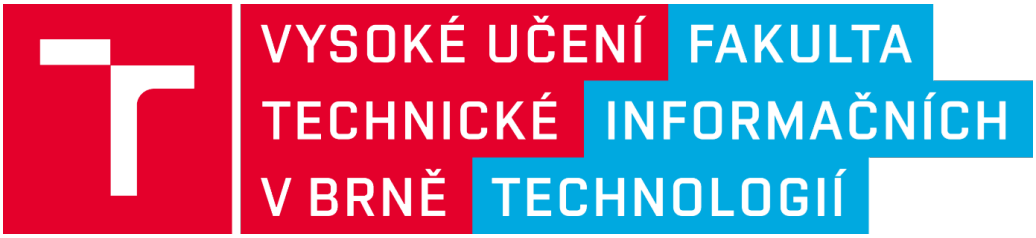


Síťové aplikace a spáva sítí

Tunelování datových přenosů přes DNS dotazy

10. listopadu 2022

Tomáš Souček, xsouce15

Obsah

1 Úvod

Cílem tohoto projektu bylo vytvořit nástroj pro tunelování dat prostřednictvím DNS dotazů. Součástí projektu byla implementace jak serverové části, tak i klientské.

2 DNS přenosy

DNS neboli Domain Name System je globální adresář doménových jmen a dalších identifikátorů síťových zařízení a služeb. V praxi je DNS rozděleno hierarchicky podle globálního prostoru doménových adres. DNS zpráva se skládá z hlavičky, otázky, odpovědi, odkazu na autoritu a dodatečných dat. [3]

DNS tunelování funguje na principu kódování dat a jejich schování do standartního DNS požadavku. Zakódovaná data se přilepí k bázové doméně a zašlou na DNS server. DNS tunelování bývá často zneužíváno pro infekci počítače oběti malwarem, jelikož DNS není primárně určeno pro přenos dat. Stává se tedy, že nebývá dostatečně monitorované a tím pádem dává útočníkovi větší pravděpodobnost úspěchu. DNS pakety také nebývají blokovány firewallem, jelikož bez DNS by nemohla efektivně fungovat žádná počítačová síť [7].

3 Návrh a implementace

Pro lepší strukturalizaci adresáře byly mnou vytvořené implementační soubory umístěny do adresářů s již předpřipraveným rozhraním. Do kořenové složky byly pak vloženy společné soubory pro celý projekt – `Makefile` a pomocná knihovna.

3.1 Klient

Klientská část, neboli sender byla implementována do souboru `dns_sender.c`. Na začátku běhu senderu dochází ke kontrole a uložení argumentů do proměnných. Načtené argumenty jsou uloženy velkými písmeny se jmény dle programové specifikace – např. bázová doména je uložena jako `BASE_HOST`. Hlavním elementem klienta je smyčka čtoucí vstup znak po znaku a ukládájící znaky do textového řetězce. Při čtení dochází také k počítání počtu načtených znaků. Dojde li ve smyčce požadovému počtu načtených znaků, pomocí funkce `sendPacket` dojde k odeslání paketu na server.

Funkce `sendPacket` zakóduje poslaný textový řetězec do base64 formátu [5], k zakódovaným datům připojí basehost doménu, změní tahle data na korektní DNS formát [8] a pošle na server. Po odeslání očekává funkce od serveru odpověď obsahující stejná zakódovaná data, jako byly poslány. Tento mechanismus je použit ke kontrole správnosti serverem obdržených dat. Dalé probíhá i kontrola, zda paket dorazil na server ještě před timeoutem [6]. Pokud ne, pak se pakety posílají znovu.

3.2 Server

Server, neboli příjemce (angl. receiver), pak byl implementován do souboru `dns_receiver.c`. Stejně jako sender si receiver načte a zkotnroluje argumenty. Následně si připraví socket a propojí si jej pomocí příkazu bind k serveru [4]. Hlavní částí programu je nekonečný cyklus, který přijme data od senderu, provede kontrolu, zda jsou data ve formátu base64 a zda zasílá sender správný `BASE_HOST`, následně data dekóduje a uloží do požadované složky a souboru [2]. Jméno souboru zjistí z prvního paketu. Po provedení těchto operací zašle zpět senderu paket obsahující zakódovaná přijatá data, aby mohlo dojít k potvrzení, že komunikace funguje. Smyčka je na konci komunikace ukončena přijmutí speciálních dat značících konec přenosu.

3.3 Pomocná knihovna

Pro přehlednost zdrojových kódů senderu a receiveru byla velká část společných funkcí přesunuta do hlavičkového souboru `library.h`. Na pomocnou knihovnu se pak z obou implementačních souborů odkazuje přes `#include`. V knihovně lze najít např. funkce spojené s kódováním a dekódováním dat nebo funkce pro práci s DNS pakety [1].

4 Testování a měření

Pro kontrolu správnosti odeslaných dat byl použit program Wireshark, který zachytával odeslané DNS pakety. Wireshark mi dost pomohl s opravením malformed paketů.
Na kontrolu stejnosti odeslaných a přijatých dat byl použit linuxový nástroj `diff`. Dále pak probíhalo testování porovnáváním výsledků a to včetně počtu poslaných a přijatých dat.

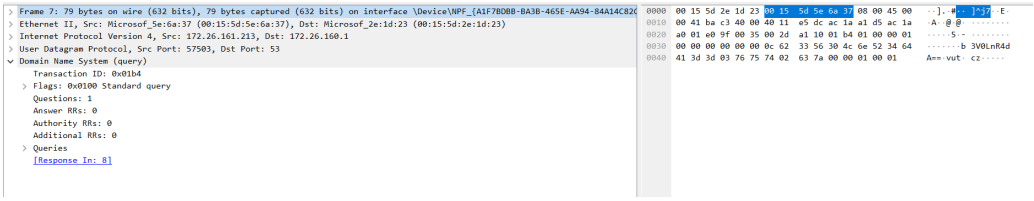
Po zhodnocení testování jsem došel k závěru, že programy odesílají i přijmají platné DNS pakety.

```
suky@LAPTOP-D4NJ7B15:/mnt/c/VUT/3BIT-Z/isa/sender$ ./dns_sender -u 127.0.0.1 vut.cz out.txt data.txt
[INIT] 127.0.0.1
[ENCD] vut.cz         0 'b3V0LnR4dA==.vut.cz'
[SENT] vut.cz         0 12B to 127.0.0.1
[ENCD] vut.cz         1 'TG9yZW0gaXBzdW0gKHprcsOhY2VuxJsgbGlwc3VtKSBqZSBvem5hxI1lbsOt.vut.cz'
[SENT] vut.cz         1 60B to 127.0.0.1
[ENCD] vut.cz         2 'IHBybyBzdGFuZGFyZG7DrSBwc2V1ZG9sYXRpbnNrw70gdGV4dCB1xb7DrXZh.vut.cz'
[SENT] vut.cz         2 60B to 127.0.0.1
[ENCD] vut.cz         3 'bsO9IHYgZ3JhZmlja8OpbSBkZXNpZ251IGEgbmF2cmhvdsOhbsOtIGpha28g.vut.cz'
[SENT] vut.cz         3 60B to 127.0.0.1
[ENCD] vut.cz         4 'ZGVtb25zdHJhdGl2bsOtIHbDvXBsxYhvdsO9IHRleHQgcMWZaSB2eXR2w6HF.vut.cz'
[SENT] vut.cz         4 60B to 127.0.0.1
[ENCD] vut.cz         5 'mWVuw60gcHJhY292bsOtY2ggdWvDoXplayBncmFmaWNrw71jaCBuw6F2cmjF.vut.cz'
[SENT] vut.cz         5 60B to 127.0.0.1
[ENCD] vut.cz         6 'ryAobmFwxZkuIGludGVybmV0b3bDvWNoIHN0csOhbmVrLCByb3p2csW+ZW7D.vut.cz'
[SENT] vut.cz         6 60B to 127.0.0.1
[ENCD] vut.cz         7 'rSDEjWFzb3Bpc8WvIMSNaSB2xaFlY2ggZHJ1aMWvIHJla2xhbW7DrWNoIG1h.vut.cz'
[SENT] vut.cz         7 60B to 127.0.0.1
[ENCD] vut.cz         8 'dGVyacOhbMWvKS4gTGlwc3VtIHRhayBwcmFjb3ZuxJsgem7DoXpvcsWIdWpl.vut.cz'
[SENT] vut.cz         8 60B to 127.0.0.1
[ENCD] vut.cz         9 'IHRleHQgdiB1a8OhemtvdsO9Y2ggbWFrZXTDoWNoICh0enYuIG1vY2stdXAp.vut.cz'
[SENT] vut.cz         9 60B to 127.0.0.1
[ENCD] vut.cz        10 'IHDFmWVkdMOtbSwgbmXFviBidWRlIGRvIGhvdG92w6lobyBuw6F2cmh1IHZs.vut.cz'
[SENT] vut.cz        10 60B to 127.0.0.1
[ENCD] vut.cz        11 'b8W+ZW4gc215c2x1cGxuw70gb2JzYWgu/w==.vut.cz'
[SENT] vut.cz        11 36B to 127.0.0.1
[ENCD] vut.cz        12 'fEVYSVR8.vut.cz'
[SENT] vut.cz        12 8B to 127.0.0.1
[CMPL] out.txt of 656B
```

Obrázek 1: Screenshot z běhu programu `sender.c`

```
suky@LAPTOP-D4NJ7B15:/mnt/c/VUT/3BIT-Z/isa/receiver$ sudo ./dns_receiver vut.cz data
[INIT] 127.0.0.1
[RECV] data/out.txt         0 12B from 127.0.0.1
[PARS] data/out.txt 'out.txt'
[RECV] data/out.txt         1 60B from 127.0.0.1
[PARS] data/out.txt 'Lorem ipsum (zkráceně lipsum) je označení'
[RECV] data/out.txt         2 60B from 127.0.0.1
[PARS] data/out.txt ' pro standardní pseudolatinský text užíva'
[RECV] data/out.txt         3 60B from 127.0.0.1
[PARS] data/out.txt 'ný v grafickém designu a navrhování jako '
[RECV] data/out.txt         4 60B from 127.0.0.1
[PARS] data/out.txt 'demonstrativní výplňový text při vytvá'
[RECV] data/out.txt         5 60B from 127.0.0.1
[PARS] data/out.txt 'ení pracovních ukázek grafických návrh'
[RECV] data/out.txt         6 60B from 127.0.0.1
[PARS] data/out.txt ' (např. internetových stránek, rozvržen'
[RECV] data/out.txt         7 60B from 127.0.0.1
[PARS] data/out.txt ' časopisů či všech druhů reklamních ma'
[RECV] data/out.txt         8 60B from 127.0.0.1
[PARS] data/out.txt 'teriálů). Lipsum tak pracovně znázorňuje'
[RECV] data/out.txt         9 60B from 127.0.0.1
[PARS] data/out.txt ' text v ukázkových maketách (tzv. mock-up)'
[RECV] data/out.txt        10 60B from 127.0.0.1
[PARS] data/out.txt ' předtím, než bude do hotového návrhu vl'
[RECV] data/out.txt        11 36B from 127.0.0.1
[PARS] data/out.txt 'ožen smysluplný obsah.h'
[RECV] data/out.txt        12 8B from 127.0.0.1
[PARS] data/out.txt '|EXIT|'
[CMPL] data/out.txt of 656B
```

Obrázek 2: Screenshot z běhu programu `receiver.c`



Obrázek 3: Zachycení a kontrola poslaných dat pomocí programu Wireshark

5 Omezení

- Projekt byl realizován pomocí protokolu UDP, takže se může stát, že při přenášení většího množství dat, dojde k přetečení UDP paketu

6 Zdroje

Reference

[1] BinaryTides: *DNS Query Code in C with Linux sockets*. [online], [viděno 17.10.2022].
URL <https://www.binarytides.com/dns-query-code-in-c-with-linux-sockets/>

[2] Blanc, A. L.: *Creating a new directory in C*. [online], [viděno 17.10.2022].
URL <https://stackoverflow.com/a/7430262>

[3] Doc. Ing. Petr Matoušek, M., Ph.D.: *Síťové aplikace a správa sítí - Systém DNS*. [online], [viděno 21.10.2022].
URL <https://moodle.vut.cz/pluginfile.php/511699/mod_resource/content/1/isa-dns.pdf>

[4] Geeksforgeeks: *UDP Server-Client implementation in C*. [online], [viděno 12.10.2022].
URL <https://www.geeksforgeeks.org/udp-server-client-implementation-c/>

[5] Nachtimwald, J.: *Base64 Encode and Decode in C*. [online], [viděno 12.10.2022].
URL <https://nachtimwald.com/2017/11/18/base64-encode-and-decode-in-c/>

[6] Neal, f.: *UDP Socket Set Timeout*. [online], [viděno 20.10.2022].
URL <https://stackoverflow.com/a/13547864>

[7] Networks, P. A.: *SDNS Tunneling – co je to?* [online], [viděno 21.10.2022].
URL <https://nextgenfw.cz/2020/03/10/dns-tunneling-co-je-to/>

[8] user405725: *Removing spaces and special characters from string*. [online], [viděno 19.10.2022].
URL <https://stackoverflow.com/a/15444792>