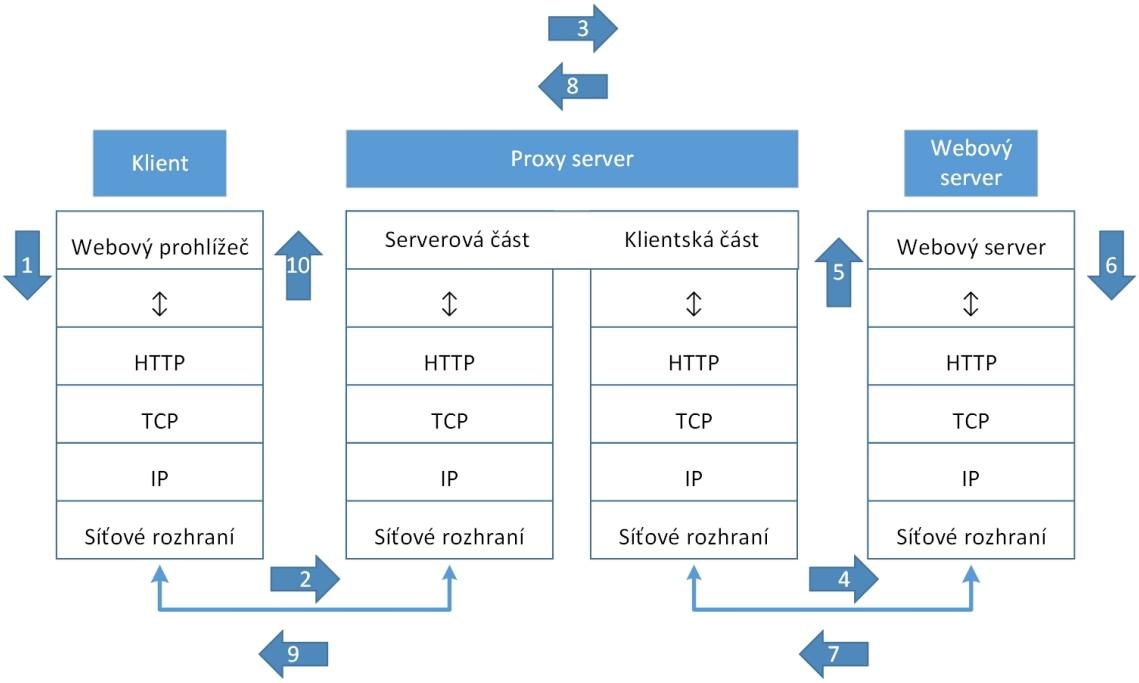
1.2 Komunikace pomocí proxy serveru

Stanice v internetu komunikují nejčastěji způsobem klient-server, což platí dvojnásob, bavíme-li se o HTTP protokolu. V této terminologii se pojmem klient myslí stanice, která zasílá dotaz. Server je pak cíl dotazu, tedy stanice, která má odpovídat. Server může být webový, FTP server, mailový server atd. Pokud je v komunikaci využit proxy server, je postaven mezi klienta a cílový server, pro který se tváří jako klient. Konkrétní umístění proxy serveru už záleží na použitém typu.

Příklad komunikace pomocí proxy serveru je zobrazen na obrázku 1.3, který slouží ke zjednodušené ilustraci postupu dat, proto opomíjí další možné uzly v komunikaci a překlad domény na IP adresu pomocí DNS serveru.

Popis komunikace pomocí proxy serveru [5]:

1. Uživatel zadává do prohlížeče webovou stránku, kterou chce zobrazit.
2. Tato informace postupuje na nižší vrstvy a přes síťové rozhraní je odeslána ve formě HTTP GET požadavku (ten je blíže popsán v kapitole 3.4 na straně 35) na síťovou vrstvu proxy serveru.
3. Proxy server zpracuje požadavek HTTP protokolu, zjistí adresáta a podle nastavení přepíše samotný dotaz. Může dojít např. ke změně hlaviček, v případě anonymního proxy serveru. Proxy server může filtrovat komunikaci, tehdy ověřuje, jestli klient má právo odeslat takový požadavek. V případě cache proxy serverů se ukládají do paměti nejčastěji navštěvované stránky, které se klientovi zobrazí hned, bez nutnosti skutečně kontaktovat webový server. Pokud klientův požadavek vyhověl všem pravidlům, je předán klientské části proxy serveru.
4. Klientská část navazuje spojení s cílovým serverem, v tomto případě s webovým serverem a zasílá HTTP GET požadavek na otevření stránky.



Obr. 1.3: Princip komunikace pomocí proxy serveru.

5. Webový server přijme a zpracuje dotaz.
6. Webový server vytvoří odpověď (webovou stránku) a předává ji nižším vrstvám.
7. Síťové rozhraní webového serveru naváže spojení s klientskou částí proxy serveru, protože jej považuje za zdroj dotazu.
8. Proxy server zpracuje odpověď webového serveru. Pokud se na příchozí data uplatňují nějaká pravidla, upraví podle nich data (filtrace, cache paměť, ...).
9. Sererová část proxy serveru předává odpověď na klienta.
10. Webový prohlížeč klienta zobrazuje dotazovanou stránku.

1.3 Praktické využití proxy serveru

Již v úvodu se zmiňuji o důvodech vzniku proxy serverů, kdy šlo především o zrychlení a zabezpečení komunikace. S příchodem kvalitnějších firewallů, dynamických stránek, a mnoha jiných obměn, se změnily i priority pro využití proxy serverů. Současné uplatnění proxy serverů je shrnuto v těchto bodech [5]:

1. **Sdílení internetového připojení** v lokálních sítích, nebo v menších geografických oblastech. Pro domácnost nebo menší organizace, které mají jedno internetové připojení, slouží proxy server ke sdílení internetu. Na hůře dostupných místech, jako jsou např. vesnice, používají někteří poskytovatelé interne-

tového připojení proxy server pro více domácností. Vznikají tak situace, kdy je polovina počítačů vesnice v lokální síti za NAT překladem, navenek tedy všechny stanice působí pod jedinou veřejnou IP adresou.

2. **Zrychlení načítání webového obsahu** je případ využití cache paměti pro zobrazení často navštěvovaných stránek. Klientovi v takovém případě přichází odpověď s webovou stránkou od proxy serveru, nikoliv od webového serveru spravující stránku, což může významně ovlivnit dobu zobrazení obsahu stránky. Nevýhodou jsou stránky s dynamickým obsahem, tehdy je třeba najít kompromis pro časový interval, kdy se aktualizují zapamatované stránky na proxy serveru, aby vůbec mělo smysl používat cache paměť.
3. **Filtrování komunikace.** Proxy server stojí uprostřed komunikace mezi klientem a serverem, takže může filtrovat tok dat. Protože filtrace probíhá na aplikační úrovni, jde o účinný způsob zabezpečení komunikace (prakticky jde o firewall). Může sloužit zaměstnavateli, aby zakázal přístup např. k sociálním sítím, stránkám sdílejícím soubory, atd. Administrátor také může předcházet spamu tak, že filtruje známé škodlivé IP adresy, které bývají sepsané jako tzv. blacklist (nebo také blocklist) – černá listina zakázaných stanic.
4. **Obcházení cenzury, nebo blokování přístupu (ban).** Paradoxně je také možné použít proxy server k obejití filtrů, nebo blokovaného přístupu. Například ban se uplatňuje na konkrétní IP adresu. Pokud se uživatel připojí pomocí proxy serveru, mění svoji IP adresu a vyhne se tak omezení. Stejným způsobem je možné proniknout za cenzuru, kterou diktátorské režimy uplatňují na své občany. Jednou z obraných technik proti DDoS útoku je zakázat všechny IP adresy, přistupující z jiné země, než v jaké je server umístěn. Útočník může reagovat použitím proxy serverů ze stejné země, jako je cíl útoku, čímž udrží DDoS v běhu.
5. **Anonymizace**, nejznámější vlastnost proxy serveru, je proces, kdy proxy server falšuje nebo zatajuje hlavičky s informací o uživatelově IP adrese, případně o své identifikaci jako proxy server. Pro komunikující strany z venčí pak proxy server působí jako klient. Tato vlastnost má své pozitivní využití, např. při skrývání své identity před útočníkem, na druhou stranu je často zneužívána ke kybernetickým zločinům.
6. **Sledování toku dat**, které uživatelé posílají z vnitřní sítě do vnější sítě (internet). Této metody využívají firmy, které se bojí úniku citlivých informací, případně jiných škodlivých aktivit svých zaměstnanců. Pomocí automatického skenování dat je možné odhalit chování, které by společnosti uškodilo, zabránit mu a dohledat viníka.
7. **Rozložení zátěže** na serverech, kde očekáváme výrazné špičky v provozu, nebo pro distribuci oblíbené služby do jednotlivých regionů. Pro nadnárodní

společnosti, které vlastní nějakou oblíbenou službu, je vhodné pro větší dostupnost vytvořit regionální proxy servery, které zpracují svůj okruh uživatelů, než aby se všechny dotazy směrovaly na jediný server.

1.4 Proxy server a anonymita

Obecnou věřeností nejčastěji využívané a známé proxy servery jsou anonymizační open proxy, na které se uživatel připojí pomocí svého webového prohlížeče a dále na internetu vystupuje pod nově získanou IP adresou. Anonymita na internetu je využívaná k nevšedním činnostem, protože připojení k proxy serveru je navíc vynaložené úsilí, které bývá často vyváženo zpomalením procesu komunikace. Neplacené anonymizační služby totiž přinášejí do konané činnosti znatelné zpoždění. Praktické využití open proxy serverů může být obejití zákazu, cenzury, nebo skrytí identity při pohybu v internetu. Příkladem jsou země ovládané diktátorským režimem, například Čína nebo bývalá Sýrie, kde tamní režim omezuje občany při pohybu v internetu, cenzuruje zobrazované zprávy a stíhá je za akt svobodného projevu. Zdokumentovaným případem je Tariq Biasi, Syrský blogger uvězněný v roce 2008 na 3 roky do vězení za kritiku Syrské politiky [6]. Dalším důvodem k anonymizaci při vyhledávání a práci na internetu je jev zvaný informační bublina, nebo také Google bubble [7], který je výsledkem sledování pohybu uživatele a obchodu s těmito daty. Prohlížeče a vyhledávací algoritmy se snaží přizpůsobit svému uživateli a zobrazují mu informace v závislosti na jeho předchozích aktivitách, čímž omezují objektivní náhled na hledané informace, a jednotlivým uživatelům zobrazují diametrálně odlišné výsledky pro stejně zadaná hesla [7]. Tento jev má velmi negativní dopad na jednotlivce, protože podporuje přirozený nekritický přístup k informacím, které vyhovují naší představě, čímž vzniká informační bublina izolující uživatele od reality.

Anonymizace je zároveň zneužívaný prostředek k nelegálním nebo nemorálním činnostem na internetu. Anonymizační síť Tor v minulosti opakovaně využívaly skupiny pedofilů ke sdílení závadného obsahu. Podle Britské studie z roku 2015 je až 80 % aktivity na dark net[2] spojena s dětskou pornografií [8]. Anonymizační proxy servery a VPN služby využívají také hacktivistické organizace při komunikaci a plánování útoků. V roce 2011 zatkla FBI člena LulzSec[3], Cody Krestingera, právě na základě

[2]Dark net, nebo také deep web je pro běžného uživatele nedostupná část internetu, která slouží převážně k ilegálním činnostem, jako je prodej drog, zbraní, sdílení pornografie, objednání vraždy apod. Nejčastěji se pro vstup do dark net používají programy Tor, I2P nebo Freenet.

[3]Lulz Security je skupina hackerů, známá vyřazením webových stránek CIA, nebo krádeží osobních dat od 24,6 miliónů uživatelů ze sítě PlayStation Network. Většina ze zakladatelů byla zatčena v roce 2011.

informací poskytnutých od poskytovatele anonymizačních služeb, společnosti Hide My Ass [9].

Proxy servery se využívají i z důvodu zabezpečení koncových stanic, např. pro vyhnutí se skenování portů, které může odhalit bezpečnostní díry instalovaných aplikací, nebo verzi operačního systému a otevřít tak útočníkovi cestu k zařízení. Pokud uživatel používá open proxy serverů s vyšším stupeň anonymity, než je transparentní proxy server, nezíská webová stránka, případně útočník, informace běžně sdílené při přenosu, jako jsou právě informace o operačním systému, webovém prohlížeči a skutečná IP adresa. Anonymizace pomocí open proxy serveru funguje na několika úrovních, a odvíjí se podle toho, jaké IP hlavičky proxy server posílá. Nejdůležitější jsou [10]:

- REMOTE_ADDR – označuje IP adresu uživatele. V tomto případě to bude vždy adresa proxy serveru.
- HTTP_X_FORWARDED_FOR – obsahuje skutečnou IP adresu uživatele.
- HTTP_VIA – říká, jestli se uživatel připojuje přes proxy server.

1.4.1 Transparentní proxy server

Transparentní proxy server nabízí nejnižší stupeň anonymizace. Jako každý proxy server změní IP adresu uživatele v REMOTE_ADDR, přesto zůstává původní IP adresa v hlavičce HTTP_X_FORWARDED_FOR. Proto není těžké pro web, nebo cílový server, zjistit pravou adresu dotazující se stanice. Hlavička HTTP_VIA se posílá také. Hlavní výhodou transparentního proxy serveru je rychlost, která je v porovnání s dalšími typy anonymizace nejvyšší. Transparentní proxy se využívají k obejití lokálního firewallu, kde je primárním cílem utajit weby, na které přistupuji, místo svojí IP adresy. Pokud má zaměstnanec v práci zakázané určité stránky, jako je třeba facebook nebo file hosting, pomocí transparentní open proxy k nim získá přístup. Další možné zneužití je u podvodů návštěvnosti na serverech, které nemají sofistikovanější způsoby počítání unikátních návštěvníků.

1.4.2 Anonymní proxy server

Anonymní proxy server neposílá uživatelovu IP adresu, místo toho v hlavičce HTTP_X_FORWARDED_FOR vyplňuje svoji adresu, nebo ji nechá prázdnou. Není tak možné zjistit skutečnou adresu stanice. Pořád ale posílá informaci v hlavičce HTTP_VIA, takže druhá strana ví, že komunikace probíhá přes proxy server. To samo o sobě nemusí být vždy vyhodnoceno negativně, protože proxy servery se používají v sítích běžně. Anonymní proxy servery nabízí dostatečnou anonymizaci za přiměřenou rychlost.

1.4.3 Elitní proxy server

Elitní proxy server poskytuje nejvyšší míru skrytí, protože posílá pouze hlavičku REMOTE_ADDR se svojí adresou. Z cílového serveru se tedy zdá, že v komunikaci není prostředníkem žádný proxy server. Z důvodu velké oblíbenosti těchto proxy serverů bývá jejich rychlost nejpomalejší.

1.5 Internetová kriminalita a proxy servery

Proxy servery figurují také u kybernetických útoků, které patří k odlišnému způsobu zneužití proxy serverů, než byly popsány výše. Informace o takovém využití nejsou příliš časté, protože proxy servery slouží především k ochraně před útokem a v případě útoku se špatně identifikují.

Při zneužití proxy serveru ke kybernetickému útoku se tedy již nejedná o klasický proxy server, ale třeba o napadené zařízení, sloužící jako proxy server bez vědomí majitele [11]. Vznikají tak botnety, rozsáhlé sítě napadených počítačů, které útočníkovi slouží ke skrytí vlastní identity, a zprostředkování výpočetního výkonu potřebného pro kybernetický útok. Z toho důvodu je těžké odhalit, jedná-li se o útok přes proxy server, nebo přes zombie stanici. Vzhledem k těmto vlastnostem bývají proxy servery spojovány s útoky typu bruteforce, DDoS, nebo se spamem.

Pro tuto práci jsou stěžejní útoky vedené přes internet za pomocí proxy serveru. Proto shrnu bezpečnostní rizika, které přináší útok na vnitřní síť pomocí proxy serveru, v tomto odstavci. Jde o útoky typu muž uprostřed, kdy útočník vstoupí do komunikace dvou stanic pomocí proxy serveru, který řídí komunikaci, a tak má plnou kontrolu nad posílanými daty. Dalším druhem útoku je BackConnect, který slouží k průniku skrz firewall, zabezpečující vnitřní síť před vnější [12]. Jedním z možných provedení útoku BackConnect je využití back connection[4] SOCKS serveru, který útočník nainstaloval na infikovaný počítač uvnitř sítě a z něj pak posílá data ven.

1.5.1 DDoS

Distributed Denial of Service je technika útoku, při které se útočník snaží vyřadit nějakou internetovou službu (např. server) tak, že ji zahltí požadavky z mnoha zařízení. Existuje mnoho typů DDoS útoků, protože se mění prostředí internetu i způsoby obrany proti nim. Jednou z možných obran proti pokusu o vytížení serveru útokem DDoS je využití proxy serveru, který rozloží vzniklou zátěž mezi více

[4]Back connection proxy pracuje s velkým rozsahem IP adres, kdy každému novému spojení přiřadí novou IP adresu, takže jedno zařízení může vytvářet mnoho spojení a každé s jinou adresou. Navenek tyto adresy nemají nic společného, takže je není možné spojit s původním zařízením.

zařízení. Na druhé straně proxy servery a anonymizační síť Tor stoupá v oblíbenosti při útocích typu DDoS. Podle zprávy Radware Emergency Response Team [13] v rozmezí let 2010 až 2012 vzrostlo 10. násobně zastoupení koncových uzlů Toru mezi IP adresami účastnících se DDoS útoku. Stále však šlo jen o 10 % všech IP adres zachycených při útocích, zbylých 90 % tedy patřilo buď přímo útočníkovi, nebo šlo proxy servery. DDoS útoky využívající proxy servery nebo jiné anonymizační služby jsou nejčastěji vedeny na aplikační vrstvě. Při měsíčním sledování aplikačních DDoS útoků v roce 2015 zjistily členové Incapsula, že téměř 20 % z útoků proběhlo pomocí anonymních proxy serverů [14].

Útočníci využívají více způsobů, jak dosáhnout anonymity při DDoS:

- **Řetěz proxy serverů:** útočník se připojuje z jednoho proxy serveru na druhý, takže vytvoří řetěz, kde každý proxy server komunikuje pouze s předchozím proxy serverem. Vystopovat útočníka s takovým krytím je velmi obtížné.Pokud si útočník vybírá proxy servery umístěné v různých zemích, ideálně znepřátelené k jeho vlastní, zvyšuje tak sílu zabezpečení svého řetězce. Nevýhodou je rychle klesající výkon s rostoucím počtem stanic v řetězci [13].
- **Shotgun DDoS útok:** je termín zavedený ve výzkumu Incapsula, který popsal zneužití proxy serverů k posílení DoS útoku na mnohem účinnější DDoS. Útočník prvně nasbírá rozsáhlý seznam veřejných proxy serverů, přes které pak posílá pomocí některého z DoS nástrojů dotazy na cíl útoku. Napadený server je v krátkém čase dotazován z mnoha různých zařízení, přičemž útočníkovi stačil jediný počítač s průměrnou výbavou. Zároveň je útočník skrytý, protože dotazy přicházejí od proxy serverů a dokáže reagovat na pokusy o geoblacklisting, tedy zablokování adres v závislosti na zemi původu.

1.5.2 Botnet

Botnet je hierarchická síť zařízení (většinou počítačů), kterou kontroluje nejvyšší uzel. Důvodem bývá rozložení výpočetní síly a získání většího počtu zdrojů. Nemusí jít vždy o ilegální činnost, ve skutečnosti je botnet běžně rozšířenou metodou v mnoha odvětvích IT. Systémy, v kterých jsou stanice zapojeny dobrovolně a s vědomím vlastníka, se nazývají cluster, nebo grid. Cluster komunikuje na úrovni lokální sítě, zatímco grid je rozšířen na komunikaci přes internet. Takové sítě stanic se často využívají k vědeckým účelům – výpočty složitých matematických úloh, projekt SETI@HOME, který se snaží najít známky života ve vesmíru, apod. Dalším příkladem masově rozšířeného programu, který využívá botnet, je program Skype od společnosti Microsoft, zde je řeč o VoIP *(Voice over IP)* botnet.

Princip funkce nelegální sítě botnet

Nelegální botnety pracují na stejném principu, pouze s tím rozdílem, že výpočetní stanice (resp. jejich vlastníci) v botnet síti netuší, že jsou jeho součástí. Nejčastějším využitím botnetu je DDoS útok, spam, brute-force prolamování hesel, zachytávání citlivých informací v síti (přístupová data k internetovému bankovnictví, čísla kreditních karet, apod.) a v menší míře jsou pak stanice botnetu nakonfigurovány jako proxy servery.

Botnet sítě se z hlediska struktury skládají ze dvou částí. První je řídící část, v anglických textech označovaná jako C&C (Command & Control), která se podle velikosti botnetu skládá z jednoho, nebo více řídících serverů. Pokud botnet ovládá pouze jedna řídící stanice, jde o centralizovaný botnet, který je jako takový lehčí pro likvidaci (stačí zneškodnit řídí uzel). Pokud botnet obsahuje více řídících stanic, jde o decentralizovaný systém, který v případě výpadku jednoho C&C přesměruje provoz na jiné C&C. Druhou skupinou v hierarchii botnetu jsou napadené stanice uživatelů, také označované jako „zombie“, které se nechtěně stávají součástí sítě botnet. K šíření botnetu se používá těchto metod [15] :

- **Sociální inženýrství** – uživatel je lstí naveden ke spuštění infikovaného souboru, prozrazení hesla, atd.
- **Malware šířený pomocí warez a cracku** – oblíbený způsob šíření botnetu, kdy je ve staženém souboru vložen kus škodlivého kódu (trojský kůň, backdoor). V tomto případě jsou infikované aplikace cracky, generátory klíčů a warez[5] software.
- **Falešné kodeky, ovladače, rogueware** – rogueware je program, který se tváří jako legitimní utilita, vylepšující běh počítače, ale jde o škodlivý software. Příkladem mohou být falešné kodeky, nástroje pro zrychlení běhu počítače, podvržené antiviry atd.
- **Pomocí napadených webových stránek** – nejatějším typem šíření je využití exploit kitů, což jsou stránky speciálně nastavené pro napadání klienta, který s nimi naváže komunikaci. Exploit kit zjistí od přistupující stanice verzi prohlížeče, operačního systému, Javy, aktivních doplňků a následně určí pomocí databáze zranitelnosti cestu, jak infikovat stanici.

Botnet a proxy server

Využití botnetu jako proxy serveru není tak časté, protože vyžaduje hlubší znalost problematiky. V případě botnet proxy serverů jsou infikované stanice nastaveny jako prostředník v komunikaci mezi útočníkem a cílem útoku. Zdrojem útoku je pak

[5]Nelegálně šířený software chráněný autorskými právy. Jde o kradený software, stahovaný často přes Torrent.

vyhodnocena tato stanice, která působí jako pravý útočník, a sankce tak dopadají na prostředníka. Pro útočníka je značnou výhodou fakt, že takto získaný proxy server je zprostředkovatel veškeré komunikace, nikoliv jen HTTP/HTTPS. Jde prakticky o SOCKS proxy server s nejvyšší mírou anonymizace.

V roce 2012 se rychle rozšířil trojan zvaný Backdoor.Proxybox, který přeměnil napadenou stanici na proxy server, připojený do botnetu útočníka. Správce botnetu využíval takto získané proxy servery k posílení své nabídky open proxy serverů, které prodával pod ruskou stránkou proxybox.name. Vlastník botnetu zaručoval 2 000[6] stálých on-line stanic, čistou (nezaznamenávanou) komunikaci a výhodnou cenu, kterou si mohl dovolit nabídnout, protože jednotlivé servery nemusel zakoupit, ani udržovat [16].

Krom skrytí útočníka se botnet proxy servery využívají k rozložení výkonu a to nejčastěji u DDoS útoků, prolamování hesla a ke čtení CAPTCHA[7] (pomocí technologií OCR[8]) a zasílání spamu.

1.5.3 Spam

Spam bývá definován jako nevyžádané masové sdělení šířené pomocí internetové sítě. Open proxy servery jsou spojovány nejčastěji s e-mailovým spamem a se spamem komentářů na fórech a webových stránkách. Problematika spamu existuje už dlouhá desetiletí a společně s vyvíjející se obranou proti spamu se zlepšují i techniky spamu. Dalším přidruženým fenoménem spamu je phishing. Jde o podvodnou metodu, kdy útočník hromadně zasílá podvržené emaily, které se snaží tvářit legitimně, za účelem získání důvěrných informací od uživatele.

Protože je spam nelegální a administrátoři se jej snaží filtrovat, nevyužívají spameři vlastní servery, ani server svého poskytovatele připojení. Proto dochází při spamu k podobné situaci jako u open proxy serverů, tedy ke zneužití špatně nakonfigurovaného serveru třetí strany ke komunikaci.

Původní metodou emailového spamu bylo zneužití open relay serverů. Relay server je poštovní SMTP server, který slouží jako prostředník pro předávání pošty (mail relaying), obvykle v rámci dané domény. Pokud je ale relay server špatně nakonfigurován a neověřuje identitu uživatele, umožňuje komukoliv na internetu využít jeho služby jako prostředníka pro zasílání zpráv jakékoliv jiné stanici v internetu. Takové servery se nazývají open relay servery.

[6]Podle výzkumu Symantec však bylo aktivních kolem 40 000 stanic v jakýkoliv čas.

[7]Zkratka pro „Completely Automated Public Turing test to tell Computers and Humans Apart“, česky plně automatický veřejný Turingův test k odlišení počítačů a lidí.

[8]Zkratka pro Optical Character Recognition, neboli optické rozpoznávání znaků. Jde o digitalizaci textu, který je snímán a posléze rozpoznán programem, aby mohl být přepsán do elektronické podoby.

Protože open relay servery začaly být hromadně filtrovány a zakazovány[9] administrátory, začalo se pro šíření spamu využívat open proxy serverů. Jedná se o stejný princip jako u open relay serverů, tentokrát jde ale o špatně nakonfigurovaný proxy server, který neověřuje uživatele a nabízí své služby prakticky komukoliv. Spamer pak směruje komunikaci na mailový server skrz proxy server, čímž si zajistí anonymitu.

Z těchto důvodů se na anti-spam blacklistech vyskytují jak adresy open realy serverů, tak adresy open proxy serverů.

V roce 2007 proběhlo více výzkumů na téma zneužití open proxy serveru. V Brazílii byla spuštěna síť deseti honeypotů, které měly za cíl zkoumat zneužití open proxy serverů k e-mailovému spamu. Za 15 měsíců běhu zachytili přibližně 525 miliónu spamu, který byl doručen na 4,8 billiónů příjemců [17]. Díky těmto měřením zjistili, že drtivá část spamu (97 %) pochází z pěti zemí: Taiwan, Čína, Spojené Státy Americké, Kanada a Japonsko. Těmto zemím pak odpovídaly i zjištěné autonomní systémy využité ke spamu. Zajímavým zjištěním bylo ale monitorování portů zneužitých na open proxy serverech k zasílání spamu:

Tab. 1.1: Přehled zneužitých portů open proxy serveru, převzato z [17].

Pořadí	**Port**	**Protokol**	**Obvyklá služba**	**%**
1	1080	SOCKS	SOCKS	37,31
2	8080	HTTP	HTTP	34,79
3	80	HTTP	HTTP	10,92
4	3128	HTTP	Squid	6,17
5	8000	HTTP	HTTP	2,76
6	6588	HTTP	AnalogX	2,29
7	25	SMTP	SMTP	1,46
8	4480	HTTP	Proxy+	1,38
9	3127	SOCKS	MyDoom Backdoor	1,00
10	3382	HTTP	Sobig.f Backdoor	0,96
11	81	HTTP	HTTP	0,96

V záznamu služeb v tabulce 1.1 jsou dva řádky věnované Proxy+ a AnalogX, což jsou proxy servery používané v domácnosti. Tyto záznamy tedy zobrazují skenování špatně nakonfigurovaných domácích proxy serverů pro zapojení do seznamu open proxy serverů. Poslední položky pro MyDoom Backdoor a Sobig.f Backdoor jsou

[9]Např. už v roce 2003 bylo pouze 1 % e-mailových serverů ve Velké Británii špatně nakonfigurováno, což je oproti 91 % nezabezpečených poštovních serverů z roku 1997 velký skok [18].

záznamy z infikovaných zařízení, pravděpodobně součástí botnetu, které spamer využívá jako prostředníka ke komunikaci.

Honeypoty sledovaly také využití svých HTTP a SOCKS modulů v závislosti na cílený port dalšího serveru. V tabulce 1.2 je přehled pro každý z modulů, rozdělený podle portu, na který bylo směrováno odchozí spojení. Pro modul HTTP s 97,62 % i modul SOCKS s 87,31 % dominuje jako cílový port třetí strany port 25/TCP, který využívá protokol SMTP. Je tedy zřejmé, že tyto honeypoty byly primárně využity k zasílání spamu. Další záznamy jsou pro HTTP GET dotazy, což je pokus o klasické využití proxy serveru pro otevření stránky.

Tab. 1.2: Dotazy na HTTP a SOCKS moduly, převzato z [17].

Modul	**Typ**	**Počet dotazů**	**%**
HTTP	spojení na port 25	89 496 969	97,62
	spojení na jiný port	106 615	0,12
	GET dotaz	225 802	0,25
	Chyba	1 847 869	2,01
	Celkem	91 677 255	100,00
SOCKS	spojení jdoucí na port 25	46 776 884	87,31
	spojení na jiný port	1 055 081	1,97
	Chyba	5 741 908	10,72
	Celkem	53 573 873	100,00

1.6 Problematika open proxy serverů

Veřejné proxy servery působí na běžného uživatele jako možnost zabezpečit svoji činnost na internetu, ve skutečnosti však skrývají mnoho bezpečnostních rizik. Seznamy open proxy serverů často obsahují servery zneužité, nebo vystavené se záměrem sbírat informace i klientech. V případě zneužitých serverů jde o laxně nastavené proxy servery, které nevyžadují autentizaci a jsou přístupné komukoliv. Skenováním portů bot odhalí nově vzniklý server, na který pošle zkušební dotaz, např. „GET http://www.yahoo.com/ HTTP/1.0 “a čeká na odpověď [19]. Pokud server vrátí stránku Yahoo, jde o špatně zabezpečený server vhodný pro zařazení mezi veřejné proxy servery. Poté zjišťuje anonymitu, kterou server poskytuje, a další parametry, jako je země původu, šířku pásma, rychlost zpracování požadavku atd.

Příklad skenování open proxy serveru

Typický sken, hledající open proxy server, zachytil a publikoval blogger Lenny Zelster [20], když zkoumal internetový šum na svém nově připojeném serveru. Mezi dotazy, přicházející na jeho server, si všiml zvláštního opakujícího se dotazu, který neměl s účelem jeho serveru nic společného. Daný klient zasílal na jeho webový server podivný HTTP GET dotaz pro otevření webové stránky, konkrétně hotel.qunar.com. Šlo o dotaz zaslaný pomocí nástroje onlineProxy.js. To je modul k freeware Totoro, nástroji pro testování webové stránky, jak se zobrazuje na různých prohlížečích (mobil, tablet, monitor, ...). Webová adresa hotel.qunar.com je zmíněna v dokumentaci pro onlineProxy.js jako příklad. Autor skenovacího skriptu ji očividně nezměnil a rozhodl se ji použít jako testovací stránku. Jednoduchým HTTP GET dotazem skript testuje, jestli mu server vyhoví, a přepošle odpověď webového serveru. V případě úspěchu bude server označen za open proxy.

Výpis 1.1: Záznam HTTP GET dotazu o zprostředkování otevření webové stránky.

```
GET http:// hotel.qunar. com/render/hoteldiv.jsp?&
   __jscallback=XQScript_4 HTTP/1.1
Accept-Encoding: gzip,deflate,sdch
Referer: http:// hotel.qunar. com/
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64)
   AppleWebKit/537.36 (KHTML, like Gecko) Chrome
   /35.0.1916.114 Safari/537.36
Host: hotel.qunar.com
```

Další zaznamenaný pokus o detekování nového proxy serveru pochází z počítače, který je infikován Toping malwarem. Tento rootkit trojan slouží k zjišťování citlivých osobních a pracovních informací o uživateli, a pokud napadená stanice používá jako operační systém Microsoft Windows, je přiřazena do botnetu útočníka jako zombie.

Výpis 1.2: Záznam HTTP GET skenu z botnet stanice.

```
GET http:// www. k2proxy. com//hello.html
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT
    6.1; WOW64; Trident/6.0; SLCC2; .NET CLR 2.0.50727; .
   NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC
    6.0; .NET4.0C; .NET4.0E)
```

Posledním zjištěným pokusem o sken je dotaz ze stanice, která se nachází na černé listině od Project Honeypot[10]. Tato stanice se také pokoušela připojit na TCP porty 135 a 1433, které jsou přiřazeny Microsoft SQL serveru.

[10]Dostupný na adrese `http://www.projecthoneypot.org/ip_63.246.129.40`

Výpis 1.3: Záznam skenování open proxy serveru od stanice zařazené do blacklistu.

```
GET http://www.baidu.com/ HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT
    6.1; WOW64; Trident/5.0; SLCC2; .NET CLR 2.0.50727; .
   NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC
    6.0; .NET4.0C; .NET4.0E)
```

Bezpečnostní rizika při použití open proxy serveru

Protože proxy server slouží ke zprostředkování komunikace, může být velmi jednoduše použit ke sledování pohybu uživatele. Při testu 443 open proxy serverů [21] z roku 2015 se ukázalo, že 79 % z nich nepovolovalo přenos pomocí HTTPS. Takové chování se dá vyložit jako pokus majitele proxy serveru sledovat komunikaci. Při druhém testu, o měsíc později, bylo ze vzorku 25 443 open proxy serverů bylo online 13 307 serverů a z toho 62 % nepodporovalo HTTPS [22].

Další rozšířenou technikou je modifikování serveru, aby všechny přenesené soubory javasascriptu byly infikovány vlastním kusem kódu, který zprostředkuje zasílání záznamu komunikace na další úložiště. Pokud se pak nastaví velmi dlouhá doba zachování cache pro .js soubory, probíhá kontrola komunikace i po odpojení se z open proxy serveru [23]. Podle již zmíněného testu open proxy serverů injektovalo vlastní javascript kód v prvním případě 8,5 % serverů, v druhém rozsáhlejším testu šlo o 20 % online proxy serverů.

Pomocí škodlivých open proxy serverů je možné vést DDoS útok, kdy proxy server vyžádá pomocí skriptu po prohlížeči načtení stránky v řádech sto pokusů za vteřinu. Útočníkovi se taky otevírá cesta k infikování zařízení malwarem a tvorby vlastního botnetu. Mezi open proxy servery byla důvěryhodná přibližně 1/4 testovaných [22].

Objekt testování	**Výsledek [ks]**	**Procent vůči online proxy**
Počet testovaných serverů	25443	/
Online stanic	13 307	/
Offline stanic	12 136	/
Nepovolující HTTPS	8 294	62 %
Modifikující JavaScript	2 747	20 %
Modifikované HTML	2 391	18 %
Neupravující obsah	8 171	61 %

Tab. 1.3: Výsledek testování škodlivosti open proxy serverů, převzato z [22].

2 DETEKCE PROXY SERVERŮ

Detekce proxy serveru je z principu obtížná záležitost. Pokud je proxy server využíván k anonymizaci, snaží se zasílat co nejméně informací o svém klientovi, stejně tak o sobě. Ideální stav z hlediska anonymity je ten, kdy nikdo zvenčí není schopný poznat, jestli komunikuje s koncovou stanicí, nebo proxy serverem. V případě dobře utajeného (elitního) proxy serveru je nemožné odhalit útočníka v reálném čase.

2.1 Detekce ze strany webového serveru

Pokud detekce probíhá ze strany webového serveru, ke kterému se testovaný klient připojuje, existuje více metod, jak zjistit přítomnost proxy serveru. Tato metoda hledání proxy serveru má vyšší šanci na úspěch, než hledání pomocí známých proxy serverů, který je popsán v následující sekci, protože přistupuje ke zjišťování pomocí vlastních metod a nečeká, až je proxy server odhalen někým jiným a zveřejněn. Zároveň také administrátor disponuje více nástroji pro detekci.

2.1.1 Detekce pomocí rozboru HTTP hlavičky

Jak již bylo popsáno v sekci 1.4, proxy server zasílá tři důležité hlavičky, podle kterých je možné určit klienta a přítomnost proxy serveru. Příkladem takové detekce je běžící PHP skript na webovém serveru, který kontroluje hlavičky příchozích HTTP dotazů a hledá známky přítomnosti proxy serveru. Jeho postup je následující:

1. Prvně načítá do proměnné `remoteaddr` informaci z hlavičky `REMOTE_ADDR`, která obsahuje IP adresu, z které byl zaslán dotaz na webový server. Tato adresa patří klientovi, pokud nepoužívá proxy server. Pokud ano, jde o adresu proxy serveru.
2. Dalším krok ověřuje, jestli je vyplněná hlavička `HTTP_X_FORWARDED_FOR`. Pokud ano, obsahuje pravou adresu klienta, a zároveň je jasné, že se jedná o proxy server.
3. Třetím krokem by mohlo být ověření hlavičky `HTTP_VIA`, kterou posílá jak transparentní, tak anonymní server. Tato hlavička udává, jestli je v komunikaci prostředník ve formě proxy serveru, nebo nikoliv.

Jak je zřejmé z popisu skriptu, touto metodou není možné rozeznat elitní proxy server, který zasílá pouze hlavičku `REMOTE_ADDR` s vlastní IP adresou. Uvádím zde příklad PHP skriptu, převzatý z [24], který vynechává krok 3. Skript je vložen jako názorná ukázka, nikoliv jako plně funkční návrh.

Výpis 2.1: Příklad PHP skriptu pro detekci proxy serveru.

```
<?php
$remoteaddr=$_SERVER["REMOTE_ADDR"];
$xforward= $_SERVER["HTTP_X_FORWARDED_FOR"];
if (empty($xforward)) {
itshape//itshapeuseritshape itshapeisitshape itshapeNOTitshape
   itshapeusingitshape itshapeproxy
$real_ip_address = $remoteaddr;
}
else {
itshape//itshapeuseritshape itshapeisitshape itshapeusingitshape
   itshapeproxy
$real_ip_address = $_SERVER["HTTP_X_FORWARDED_FOR"];
}
?>
```

2.1.2 Detekce pomocí vedlejších znaků

Protože metoda ověřování hlaviček nemusí být vždy úspěšná, je možné ji doplnit o nepřímé metody detekce proxy serveru. Tyto metody vycházejí z průvodních znaků, které proxy server může zanechat v komunikaci. Jde o kombinaci pasivních i aktivních metod, které je vhodné používat po zvážení, protože při ověřování většího množství klientů může dojít k zpomalení, nebo zahlcení serveru. Zde je jejich seznam, převážně získaný z [25]:

- **Reverzní vyhledávání přes DNS:** provede opačný proces překladu adresy pomocí DNS. V tomto případě se na základě přistupující IP adresy dotazuji DNS serveru na přiřazenou doménu, která může napovědět, jestli jde o proxy server. Jde o aktivní metodu.
- **Tor detekce:** je hledání získané IP adresy v databázi koncových uzlů sítě Tor. Jde o další aktivní metodu.
- **Geolokace:** porovnává získanou IP adresu s lokální geolokační databází. Jde o pasivní metodu.
- **Reverzní TTL ověření:** je metoda, při které se zjišťuje nejnižší TTL *(Time To Live)*, potřebný pro přenos TCP paket mezi serverem a dotazující se stanicí. Následně se porovnává se zjištěnou hodnotou pomocí traceroute. Zjišťuje se, jestli se výrazně neliší počet skoků pro HTTP a ICMP provoz, tedy jestli nejde každý z těchto toků jinou cestou.
- **Testování portů:** je aktivní metoda, při které se na klientovi testují porty,

které často využívají proxy servery. Pokud jsou porty otevřené, pokouší se na ně server doručit HTTP dotazy a vyhodnocuje výsledek.

- **Reverzní traceroute:** další aktivní metoda, při které se server dotazuje pomocí nástroje traceroute na přistupujícího klienta a poté zpracovává nalezené uzly.
- **Test zasláním dat:** některé proxy servery bývají nakonfigurovány tak, že odpovídají jinak na dotaz, který obsahuje data, než na prázdný dotaz. Tato metoda tedy porovnává odpověď na oba druhy dotazu.

2.2 Detekce pomocí seznamu známých stanic

Další možností detekce proxy serveru je procházení veřejných seznamů proxy serverů, které uvádějí jejich IP adresu, nebo doménové jméno. Takový postup je vhodným doplňkem pro předchozí metody detekce. Výhodou této detekce je, že lze zpětně analyzovat zaznamenané IP adresy. Zároveň nevyžaduje vlastnictví webového serveru, ani přímý kontakt s podezřelým klientem. Pro vytvoření databáze známých proxy serverů se nabízí webové stránky, které poskytují seznamy open proxy serverů. Většina volně dostupných seznamů nebývá příliš kvalitních, především z hlediska aktuálnosti. Existují ovšem také placené seznamy, u kterých je předpokládaná vyšší kvalita, jak v životnosti nabízených stanic, tak v jejich původu. Tento způsob detekce proxy serveru odpovídá zadání práce. Jeho implementace do programu a výsledky měření jsou uvedeny v následujících kapitolách.

2.3 Honeypot

Honeypot je informační systém, který se úmyslně vystavuje útoku, aby mohl analyzovat útočníkovo chování. V roli útočníka může být člověk, nebo škodlivý software (tzv. malware). Honeypot je návnadou, pomocí které se detailně popíše fungování nějaké závadné činnosti. Zjišťuje se útočníkovo chování, používané nástroje, motivace a zneužité bezpečnostní díry. Jeho výstupy slouží k hledání nových postupů při obraně, pro aktualizaci antivirů, studiím ohledně počítačové bezpečnosti, atd. Nejde tedy o klasickou metodu detekce, která je převážně defensivní, ale o úmyslné vystavení se útoku.

Podle míry interakce, kterou honeypot útočníkovi poskytne, se rozdělují na [17]:

- Honeypot s vysokou interakcí *(High-Interaction Honeypot)* je systém, který nabízí útočníkovi reálný operační systém, služby a programy sním spojené. Tento druh honypotu musí být pečlivě odstíněn od dalších systémů, do kterých by mohl útočník neplánovaně proniknout. Jde o náročně proveditelný experiment,

který vyžaduje mnoho času a odborných znalostí. Na druhou stranu poskytuje velmi konkrétní a komplexní data, proto nachází své uplatnění při podrobném zkoumání chování a motivace útočníka.

- Honeypot s nízkou interakcí *(Low-Interaction Honeypot)* je nástroj, který emuluje prostředí pro interakci s útočníkem. Jde tedy o vytvoření virtuálního operačního systému, se všemi službami a nástroji uzpůsobenými k napadení. Tento druh honeypotu má malou šanci na odhalení, nebo zneužití k napadení dalších systémů vlastníka honeypotu. Honeypot s nízkou interakcí je vhodný pro klasickou analýzu útočníka, tedy ke sledování port scanu, příznaku útoku, analýze trendů a sběru malware.

V případě proxy serverů se honeypoty nazývají také proxypot, nebo open proxy honeypot. Proxypoty se tváří jako klasický open proxy server, tedy špatně nastavený proxy server, bez kontroly přístupu. Významná studie, popisující nastavení a aplikaci open proxy honeypotu, je popsána v kapitole 1.5.3 Spam, na straně 22.

3 PROGRAM PRO DETEKCI PROXY SERVERU

Tato práce se zabývá analýzou využití proxy serverů v oblasti internetové kriminality. Analýza má být provedena testováním černých listin, tzv. blacklistů, tedy seznamů podvodných, nebo škodlivých IP adres. Původní záměr byl zmapovat procentuální využití proxy serverů v internetu obecně, a to aktivními metodami. Sestavený program by se dotazoval přímo stanic a ze získaných informací by zjišťoval, jestli stanice je, nebo není za proxy serverem. Tento přístup narazil na řadu obtíží, protože proxy server je při aktivním dotazování obtížné rozeznat. Dalším problém je fakt, že nejčastějším uživatelským využitím proxy serveru je pokus o anonymizaci své internetové aktivity, tedy jejich nejširší působnost nalezneme v protokolech HTTP a HTTPS, což není možné zjistit metodou přímého dotazu na stanici. Původní záměr využít trasovací nástroje k odhalení proxy serveru v cestě mezi tazatelem a koncovou stanicí se tedy ukázal nad možnosti této práce.

Z těchto zkušeností vyplývá, že pro dosažení relevantních výsledků nejsou aktivní metody vhodnou cestou. Ideálním způsobem, jak dosáhnout cíle této práce, je analýza toku dat ze serveru, kam přistupuje velké množství uživatelů ze všech částí světa a kde je anonymita vhodná, nikoliv však potřebná. Tedy server, který by dokázal zmapovat přirozené chování uživatele na internetu a přitom nebyl nepřiměřeně napadán strojově (bots). Z hlediska serveru je mnohem jednodušší analyzovat vlastní provoz, a hledat v něm proxy server pomocí známých IP adres, čtením hlavičky, apod., než je hledání proxy serveru aktivním dotazováním jednotlivých stanic. Zřízení takového serveru je ovšem také nad rámec diplomové práce.

Další zvolenou metodou, jak zmapovat procentuální využití proxy serverů v internetu, bylo zachycení vzorku komunikace na serveru, který by se následně analyzoval pomocí seznamu známých proxy serverů. Pro tento účel vznikly dva skripty, které porovnávaly vstupní seznam zachycených IP adres s vlastní databází. Jeden skript běžel on-line na soukromé doméně a záměrem bylo rozšířit jej na webový nástroj. Ukázalo se, že je problém udržet vlastní databázi aktuální při omezených prostředcích, takže by měření podléhalo výrazné chybě už od měřícího nástroje. Zároveň by bylo třeba získat široké spektrum datového toku z delšího časového úseku, aby šlo o relevantní vzorek. Proto došlo ke změně přístupu k řešenému problému a pozornost byla zaměřena na seznamy IP adres již identifikovaných jako škodlivé, což znamená, že se vyskytly v nějakém typu kybernetického útoku.

3.1 Zvolená metoda pro sestavení programu Proxy checker

Jak již bylo zmíněno výše, je prakticky nemožné odhalit většinu proxy serverů aktivní metodou. Proto je běžnou praxí, že se proxy servery kontrolují metodou vyhledávání v databázi. Seznamy open proxy serverů jsou, z hlediska volného přístupu, nejdostupnější formou jak získat přehled o aktuálně běžících veřejných proxy serverech. Webové stránky, které nabízí vlastní seznamy proxy serverů, si udržují svoji databází aktuální a to ověřováním publikovaných IP adres alespoň jednou za den. Více o způsobu získávání open proxy serverů do těchto seznamů je v kapitole 1.6. Je běžnou praxí, že krom IP adresy a portu uvádí tyto seznamy další parametry, jako je rychlost, země původu a transparentnost.

Seznamy open proxy serverů velmi často fungují jako marketingový prostředek společnosti, která nabízí nějaký druh placené anonymizace, případně ochrany před spamem, DDoS a dalšími kybernetickými útoky. Komunikace s těmito seznamy je nejistá, společnosti neuvádí dokumentace k běžící webové službě a z logického důvodu není ochota nabízet svoje seznamy pro strojové zpracování. Vyhledávání bývá omezené prodlevou mezi jednotlivými dotazy, sadou dotazů, nebo počtem dotazů na den. Existují možnosti, jak si sestrojit vlastní seznam open proxy serverů, například nástroji zvanými proxy harvester, takový postup ale přidává další aplikaci třetí strany do cyklu ověřování IP, čímž se zvyšuje riziko chyby, nebo nepřesnosti.

Mnohem efektivnější se ovšem ukázalo využití Proxy Checker API [26]. Tato webová služba nabízí krom placené verze také veřejný IP proxy list a vlastní API, pomocí které může být zasláno více dotazů na ověření adresy, přičemž jediným limitem je výkonnost serveru. Proxy list je aktualizován každou hodinu, kdy si PHP skript ověřuje dostupnost serveru a další jeho parametry.

Vstupním souborem vytvořeného programu jsou data z blocklistu. Blocklist je seznam IP adres, které se účastnily kybernetického útoku, spamu apod., a bývá využíván administrátory serverů k blokování, aby předešli nežádoucím interakcím. K účelům práce byly využity seznamy z více zdrojů veřejně přístupných na internetu. Protože seznamy blocklistů obvykle nepřiřazují IP adrese port, který je potřebný při vyhledávání v databázi open proxy serverů, byly všechny IP adresy ověřovány pro 4 nejčastěji používané porty [27] proxy serverů: 80, 1080, 3128, 8080. Program je ale sestaven tak, že dokáže zpracovat smíšená vstupní data, co se portů týče. To znamená, že je možné do vstupního souboru zadat adresy s porty i bez portů a následně přiřadit porty, které se budou ověřovat pro samostatné IP adresy.

Požadavkem na skript byla schopnost umět automaticky zpracovat rozsáhlý vstupní soubor. Protože program pracuje na principu porovnávání známých open

proxy serverů se vstupním seznamem IP adres, předpokládá se rozsáhlý vstupní soubor. Rychlost ověřování jednotlivých IP adres záleží jen na rychlosti odezvy serveru.

3.2 Nástroj proxy checker API

Protože je webová API od proxyipchecker.com často zmiňována v textu, a zároveň je důležitou součástí programu Proxy checker, je jí vyhrazena samostatná kapitola, která shrne celkové fungování tohoto nástroje. Proxy IP checker jako služba běží na AJAX serveru. Disponuje rozsáhlou databází proxy serverů, kterou (dle textu na jejich webových stránkách) aktualizují každou hodinou. Pro získávání stanic do databáze používají několik PHP skriptů, bohužel podrobnější informace o tomto skriptu už neuvádějí. Nejpravděpodobnější bude technika skenování portů a možnosti připojení k špatně nakonfigurovanému serveru popsaná v kapitole 1.6 na straně 24.

Služby proxy IP checker jsou zaměřeny na kontrolu vlastního proxy serveru a možností, které takový server nabízí klientům. Tato služba kontroluje každou připojenou stanici a vrací informace o transparentnosti, IP, portu apod. Krom toho nabízí také kontrolu hlaviček a dává k dispozici vlastní seznam open proxy serverů a možnost ověřit si seznam zadaných IP adres.

Nabízené API slouží právě k zprostředkování dotazu na databázi IP adres a následné odpovědi. Kromě porovnání informací s databází provádí také dotaz na stanici a vrací zjištěnou rychlost spojení a dobu zpracování jednoho požadavku. Při měření těchto parametrů je nastaven časový interval (obvykle na 10s), pokud stanice včas neodpoví, považuje se za nedostupnou.

Pro komunikaci s API se používá HTTP metoda POST, která je více rozebraná v kapitole 3.4 na straně 35. V tělu dotazu se zadává dvojice IP:post ve tvaru `ip=127.0.0.1&port=1111`. Druhou možností je zažádat o pokročilejší měření přenosové rychlosti přidáním řetězce `&bw=1`, takže POST dotaz bude obsahovat `ip=127.0.0.1&port=1111&bw=1`. Dotaz bez detailního měření bandwidth je vzhledem k tématu práce vhodnější, server v tomto případě reaguje rychleji. Odpověď pak přichází ve formátu `response_time;speed;country;type`, jeho jednotlivé parametry jsou rozepsány níže v kapitole 3.3.

3.3 Struktura programu

Program je psaný v jazyce Python 2.7.11, spouští se pomocí konzole a pro jeho úspěšné spuštění je třeba mít nainstalovaný Python interpreter[1]. Program vyžaduje dva povinné vstupní parametry:

1. **Seznam IP adres** - je vstupní soubor, ze kterého bude program načítat IP adresy na zpracování. IP adresy mohou být zapsány společně s portem ve formátu 52.53.254.94:8083, nebo samostatně bez portu. Požadavkem je, aby na každém řádku byla jen jedna IP adresa. Vstupní soubor se zadává parametrem `-l`, tedy např.: `-l ip_list.txt`, případně s cestou k souboru `-l \data\ip_list.txt`.
2. **Seznam portů** - udává, které porty budou přiřazeny samostatným IP adresám. Pro každý zadaný port se otestují všechny samostatné IP adresy. Opět platí pravidlo, že každý port musí být zapsán na samostatný řádek.

Volitelným parametrem je mód `-m`, který udává, jak rozsáhlé budou informace vypsané o hledaných IP adresách. Jeho možnosti jsou následující:

1. `-m 1` je základní mód, který se spouští i bez zásahu uživatele. Jde o zjednodušený výpis, který obsahuje položky: proxy, transparency a country .
2. `-m 0` je obsáhlejší mód výpisu, který obsahuje všechny parametry zasílané z databáze: proxy, transparency, country, speed a response time.

Významy jednotlivých parametrů jsou:

- **Proxy** má dva stavy, které vyjadřují úspěšnost nalezení proxy serveru. Stav 1 říká, že kontrolovaná IP adresa je proxy server, stav 0 říká, že nejde o proxy. Tento stav se vyhodnocuje z přijatého stupně transparentnosti, pro úspěšné nalezení proxy serveru musí být transparentnost v rozsahu 1 až 3.
- **Transparency** udává stupeň anonymity, který proxy server zprostředkovává. Transparentnost 1 je transparentní proxy, 2 je pro anonymní proxy, 3 je elitní proxy server a 0 znamená, že nejde o proxy server. Význam transparentnosti proxy serverů je popsán v kapitole 1.4 na straně 17.
- **Country** je země původu, v které se hledaná IP adresa nachází.
- **Speed**, nebo také bandwidth, udává přenosovou rychlost, kterou proxy server vyhradí danému dotazu. Vrácená hodnota je v B\s.
- **Response time** je doba, kterou trvá proxy serveru vyřídit jednu operaci\dotaz.

Výstupem programu jsou soubory `result_`*`datum`*`.txt` a `status.txt`. Do souboru `result.txt` se ukládají všechny informace o testovaných IP adresách, přičemž míra obsáhlosti záznamu je ovlivněna zvoleným módem při spuštění. Soubor `status.txt` obsahuje dodatečné informace o délce běhu programu, počtu portů,

[1]Dostupný z oficiálních stránek Pythonu: `https://www.python.org/downloads/`

které byly přidány k IP adresám, všechny nalezené proxy servery a údaje o datu spuštění.

Spuštění programu pak může vypadat následovně:

```
>> python Proxy checker.py -l list.txt -p ports.txt -m 0
```

Na vývojovém diagramu (obr. 3.1) je znázorněn princip zpracování jednotlivých požadavků. Výsledky se průběžně vypisují do konzole a zároveň se zapisují do výstupních souborů. Jádrem programu jsou dvě funkce, přičemž main funkci `proxychecker` je vhodné rozdělit na tyto části:

1. **proxychecker** má vstupní parametry: odkaz na soubor se seznamem IP, načtené porty pro doplnění samostatných IP adres, mód a počet běhů, kterých zbývá do konce. Pro otevření vstupního souboru a jeho následného zpracování je osvědčenou praxí použít tvrzení `with`, které zaručuje bezpečné zavření souboru, jakmile doběhne přiřazený blok kódu, nezávisle na vzniklých chybách.
2. **POST request** je část hlavní funkce, která obstarává zasílání dotazu na webovou API. K tomu jsou využity knihovny `urlparse` a `httplib`.
3. **outformat** je funkce, která analyzuje odpověď serveru na dotaz ohledně IP adresy a vrací formát vhodný k zápisu podle zvoleného módu.
4. **arg.parse** je část kódu, která slouží pro administraci nápovědy a vstupních parametrů, ty se používají dál v kódu jako argumenty jednotlivých položek arg.parse modulu. Je třeba importovat knihovnu `argparse`.

3.4 Komunikace se serverem metodou POST

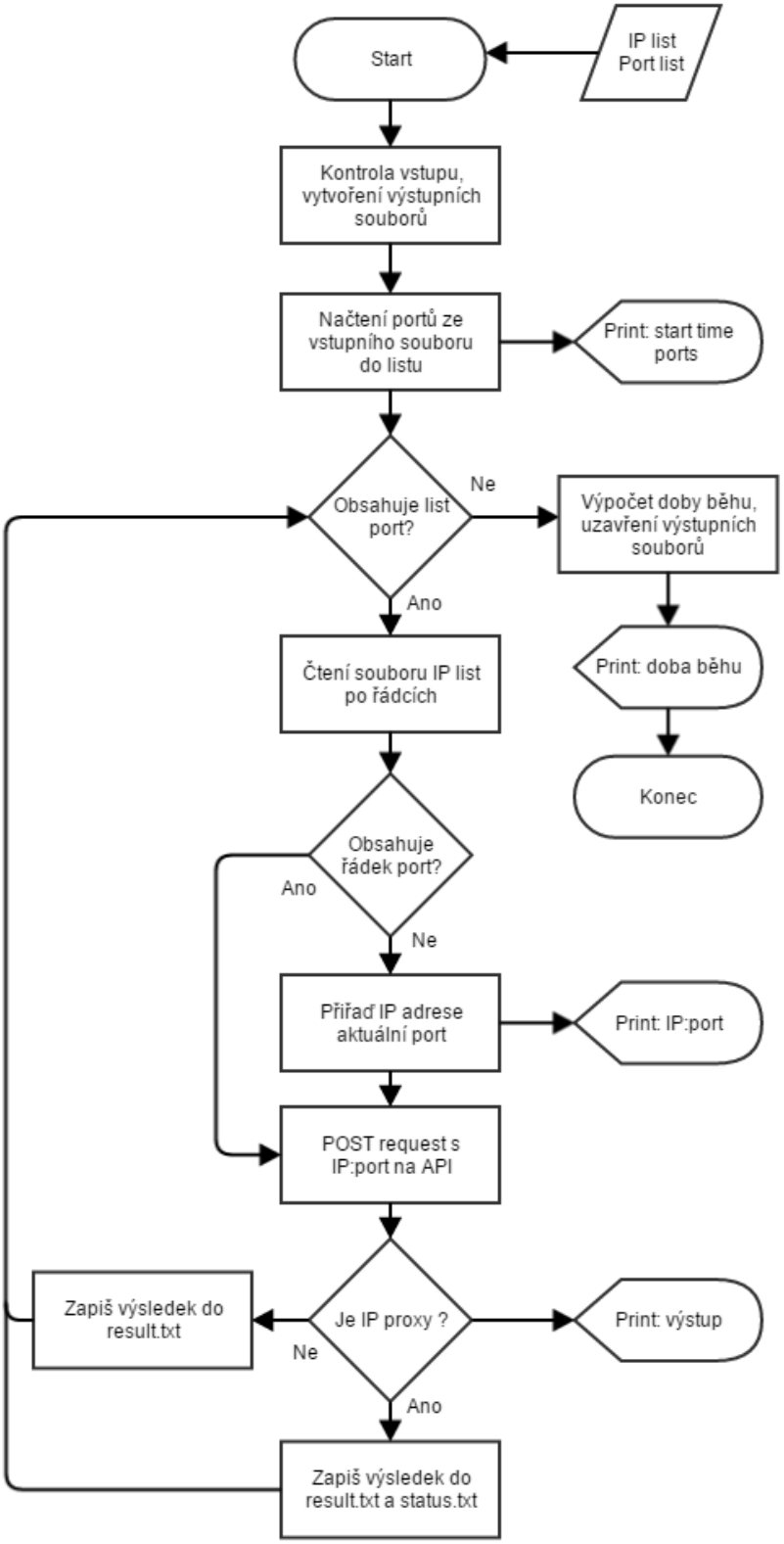
Proxy IP checker[2] je API, které program využívá pro zprostředkování dotazů k databázi proxy serverů. Podle dokumentace se zasílá HTTP POST dotaz na URL `http://proxyipchecker.com/api.html` s IP adresou v obsahu dotazu a to ve formátu určeným dokumentací: ip=127.0.0.1&port=1111 .

POST je jednou z metod HTTP dotazu a slouží k zasílání dat na server. Struktura HTTP dotazu je následující

První řádek v této struktuře se nazývá dotazový řádek a skládá se z těchto částí:

- Metoda – udává, jaký druh metody se používá. Nejčastějšími metodami HTTP dotazu jsou GET, POST a HEAD.
- Cesta – cesta je obvykle částí URL za doménou hostitele, používá se ale i plná cesta.
- Protokol – tato část obsahuje informace o použitém protokolu a jeho verzi. HTTP verze 1.1 je nejčastější pro moderní prohlížeče.

[2]Dokumentace je dostupná na `http://proxyipchecker.com/api.html`



Obr. 3.1: Vývojový diagram prochy checker skriptu.

Obr. 3.2: Struktura HTTP dotazu.

Následující řádky jsou hlavičky dotazu ve formátu *název: hodnota*. Hlavičky slouží pro dodání různých informací serveru o klientově prohlížeči, doplňcích, verzi Javy, apod. Např. User-agent udává informace o použitém prohlížeči a operačním systému tazatele, nebo hlavička Content-type, která slouží pro určení URL kódování dat. Stejně tak se v hlavičce přenáší cookie data. Přesto je většina hlaviček dobrovolná.

Poslední částí dotazu je obsah, což jsou data, které posílám serveru. Příklad POST dotazu pro kontrolu IP adresy na serverovou databázi vypadá následovně:

Výpis 3.1: Příklad HTTP POST dotazu.

```
POST http://api.proxyipchecker.com/pchk.php HTTP/1.1
User-Agent: Fiddler
Content-type: application/x-www-form-urlencoded
Host: api.proxyipchecker.com
Content-Length: 27

ip=158.106.79.99&port=3128
```

Odpověď začíná stavem 200 OK, což značí úspěšné zpracování dotazu a zaslání odpovědi. V programu využívám kontrolu tohoto stavu pro další zpracování odpovědi. Poté následují hlavičky potřebné pro komunikaci klient-server a v obsahu odpovědi jsou data nalezená v databázi. Jejich dekódování bude popsáno níže.

Výpis 3.2: Odpověď serveru.

```
HTTP/1.1 200 OK
Server: nginx
Date: Thu, 05 May 2016 17:22:52 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Vary: Accept-Encoding
Expires: Thu, 01 Jan 1970 00:00:01 GMT
Cache-Control: no-cache
Pragma: no-cache
Cache-Control: no-store, no-cache, must-revalidate,
post-check=0, pre-check=0

16
0.31;3;United States;1
0
```

3.4.1 Metoda POST v kódu programu

Tato kapitola se věnuje metodě POST tak, jak je použita v programu. Hlavičky nejsou v dokumentaci blíže specifikované, proto jsem použil jen základních parametrů, potřebných pro úspěšné spojení. Jde o hlavičky „Content type“ a „host“. K zaslání HTTP dotazu je použita knihovna `httplib`, pro parsování url je pak použita knihovna `urlparse` .

Výpis 3.3: POST request.

```
url = 'http://api.proxyipchecker.com/pchk.php'
urlparts = urlparse(url)
conn = httplib.HTTPConnection(urlparts.netloc, urlparts.
   port or 80)
conn.request('POST', urlparts.path, data.strip(),
    headers={'Content-type': 'application/x-www-form-
       urlencoded'})
response = conn.getresponse()
```

Jak je vidět z ukázky kódu, plná adresa pro POST dotaz je upravena až kódem. Program byl původně navržen tak, aby bylo možné vkládat více adres na různé servery poskytující open proxy listy. Kvůli nestabilitě výstupu a celkové neochotě

těchto serverů spolupracovat byl nakonec zvolen jiný postup, metoda ale zůstala. `urlparse` zde rozkládá části URL na jednotlivé prvky:

- `scheme` – udává protokol komunikace, nejčastěji jde o HTTP a HTTPS.
- `netloc` – network location je doménová adresa společně s portem, je-li zadán.
- `path` – je cesta za doménou hostitele, jak již bylo zmíněno výše u dotazového řádku.

V případě použité adresy `http://proxyipchecker.com/api.html` má proměnná `urlparts` hodnotu

```
ParseResult(scheme='http', netloc='api.proxyipchecker.com',
    path='/pchk.php')
```

`HTTPConnection` slouží k provedení transakce se serverem pomocí HTTP. Pro úspěšnou komunikaci potřebuje mít vyplněné parametry `host:port`. V případě, že není zadán port, dosazuje se základní port 80, což je i případ programu Prochy checker. Port není obsažen v hodnotě `netloc`, ani nebyl získán pomocí `urlparse.port` (toto pole je prázdné), proto je na konec řádku přidána pojistka, která přidá port manuálně. Port 80 za logickým operátorem *or* se dosadí pouze tehdy, pokud žádná předchozí funkce port nedosadí.

`request` je funkce spadající pod knihovnu `httplib`, která pomáhá určit jednotlivé parametry HTTP POST dotazu. První dvě položky slouží k vyplnění dotazového řádku (metoda a cesta), poté se zapisují data, která budou obsažena v obsahu dotazu. Proměnná `data` obsahuje vhodně upravenou IP adresu pro dotaz na server: `ip=158.106.79.99&port=3128`. Poslední položkou jsou zvlášť určené hlavičky.

3.5 Metoda arg.parse

Tento modul slouží k parsování[3] nastavení a argumentů zadávaných skrz konzoli a pro vytvoření uživatelsky přívětivého prostředí v rozhraní příkazového řádku. Pro jeho použití je třeba importovat knihovnu `argparse`. Pomocí modulu `argparse` je možné vytvořit nápovědu k užívání programu, nastavit, jaké vstupní parametry jsou povinné a jaké volitelné, umožňuje nastavit základní hodnotu parametrům, které uživatel nezadal atd. Parametry zadané pomocí příkazového řádku jsou pak získány jako argumenty `sys.argv`. Dále je vložen výňatek nastavení tohoto parseru z kódu programu:

[3]Parsování je proces, při kterém se rozebírá vstupní řetězec na jednotlivé položky, aby bylo možné získat užitečné informace a odfiltrovat přebytečné znaky, např. pozůstatky formátování.

Výpis 3.4: Argument parser.

```
parser = argparse.ArgumentParser(prog='Proxy checker',
                        description='This script looks into
                           your list of IPs, if there is a
                           Proxy serer. ',
                        epilog='All input files HAVE TO be
                           single-lined document with one
                           value on each line!')
parser.add_argument('-l', '--list', dest='filename',
   required=True,
                    help='Name of the file with IP
                       addresses you want to check. If FILE
                        is not located in same folder, '
                         'write full path. Example: C:\
                            Proxy Checker\Data set\ip_list.
                            txt ', metavar='LIST',
                    type=lambda x: is_valid_file(parser, x)
                       )
parser.add_argument('-p', '--ports', dest='portname',
   required=True,
                    help='Name of file or path to
                       configuration file for ports. One ip
                        list will be checked for all '
                         'inserted ports.',metavar='PORTS',
                    type=lambda x: is_valid_file(parser, x)
                       )
parser.add_argument('-m', '--mode', dest='mode', default=1,
    type=int,
                    help='mode 0 set complete output
                       information, mode 1 set simple
                       output info(it's default mode)',
                    metavar='MODE')
```

Prvně je třeba vytvořit objekt parser, což se provádí pomocí `ArgumentParser`. Do objektu `parser` se pak ukládají všechny informace potřebné k parsování informací z příkazového řádku do dat čitelných Pythonem.

```
>>> parser = argparse.ArgumentParser(prog='Proxy checker',
   description='Check my IP.')
```

Objekt vytvořený pomocí `argparse.ArgumentParser` může mít více parametrů, v kontextu této práce byly využity následující:

- `prog` – jméno programu, např. 'Proxy checker'. Zobrazuje se v nápovědě.
- `description` – jde o krátký popis programu, zobrazený v nápovědě hned pod názvem programu.
- `epilog` – slouží k zapsání dodatečných informací k programu na konec nápovědy. Tento text bude zobrazen až za argumenty a jejich vysvětlením.

Následně se přidávají argumenty, s kterými bude program operovat, což se provádí použitím metody `add_argument()`. Tato metoda udává, jak bude `ArgumentParser` zacházet se vstupem z příkazového řádku pro tento konkrétní parametr, aby jej převedl do objektu pro Python. Zavolaná metoda `parse_args()` pak vrátí objekt s atributy odpovídající jednotlivým položkám `add_argument()`. Argumenty se dělí na povinné a volitelné. Povinné argumenty musí být vyplněny při každém spuštění programu. Volitelné, jak už z názvu vyplývá, nemusí být zadány vždy.

Výpis 3.5: Příklad nastavení argumentu pro parser.

```
parser.add_argument('-l', '--list', dest='filename',
   required=True,
                    help='This is list of IPs to check. ',
                       metavar='LIST',
                    type=lambda x: is_valid_file(parser, x)
                       )
```

V případě nastavení, jak je zobrazeno v kódu 3.5, bude mít objekt `parser` atribut `filename`, který bude odkazovat na soubor. Program Proxy checker využívá při nastavení argumentu těchto parametrů:

- `flag` – nastavuje, jak se bude argument zadávat v příkazové řádce. Pokud je vlajka nastavena pomocí jedné, nebo dvou pomlček, jde o argument nezávislý na pořadí ('-l', '– list'). Pokud je třeba nastavit poziční argument, zapisuje se bez použití pomlček ('list').
- `dest` – jméno atributu, který bude přidán k objektu `parser`. Pomocí tohoto jména se pak určuje, který atribut požaduji od funkce `add_argument()`. Př.:

  ```
  args = parser.parse_args()
  with open(args.filename, 'r') as ...
  ```

- `required` – nastavuje, jestli jde o argument povinný, nebo volitelný.
- `help` – slouží k vypsání informací o argumentu, pokud se zavolá nápověda.
- `metavar` – slouží k jednoznačné identifikaci argumentu, například při volání nápovědy. V základu se používá hodnota `dest`, pokud není metavar nastaven jinak.
- `type` – tento parametr určuje, v jakém formátu přichází argument z příkazové řádky. Pokud není nastaveno jinak, bere se vstup jako klasický string. Pomocí

parametru `type` je možné ověřovat vstup jestli je zadáván správně. Z této části je také možné volat funkce. V kódu programu se například volá funkce pro ověření správného souboru na vstupu `is_valid_file`, viz 3.5.

3.6 Zpracování výstupu

Proxy checker API posílá výstup ve formátu [response_time;speed;country;type], např. `0.69;1;Israel;3`. Protože je odpověď standardizovaná, nebylo třeba získávat data pomocí regulárních výrazů, či jiné pokročilejší metody extrakce dat. Rozklad řetězce odpovědi probíhá pomocí indexů. Jako styčné body jsem zvolil středník, které vyhledávám pomocí `for` cyklu s funkcí `enumerate`, která umožňuje vyhledat opakující se položku. Získané indexy středníků jsou postupně zapisovány `for` cyklem do pole `indexes`. Pomocí známé polohy oddělovače (středník) pak rozděluji řetězec znaků a získávám hodnotu jednotlivých parametrů (ty jsou popsány výše v kapitole 3.3 Popis programu).

Výpis 3.6: Zpracování odpovědi ze serveru.

```
indexes = [i for i, oddelovac in enumerate(text) if
   oddelovac == ';']
        editedtext = 'Proxy: ' + answer + '\t' + '
           Transparency: ' + text[indexes[2] + 1:] + '\t' +
            'Country: ' + \
                    text[indexes[1] + 1:indexes[2]] + ' \
                       tSpeed[B/s]:' + text[indexes[0] +
                       1:indexes[1]] + \
                    '\tResponse time: ' + text[:indexes
                       [0]]
```

Při sestavování řetězce `editedtext`, jak je vidět ve výňatku z kódu 3.6 na druhém řádku, je odkaz na návratovou hodnotu jiné funkce – `answer`. Jde o jednoduchou funkci, která si načítá pomocí již získaných indexů hodnotu pro transparentnost proxy serveru a podle toho pak vrací odpověď, jestli jde o proxy. Pokud je hodnota transparency vyšší jak 0, znamená to, že u IP adresy byl zjištěn jeden z druhů nabízené anonymity (transparentní pro 1, anonymní pro 2 a elitní pro 3). Tím je potvrzen proxy server. Pokud je hodnota 0, je tato IP adresa vyhodnocena jako nefunkční proxy server, nebo jako IP adresa jiného zařízení. Tento postup se ukázal jako nejefektivnější. Je totiž poměrně častým jevem, že stanice ze seznamu open proxy serverů je dostupná, přesto ji nástroje pro detekci proxy serveru neidentifikují.Předpokládané důvody těchto zjištěný jsou popsány na konci kapitoly 5, která diskutuje výsledky měření proxy serverů.

4 VYHODNOCENÍ PRÁCE PROGRAMU PROXY CHECKER

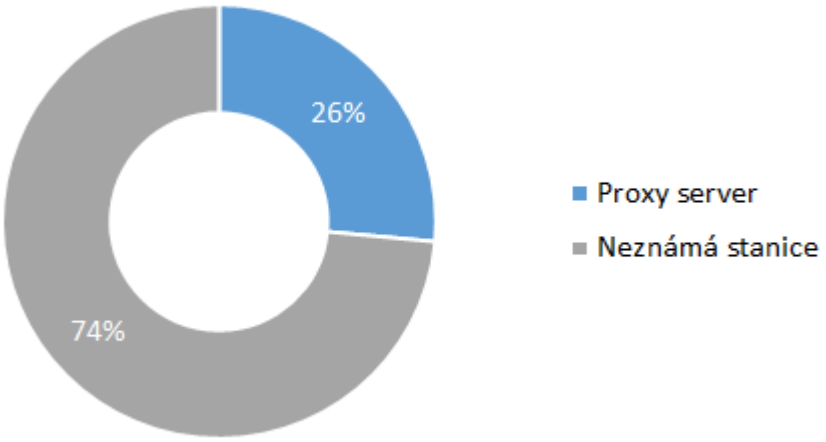
Při měření funkčnosti programu jsem využíval veřejných seznamů open proxy serverů, aby bylo možné zpětně dohledat a ověřit zjištěné informace. Velkou nevýhodou těchto seznamů je nestabilní životnost inzerovaných stanic, a tak – i přes ujištění, že každá stránka ověřuje svoje stanice co hodinu – je běžné, že více jak polovina IP adres není platná. Je třeba brát takové chování v úvahu a podle toho nahlížet na výsledky jednotlivých měření. V kapitole 4.1 je shrnuto základní testování efektivity programu. Tento test jsem následně rozšířil na všechny parametry, které je možné zjistit o stanici. Rozbor naměřených dat je obsažen v kapitole 4.2. Kapitola 4.3 zastřešuje rozsáhlejší testování pro více nástrojů, jejich vzájemné porovnání a zhodnocení obecné efektivity vytvořeného programu.

4.1 Základní měření funkčnosti programu

Základní měření funkčnosti programu je klasický test, který probíhá po sestavení programu, a je zaměřené na správné určení základních parametrů stanic – transparentnost, zemi původu a především, jestli je stanice proxy server. Výsledky testu jsou zde zmíněny proto, že dobře ilustrují kolísavou životnost open proxy serverů. Pro test bylo použito 306 IP adres, z nichž bylo 81 identifikováno jako proxy server a zbylých 225 jako neznámé stanice. Pro toto měření bylo relevantní především zjištění dalších nalezených informací – tedy anonymita proxy serveru a místo původu. Získané parametry byly ručně porovnány s informacemi, které udávala zdrojová stránka, a byla zjištěna 100 % shoda.

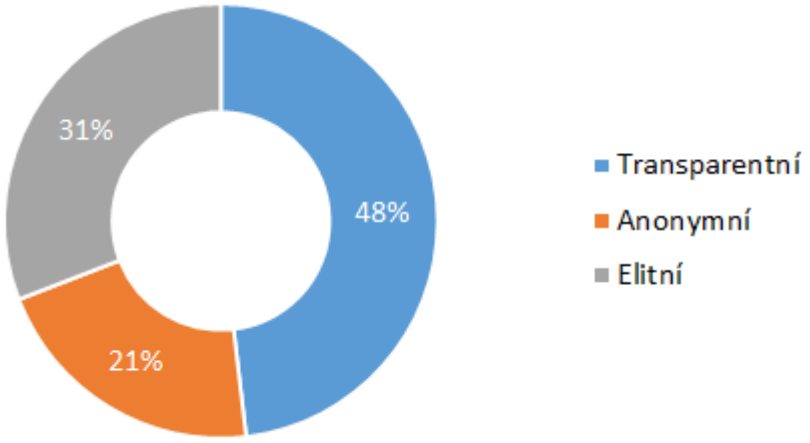
Díky zjištěním z tohoto měření jsem začal věnovat větší pozornost aktivitě stanic, protože nebylo zprvu jisté, co způsobilo tak nízkou úspěšnost při hledání proxy serverů. Bylo třeba určit, jestli chyba nastala na straně programu, nebo seznamu open proxy serverů.

Kvalitní seznamy open proxy serverů jsou často pouze nadstavba placené služby, kterou webová stránka poskytuje. Je v zájmu poskytovatele placené služby, aby jeho reklama – volně využitelný produkt, dosahoval vysoké kvality. Z toho důvodu jsou úspěšné veřejné seznamy zaměřené na anonymní a elitní proxy servery, které poskytují uživateli vysokou míru anonymity. HMA! patří k nejznámějším poskytovatelům open proxy serverů, proto je zajímavý výsledek zjištěného rozložení transparentnosti, které, jak je vidět na grafu 4.2, ukazuje nejčetnější výskyt pouze transparentních proxy serverů. Důvodem takového výsledku není omezená nabídka ze strany seznamu adres. Proxy servery ze třídy transparentních a anonymních jsou snadno zjistitelné



Obr. 4.1: Poměr zjištěných a nezjištěných stanic.

z informací v hlavičce, naopak elitní proxy servery se odhalují velmi těžko. Protože HMA! má ve své nabídce nejčastěji servery poskytující anonymitu „High + Keep Alive“, což je označení používané pouze na jejich stránkách, předpokládám, že velká část jimi inzerovaných proxy serverů nedokáže běžný nástroj pro zjišťování proxy serverů odhalit. Podrobněji je tato problematika popsána v následující kapitole 4.2



Obr. 4.2: Rozložení transparentnosti nalezených serverů.

4.2 Pokročilé měření funkčnosti programu a parametrů proxy serveru

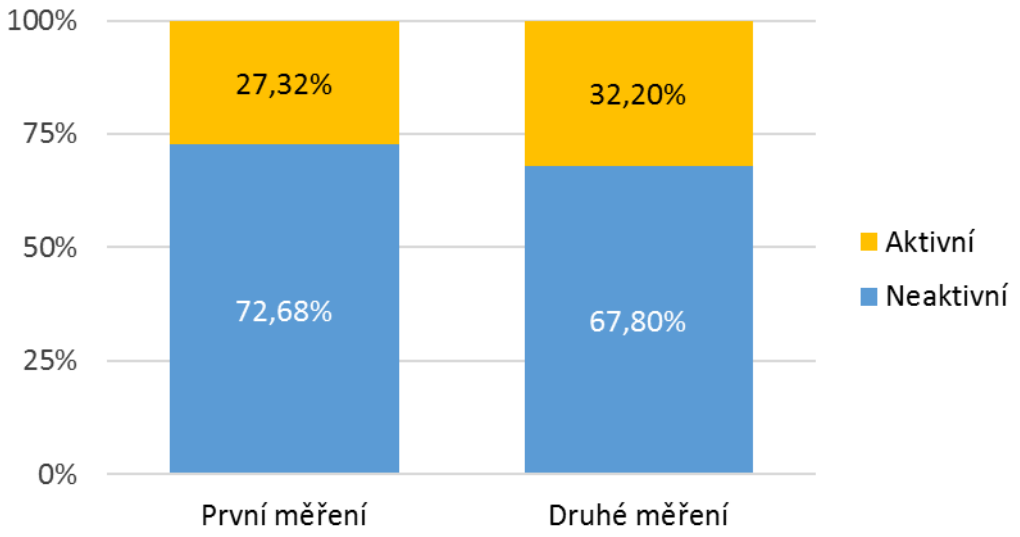
Toto měření mělo za cíl zjistit a ověřit podrobnější informace o open proxy serverech. Důraz se začal klást na parametry *rychlost (speed)* a *doba odezvy (response time)*, které jsou získávány z IP adres nezávisle na tom, jestli jsou označeny za proxy servery. Protože měření rychlosti má nastavený timeout, který měřená stanice nemusí splnit z důvodu špatného spojení, nikoliv z důvodu, že není aktivní, není tento parametr stěžejním při určování on-line stanic. Doba odezvy je na druhou stranu snadněji zjistitelný parametr, proto jsou informace o aktivních a neaktivních stanicích získané z navráceného parametru doby odezvy. Pokud stanice neodpoví ani na měření doby odezvy, je označena za neaktivní. Je standardním jevem, že je ke stanici zjištěna doba odezvy ale hodnota rychlost nikoliv.

Pro první měření bylo použito 200 IP adres ze seznamu open proxy serverů. Na 6 adres nepřišla odpověď vůbec, ze zbylých 196 IP adres bylo zjištěno 53 aktivních stanic, z toho 27 určeno jako proxy server. V porovnání s dřívější zkušeností se zdál výskyt identifikovaných stanic ještě zoufalejší, protože oproti základnímu testu, který nalezl 26 % stanic, bylo nyní zjištěno pouhých 14 %. Ovšem díky přidanému měření aktivity stanic se ukázalo, že jsem čerpal z očividně neaktuálního seznamu – téměř 73% stanic nebylo aktivních. V tomto světle je efektivita programu uspokojivá, z 53 stanic identifikoval 27, což je přibližně polovina.

Tab. 4.1: Výsledek třetího měření funkčnosti programu.

Měřeno	**Zjištěných [ks]**
Celkem stanic	194
Neaktivních stanic	141
Aktivních stanic	53
Proxy serverů	27
Zjištěna rychlost	39
Zjištěna doba odpovědi	53

Ukázalo se, že pro relevantní měření je třeba se zaměřit především na zdroj testovaných dat, protože volně dostupné seznamy open proxy serverů trpí osobitými nedostatky. Vyloučil jsem ze zdrojových dat web HMA! a vytvořil seznam dat obsahující 1 000 unikátních IP adres. Do výběru se dostaly jen stanice, u nichž proběhl



Obr. 4.3: Poměr aktivních a neaktivních stanic z obou měření.

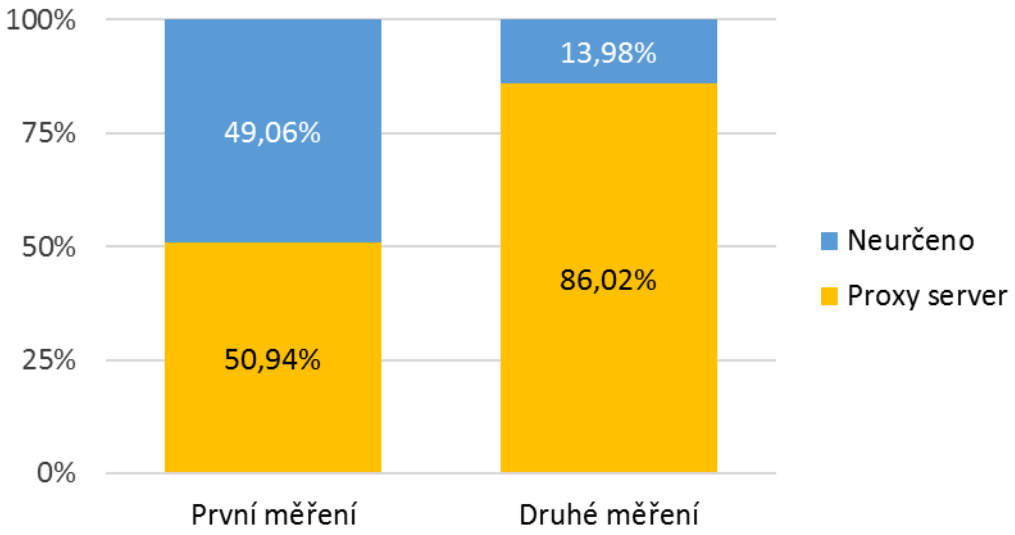
test životnosti alespoň do 2 hodin[1]. Zároveň byly vyloučeny stanice z dřívějších seznamů s označením „High + KA“. Výsledek byl pozitivnější, než v předchozích případech. Z celkového počtu 1 000 IP adres bylo aktivních 322 stanic a z toho se podařilo určit 277 proxy serverů. Efektivita nálezu činí 86% nalezených stanic.

Tab. 4.2: Výsledek druhého měření funkčnosti programu.

Měřeno	**Zjištěných**
Celkem	1000
Naktivních	678
Aktivních	322
Proxy server	277
Rychlost	131
Doba odpovědi	322

V grafu 4.4 je zobrazena nestabilní životnost open proxy serverů v porovnání obou měření. Při pečlivějším výběru stanic v druhém měření je znát poměrový nárůst aktivních stanic vůči stanicím z prvního měření. Stále jde ze strany seznamu proxy serverů o velkou úmrtnost stanic. Jak potvrdily i další testy, je 30 % až 40 % životnost stanic běžným jevem. Druhý graf 4.3 zobrazuje úspěšnost identifikace open proxy serverů vůči všem aktivním stanicím, která v rozsáhlejším testu dosáhla 86 %.

[1]Test životnosti provádí vždy skript webové stránky, z které čerpám data. Jsem tedy odkázaný na důvěru v jejich měření.



Obr. 4.4: Počet nalezených proxy serverů vůči aktivním stanicím.

4.3 Srovnání efektivity s jinými nástroji

V tomto testu byla data, stejně jako v předchozím měření, vybrána mimo listy HMA! a měření bylo rozloženo na 10 dní. Každý den proběhl test 100 IP adres, u kterých proběhl test dostupnosti nejpozději do 1. hodiny. Do ověřování celkem 1 000 IP adres byly zároveň zapojeny tři další nástroje pro hledání proxy. Měření mělo za cíl porovnat úspěšnost nalezení proxy serverů vůči ostatním nástrojům, které slouží jako nadstavba pro placené služby v oblasti seznamů proxy serverů i pokročilejšího a přesnějšího ověřování proxy serverů. Jde tedy o nástroje profesionální. Jejich seznam je uveden v tabulce 4.4, která shrnuje výsledky jednotlivých sad IP adres, i celkovou úspěšnost vůči všem stanicím. Data ve sloupcích nástrojů udávají počet identifikovaných serverů, které tyto nástroje našli v dané sadě. Tentokrát nebyla brána v potaz aktivita všech stanic, jako u předchozích měření, protože ostatní nástroje ve své neplacené variantě nezobrazují hodnoty nerozpoznaných stanic. Svorně je označují za mrtvé a další informace nejsou poskytnuty.

Tabulka 4.3 obsahuje seznam webů, ze kterých jsem čerpal data pro měřící sadu. Webovým stránkám je přiřazen klíč, podle kterého jsou pak zapsány v tabulce 4.4.

Úspěšnost nalezení stanic byla v tomto měření nejvyšší ze všech měření. Průměrná úspěšnost činí 38 % , přičemž 3 ze 4 nástrojů nalezlo přes 40 % stanic. Jediný nástroj od freeproxy.ru klesl na 29,30 %. Příčinou vyššího počtu nalezených stanic je z velké části pečlivější výběr dat, tedy rozložení testu na delší časový úsek s menšími bloky IP adres. Pro snazší srovnání efektivity vytvořeného programu s ostatními nástroji jsem vytvořil graf4.5. Kvůli lepší přehlednosti jsem propojil body měření čarou, cílem grafu je zobrazit shodný průběh měření a jasně vymezit odchylky. V grafu je

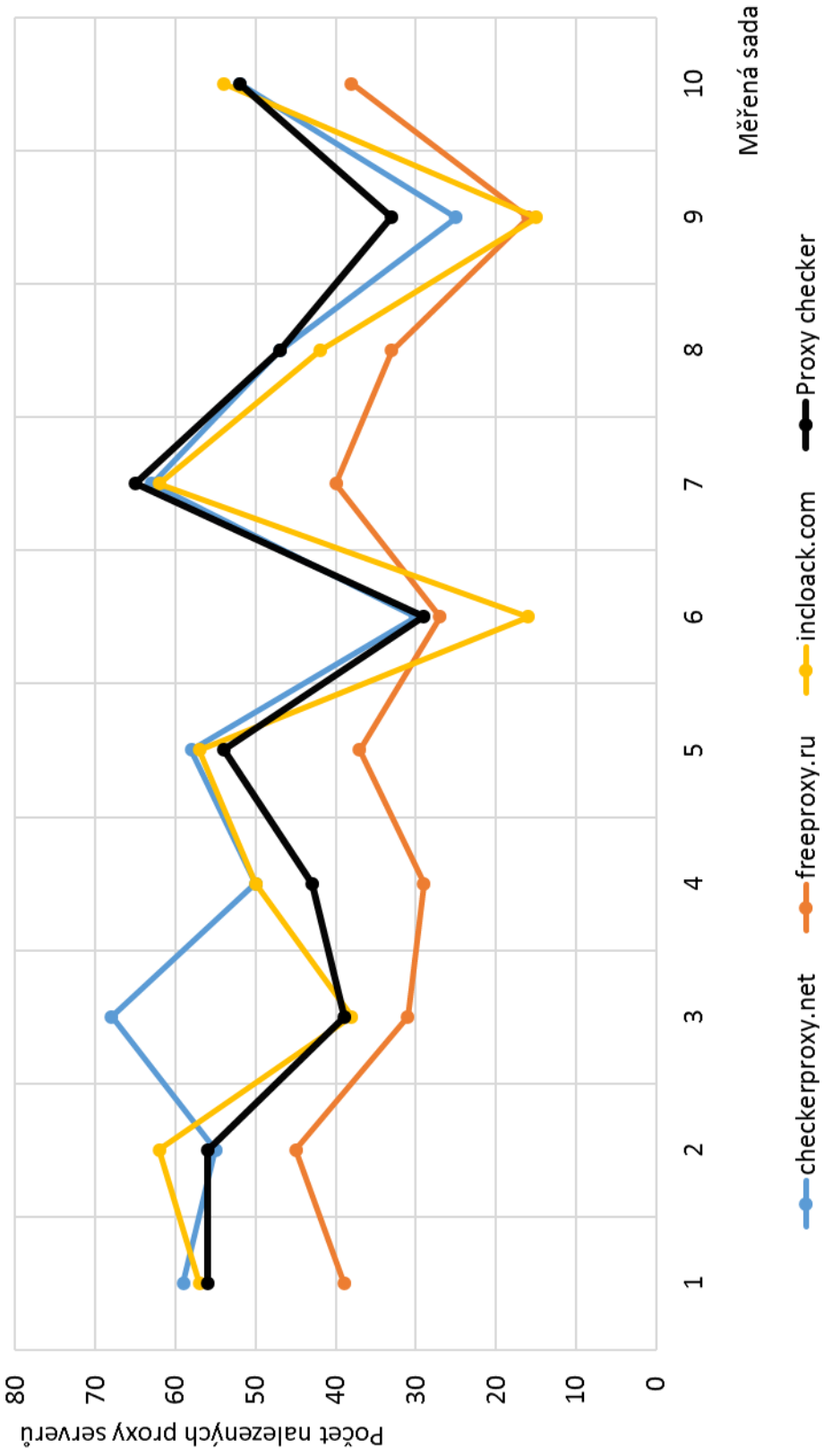
Tab. 4.3: Seznam použitých webů s nabídkou proxy serverů.

Klíč	**Adresa webu**
A	free-proxy.cz
B	free-proxy-list.net
C	samair.ru

tedy třeba sledovat odlišnosti jednotlivých průběhů mezi sebou, než samotné propady a vzestupy, protože každá sada IP adres byla pro všechny nástroje stejná. Kvalitu těchto vzorků nebylo možné dopředu odhadnout, přesto si udržovali průměrnou životnost po celou dobu, krom 6. dne, který byl poněkud nešťastným pro zdroj dat – free-proxy.cz. Z výsledků měření vyplývá, že sestavený program Proxy checker dosahuje stejné efektivity, jako ostatní dostupné nástroje, v jednom případě i vyšší (freeproxy.ru).

Tab. 4.4: Počet detekovaných stanic pro jednotlivé měřící sady.

Zdroj dat	**Nástroje pro zjišťování proxy serveru**			
	checkerproxy.net	freeproxy.ru	incloack.com	**Proxy checker**
A	59	39	57	56
B	55	45	62	56
C	68	31	38	39
B	50	29	50	43
A	58	37	57	54
A	30	27	16	29
B	63	40	62	65
B	47	33	42	47
B	25	16	15	33
A	52	38	54	52
Úspěšnost	**42,40 %**	**29,30 %**	**40,50 %**	**41,00 %**



Obr. 4.5: Průběh ověřování proxy serverů pro různé nástroje.

5 VÝSLEDKY MĚŘENÍ

Dle zadání práce jsem sestavil seznam 15 256 unikátních IP adres, které byly v roce 2015 zapsány na některý ze zveřejněných blacklistů. Doporučením bylo použít veřejně dostupné černé listiny pro spam a phishing. Krom těchto černých listin jsem tvořil seznam z dalších škodlivých adres, od nichž bylo zaznamenáno skenování portů, nebo z nich byl veden útok na některou z následujících služeb: FTP, SSH, Telnet, SMTP, POP3, IMAP, SNMP DDoS útok, DDoS DNS amplifikační útok. Seznam útoků: útoky bruteforce k prolomení hesla, amplifikační DDoS zneužívající DNS, NTP a SNMP, spam, jiné DDoS útoky.

Předpokladem bylo nalezení open proxy serverů v tomto rozsahu a následná analýza využití open proxy serverů v ilegálních, nebo neetických činnostech za rok 2015. Jak již bylo zmíněno výše, černé listiny uvádějí holé IP adresy bez portů, přičemž znalost portu je nezbytná pro identifikaci proxy serveru[1]. Proto jsem každou IP adresu ověřoval pro čtyři porty, které jsou nejčastěji používány proxy servery [17] [27]. Jejich přehled je v tabulce 5.1. Celkem tedy proběhlo 61 024 dotazů na detekci proxy serveru. Seznam použitých IP adres je v elektronické příloze.

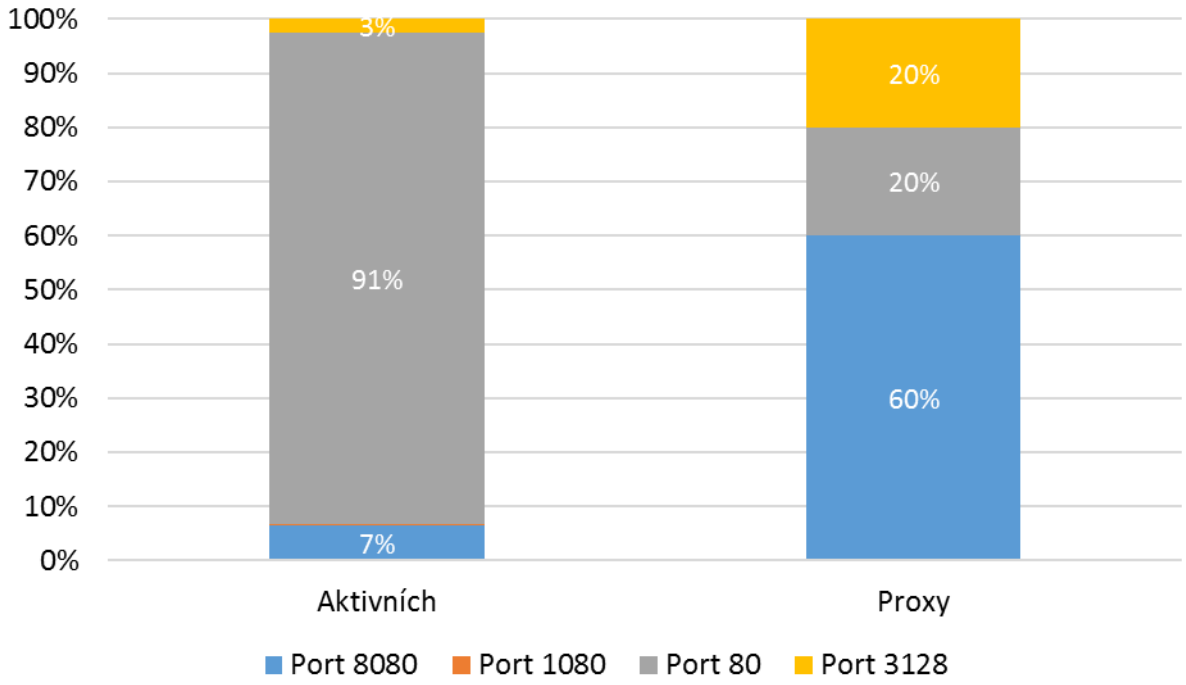
Tab. 5.1: Přehled ověřovaných portů .

#	**Port**	**Protokol**	**Obvyklá služba**
1	80	HTTP	HTTP
2	3128	HTTP	Squid
3	1080	SOCKS	SOCKS
4	8080	HTTP	HTTP

V tabulce 5.2 je přehled výsledků měření pro jednotlivé porty. Protože server neodpověděl, nebo zaslal chybnou odpověď, na 1/3 všech dotazů, jsou data hodnocena vůči získaným odpovědím. Důvod těchto výpadků není jasný, může jít o dočasné přetížení serveru, nebo špatnou IP adresu. Chybná odpověď se náhodně vyskytovala i v dřívějších měřeních funkčnosti, přesto neovlivnila srovnávací test nástrojů, viz kapitola 4.3.

Z naměřených dat je vidět, jak málo bylo nalezeno proxy serverů, či aktivních stanic obecně. Bylo zjištěno pouze 1 081 aktivních stanic, což jsou 2,3 % z celkového seznamu. Detekovaných proxy serverů je 5, což je zcela zanedbatelné číslo vůči rozsahu měření. Přehled výsledků měření zobrazuje graf 5.1, který vyjadřuje poměr zjištěných hodnot vůči jednotlivým portům.

[1] Věřené seznamy proxy serverů jsou psány vždy v kombinaci IP adresa: port, proto je zapotřebí znát celý řetězec při hledání v těchto databázích.



Obr. 5.1: Poměr aktivních a identifiovaných stanic vůči portům.

Tab. 5.2: Zjištěné počty stanic pro jednotlivé porty.

Port	**Celkem st.**	**Neaktivních st.**	**Aktivních st.**	**Proxy**
8080	11 321	11 250	71	3
1080	10 550	10 549	1	0
80	13 051	12 070	981	1
3128	10 876	10 848	28	1
Souhrn:	**45 798**	**44 717**	**1 081**	**5**
Procentuální poměr:		**97,64 %**	**2,36 %**	**0,01 %**

Zprvu se zdály naměřené hodnoty jako chybné, proto jsem v lednu 2016 provedl několik náhodných měření černých listin, které jsem získal z `www.openbl.org/lists.html` a `www.nothink.org/honeypots.php`. Tyto webové stránky přidávají do svých černých listin IP adresy průběžně, nejde tedy o jednorázově sestavené seznamy, nebo ukončené projekty, což byl případ předchozího zdroje dat. V obou případech je možné si volit data přidaná za poslední měsíc a týden, u openbl.org i za posledních 24 hodin. V těchto testech jsem používal data za poslední týden, případně za den. Výsledky byly totožné, jako u hlavního měření. Proxy servery se mi nepodařilo najít v relevantní míře vůči obsahu testovaných dat.

Hledání open proxy serverů v seznamu škodlivých IP adres je zpětný postup, kdy útok již proběhl a hledání probíhá v určitém časovém zpoždění od zaznamenané události. Snad největší míru neúspěchu zanáší do takového měření životnost open proxy serverů, viz kapitola 4.1 na straně 43. Jakmile majitel zneužitého serveru zjistí, že má na stanici podivně velký provoz, pravděpodobně server odstřihne, nebo zabezpečí. Taková interpretace by vysvětlovala zmiňovaný fenomén, kdy se i v seznamu proxy serverů vyskytuje větší část aktivních stanic, než zjistitelných open proxy serverů. Původní open proxy server je stále dohledatelný, pouze není možné se na něj připojit bez autorizace.

Mezi open proxy servery ovšem nepatří jen špatně nakonfigurované proxy servery, jejich součástí jsou honeypoty, proxy servery patřící společnostem, které nabízí placené služby, apod. Některé weby dokonce upozorňují, že open proxy servery s nejvyšší anonymitou mohou být honeypoty, nebo nedůvěryhodné servery. Jejich kolísavá životnost sice významnou měrou stěžuje hledání, je však nepravděpodobné, že by byla jedinou příčinou nenalezení téměř žádných proxy serverů.

Je třeba brát v potaz, že zkušenější útočník má k dispozici spolehlivější systémy pro anonymizaci, než je seznam open proxy serverů (viz kapitola 1.5 na straně 19). Nabízí se např. botnety, VPN, placené proxy servery, Tor. Open proxy servery budou mít vyšší zastoupení u běžného uživatele, který chce skrýt anonymitu z důvodů jiných, než je kybernetický útok, případně u útočníka s menšími zkušenostmi.

Zvláštní kapitolou je spam. Protože jsou IP adresy open proxy serverů běžně blokovány administrátory kvůli spamu, je diskutabilní, jakou měrou jsou stále využívány ke spamu. Konkrétně u blacklistů[2] zahrnující spam (např. z `www.openbl.org/lists.html`) jsem však očekával, že získám více proxy stanic, výsledky však hovoří jinak. Otázkou je, jak často se podaří zaznamenat adresu, odkud spam přichází, protože spam se šíří masově z mnoha zdrojů a jen jednou z více možných metod, jak šířit spam, je použití seznamu open proxy serverů.

[2]Vybíral jsem takové blacklisty, které nebyly složeny čistě ze seznamu open proxy serverů a open relay serverů, jak se stalo běžnou praxí u spam blacklistů. Tyto blacklisty totiž nejsou založeny na IP adresách, které opravdu provedly útok, ale na IP adresách, které by ten útok provést mohly.

6 ZÁVĚR

Cílem práce bylo provést odhad, kolik stanic v internetu komunikuje pomocí proxy serveru, a to s využitím seznamu podvodných IP adres, zahrnujících spam a phishing. Pro účely měření byl seznam rozšířen na větší rozsah podvodného a nelegálního chování na internetu. Pro detekci proxy serveru byl sestaven program Proxy checker, který porovnal vytvořený seznam IP adres s kvalitní databází známých proxy serverů. U stanic jsem také zjišťoval, jestli jsou stále aktivní. Po prohledání databáze o 15 256 IP adresách na 4 nejčastěji používané porty u proxy serverů (80, 3128, 1080, 8080) bylo nalezeno pouze 5 proxy serverů (podrobný rozbor je v kapitole 5, strana 50). Důvodem takového výsledku může být nestabilita veřejných proxy serverů, přechod k pokročilejším metodám anonymizace útočníků, či potřeba použít pokročilejší metody k identifikaci zneužitých proxy serverů.

V této práci jsem shrnul poznatky ohledně proxy serverů, jejich pozici v komunikaci a působení v rámci internetové kriminality. Teoretická část rozebírá princip fungování proxy serveru, navrhuje metody vhodné pro jeho detekci a shrnuje známé výskyty proxy serverů u různých druhů útoku. Následně je rozebrána technická stránka vyhledávání proxy serveru v seznamu IP adres, technický popis jednotlivých komponent a diskuze měření efektivity vzniklého programu. V poslední kapitole jsem zobrazil a komentoval výsledky měření.

Téma proxy serverů v internetu je málo zmapované jak z hlediska kvalitní literatury, tak z hlediska výzkumů. Je třeba otestovat jednotlivé metody detekce, aby se našla účinná cesta pro analýzu využití proxy serverů, a to jak pro nelegální a neetické aktivity uživatelů, tak pro použití běžným uživatelem. Zde se nabízí téma analýzy chování a motivace uživatele, který se rozhodl anonymizovat svůj pohyb v internetu. Veřejné proxy servery totiž představují pro nezkušeného uživatele bezpečnostní riziko v podobě odposlechu komunikace nebo infekce malwarem.

V případě analýzy chování útočníka, nebo zneužití proxy serverů v reálném čase, je nutné použít pokročilejší techniky detekce. Vhodnou metodou, která nevyžaduje náročné profesionální technologie, se ukázalo použití honeypotů s nízkou interakcí. Open proxy honeypoty nabízí široké spektrum možností testování a analýzy. Důležitý je sběr dat po delší časový úsek a sledování celkového provozu. Pro identifikaci probíhajícího typu útoku je vhodné se zaměřit také na odchozí komunikaci a porty, na které je směrovaná.

Tato práce ukázala úskalí, které doprovází detekci proxy serverů pomocí metody ověřování veřejných seznamů. Svými výsledky dokazuje obtížnost dohledání proxy serveru, který má sloužit k anonymizaci uživatele. Alternativních metod k odhalení proxy serveru není mnoho. Jakmile útočník použije proxy server, který nezasílá pravdivé informace o sobě, ani o uživateli, je téměř nemožné jej odhalit.

LITERATURA

[1] *FTP and proxy server* [online]. 2014, poslední aktualizace 10. 7. 2014 [cit. 4. 4. 2016]. Dostupné z URL: <http://www.idc-online.com/technical_references/pdfs/data_communications/Ftp_and_Proxy_server.pdf>

[2] BARNETT, R.C. *Open proxy honeypots* [online]. 2004, poslední aktualizace 30. 3. 2004 [cit. 4. 4. 2016]. Dostupné z URL: <http://honeypots.sourceforge.net/open_proxy_honeypots.pdf>.

[3] *Turris – uživatelská dokumentace* [online]. [cit. 4. 4. 2016]. Dostupné z URL: <https://www.turris.cz/doc/navody/squid>.

[4] FERRO, G. *Fast Introduction to SOCKS Proxy* [online]. 2008, poslední aktualizace 6. 6. 2008 [cit. 28. 4. 2016]. Dostupné z URL: <http://etherealmind.com/fast-introduction-to-socks-proxy/>.

[5] JEŘÁBEK, J. *Pokročilé komunikační techniky* Brno: 3. 4. 2014. 175 s. [cit. 4. 4. 2016].

[6] GUITTON, C. A review of the available content on Tor hidden services: The case against further development *Computers in Human Behavior* [online]. 14. 8. 2013, roč. 29, č. 6. [cit. 6. 5. 2016]. Dostupné z URL: <http://www.sciencedirect.com.ezproxy.lib.vutbr.cz/science/article/pii/S0747563213002690>. ISSN 0747-5632.

[7] RYBKA, M. *Jak je důležité nežít v informační bublině* [online]. 4. 12. 2015 [cit. 17. 4. 2016]. Dostupné z URL: <http://pctuning.tyden.cz/multimedia/hry-a-zabava/38375-jak-je-dulezite-nezit-v-informacni-bubline>.

[8] OWEN, Gareth, SAVAGE, Nick. *The Tor Dark Net* [online]. 2015 [cit. 17. 4. 2016]. Dostupné z URL: <https://www.cigionline.org/sites/default/files/no20_0.pdf>.

[9] ANDRE *What everybody ought to know about hidemyass* [online]. 12. 10. 2013 [cit. 17. 4. 2016]. Dostupné z URL: <https://invisibler.com/lulzsec-and-hidemyass/>.

[10] Differences between transparent, anonymous and elite proxy. *Tiq's tech-blog* [online]. 22. 6. 2012 [cit. 17. 4. 2016] Dostupné z URL: <https://tech.tiq.cc/2012/06/differences-between-transparent-anonymous-and-elite-proxy/>.

[11] GREENEMEIER, L. Seeking Address: Why Cyber Attacks Are So Difficult to Trace Back to Hackers. *Scientific American* [online]. 11. 6. 2011 [cit. 18. 4. 2016] Dostupné z URL: <http://www.scientificamerican.com/article/tracking-cyber-hackers/>.

[12] SOOD, Aditya, ENBODY, Richard. *Targeted Cyber Attacks: Multi-staged Attacks Driven by Exploits and Malwar* 21. 4. 2014, 158 s. [cit. 18. 4. 2016] ISBN 0128006048

[13] Shooting Behind the Fence. *Radware ERT Research Paper* [online]. 2013 [cit. 19. 4. 2016] Dostupné z URL: <https://security.radware.com/uploadedFiles/Resources_and_Content/Attack_Tools/Shooting_Behind_the_Fence_ERT_Research_Paper.pdf>.

[14] GAYER, Ofer, ZEIFMAN, Igal. Shotgun DDoS Attacks Originating from Anonymous Proxies *Incapsula Blog* [online]. 31. 3. 2015 [cit. 19. 4. 2016] Dostupné z URL: <https://www.incapsula.com/blog/shotgun-ddos-anonymous-proxy.html>.

[15] KLUBAL, M. *Problematika sítí typu botnet.* Brno, 2013. 80 s. Diplomová práce na Provozně ekonomické fakultě Mendelovy univerzity v Brně. Vedoucí diplomové práce Ing. Jan Přichystal, Ph.D.

[16] BINGHAM, J. *The Russian mastermind behind Backdoor.Proxybox* [online]. 2011, poslední aktualizace 8. 10. 2012 [cit. 28. 4. 2016]. Dostupné z URL: <http://www.symantec.com/connect/blogs/russian-mastermind-behind-backdoorproxybox>.

[17] STEDING-JESSEN, Klaus, VIJAYKUMAR, L. Nandamudi, MONTES, Antonio. *Using Low-Interaction Honeypots to Study the Abuse of Open Proxies to Send Spam* [online]. 2007, poslední aktualizace 4. 12. 2007 [cit. 7. 5. 2016]. Dostupné z URL: <http://www.lac.inpe.br/~vijay/download/Papers/Infocomp_2008_1.pdf>.

[18] LEYDEN, J. *Open relay spam is 'dying out'* [online]. 2003, poslední aktualizace 12. 6. 2003 [cit. 6. 5. 2016]. Dostupné z URL: <http://www.theregister.co.uk/2003/06/12/open_relay_spam_is_dying/>.

[19] Exposing the Underground: Adventures of an Open Proxy Server. *Secure Works* [online]. 21. 5. 2011 [cit. 7. 5. 2016]. Dostupné z URL: <https://www.secureworks.com/blog/proxies>.

[20] ZELSTER, L. *Internet Noise and Malicious Requests to a New Web Server* [online]. 2014, poslední aktualizace 13. 8. 2014 [cit. 6. 5. 2016]. Dostupné z URL: <https://zeltser.com/malicious-web-requests/>.

[21] HASCHEK, Ch. Analyzing 443 free proxies – Only 21% are not shady. *Christian Haschek's blog* [online]. 21. 6. 2015 [cit. 6. 5. 2016]. Dostupné z URL: <https://blog.haschek.at/2015-analyzing-443-free-proxies>.

[22] HASCHEK, Ch. Let's analyze over twenty thousand proxies. *Christian Haschek's blog* [online]. 4. 7. 2015 [cit. 6. 5. 2016]. Dostupné z URL: <https://blog.haschek.at/2015-lets-analyze-twenty-thousand-proxies>.

[23] HASCHEK, Ch. Why are free proxies free? *Christian Haschek's blog* [online]. 29. 5. 2013 [cit. 6. 5. 2016]. Dostupné z URL: <https://blog.haschek.at/post/fd9bc>.

[24] Simple PHP Proxy Detection Script and Client Real IP Address. *PHP Developer* [online]. 2012 [cit. 10. 5. 2016]. Dostupné z URL: <http://www.php-developer.org/simple-php-proxy-detection-script-and-client-real-ip-address/>.

[25] FULKERSON, E. *What is my proxy*[online]. [cit. 10. 5. 2016]. Dostupné z URL: <http://www.whatismyproxy.com/list_of_proxy_tests_performed.php>.

[26] Proxy Checker API. *Proxy IP checker* [online]. [cit. 7. 5. 2016]. Dostupné z URL: <http://proxyipchecker.com/api.html>.

[27] Proxy List by Port Number. *Xroxy* [online]. [cit. 27. 4. 2016]. Dostupné z URL: <http://www.xroxy.com/proxy-port.htm>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

API rozhraní pro programování aplikací – Application Programming Interface

CAPTCHA plně automatický veřejný Turingův test k odlišení počítačů a lidí – Completely Automated Public Turing test to tell Computers and Humans Apart

C&C řídící jednotka botnetu – Command & Control

FBI federální úřad pro vyšetřování – Federal Bureau of Investigation

HMA! webový server poskytující anonymizační služby – Hide My Ass!

FTP síťový protokol pro přenos souborů mezi počítači – File Transfer Protocol

DDoS distribuovaný útok DoS – Distributed Denial of Service

DoS útok na internetové služby s cílem odstavení služby uživatelům – Denial of Service

HTML značkovací jazyk hypertextu pro vytváření stránek v systému World Wide Web – HyperText Markup Language

HTTP internetový protokol určený pro výměnu hypertextových dokumentů ve formátu HTML – Hypertext Transfer Protocol

HTTPS zabezpečený internetový protokol určený pro výměnu hypertextových dokumentů ve formátu HTML – Hypertext Transfer Protocol Secured

ICMP služební protokol pro diagnostické a směrovací účely – Internet Control Message Protocol

IMAP internetový protokol pro vzdálený přístup k e-mailové schránce prostřednictvím e-mailového klienta – Internet Message Access Protocol

IP internetový protokol — Internet Protocol

NAT překlad síťových adres – Network Address Translation

NTP protokol pro synchronizaci vnitřních hodin počítačů v paketové síti – Network Time Protocol

OCR optické rozpoznávání znaků – Optical Character Recognition

PHP programovací jazyk pro vývoj webových stránek – Hypertext Preprocessor

POP3 internetový protokol pro stahování emailových zpráv ze vzdáleného serveru na klienta – Post Office Protocol

SMTP internetový protokol určený pro přenos zpráv elektronické pošty – Simple Mail Transfer Protocol

SOCKS internetový protokol využívající ke komunikaci proxy server – Socket Secure

SSH zabezpečený komunikační protokol v počítačových sítích – Secure Shell

TCP protokol transportní vrstvy pro spolehlivý přenos dat – Transmission Control Protocol

TTL parametr životnosti paketů protokolu IP – Time To Live

URL jednotná adresa zdroje udávající plnou cestu k umístění informací v Internetu – Uniform Resource Locator

VoIP technologie pro přenos digitalizovaného hlasu prostřednictvím počítačové sítě nebo Internetu – Vioce over Internet Protocol

VPN virtuální privátní síť určená k zabezpečenému spojení stanic – Virtual Private Network

SEZNAM PŘÍLOH

A Obsah přiloženého CD **60**

A OBSAH PŘILOŽENÉHO CD

- Elektronická verze diplomové práce.
- Sestavený program Proxy checker.
- Hlavní měřící sada IP adres.