
# PROJEKTOVÁ DOKUMENTÁCIA

## Tunelovanie dátových prenosov cez DNS dotazy

14.11.2022 Tomáš Homola (xhomol28)

# Obsah

# 1 Úvod

Cieľom projektu je implementácia nástroja pre tunelovanie dát prostredníctvom DNS dotazov. Súčasťou implementácie je klientská i serverová časť aplikácie. Ako komunikačný protokol medzi klientskou a serverovou časťou je zvolený UDP protokol.

# 2 Implementácia

## 2.1 Klient

## 2.2 Možnosti spustenia

```
dns_sender [-u UPSTREAM_DNS_IP] {BASE_HOST} {DST_FILEPATH} [SRC_FILEPATH]

$ dns_sender -u 127.0.0.1 example.com data.txt ./data.txt
$ echo "abc"|dns_sender -u 127.0.0.1 example.com data.txt
```

Prepínače:

- `-u` vynútenie vzdiaľeného DNS serveru
  - Pokiaľ nie je zadaný, využije sa vychodzí DNS server nastavený v systéme

Pozičné parametry:

- `BASE_HOST` slouží k nastavení bázové domény všech přenosů
- `DST_FILEPATH` cesta pod kterou se data uloží na serveru
- `[SRC_FILEPATH]` cesta k souboru který bude odesílán
  - pokud není specifikováno pak program čte data ze STDIN

## 2.3 Implementácia

Prv parsujeme argumenty pomocou funkcie `parsingArguments()`. Vstupné dáta ukladáme do štruktúry `Sender_args`.

Nasleduje vytvorenie socketu a vytvorenie DNS štruktúry pozostávajúcej z DNS hlavičky a sekcie Resource Records.

QNAME časť štruktúry za pomoci funkcie `ChangeToDnsFormat()` upravíme do správneho tvaru, v ktorom sa bodky nahradia za čísla vyjadrujúce počet nasledujúcich znakov po ďalší výskyt bodky. Následne QNAME zakódujeme za pomoci `base64`.

DNS packety na server posielame postupne. Cez prvý DNS packet sa pošle cesta, pod ktorou sa uložia dáta na serveri. Nasleduje cysklus `while`, ktorý číta dáta zo súboru obsahujúceho dáta na odosielanie. Z dôvodu zjednodušenia implementácie čítame po 45 znakoch. Dáta sa zakódujú, pripoja sa k nej potrebné informácie, aby to spĺňalo správnu štruktúru DNS formátu a odošlú na server pomocou `sendto()`. Po odoslaní všetkých dát zo súboru v DNS paketoch sa cyklus ukončí.

Ako posledný paket posielame informáciu o tom, že sa všetky pakety odoslali na server a zatvoríme socket.

## 2.4 Obmedzenia a nedostatky

- nepodporuje IPv6
- málo kontrol chybných prípadov
- `chunkSize` u testovacích funkcií je určený pravdepodobne zle

## 2.5 Server

## 2.6 Možnosti spustenia

```
dns_receiver {BASE_HOST} {DST_DIRPATH}
```

```
$ dns_receiver example.com ./data
```

Pozičné parametry:

- `BASE_HOST` slúži k nastaveniu bázovej domény k prijatiu dát
- `DST_DIRPATH` cesta, pod ktorou sa budú všetky prichádzajúce dáta/súbory ukladať

## 2.7 Implementácia

Ako prvé parsujeme argumenty pomocou funkcie `parsArguments()` a ukladáme ich do štruktúry `Receiver_args`. Následne vytvoríme socket a postupne príjmame dáta od klienta cez funkciu `recvfrom()`.

Z prijatých dát vytiahneme iba zakódovanú časť - QNAME a dekódujeme ju s `base64_decode()`.

Prvý paket obsahuje cestu k súboru, do ktorého uložíme prijaté dekódované dáta. Následne pakety obsahujú dáta, ktoré zapíšeme do daného súboru. Posledný paket obsahuje správu, ktorá identifikuje, že bol prijatý posledný paket.

Pri prijatí každého paketu odosielame za pomoci `sendto()` klientovi spätnú väzbu o prijatí paketov vo forme správy. Po prijatí všetkých paketov server čaká na prijatie nových.

## 2.8 Obmedzenia

- nepodporuje IPv6
- málo kontrol chybných prípadov
- `chunkSize` u testovacích funkcií je určený pravdepodobne zle

# 3 Testovanie

Program sa testoval za pomoci aplikácie Wireshark. Vďaka výpisom sme si overili správnosť DNS formátu.

```
15 1.970391433   192.168.137.7   8.8.8.8           DNS      84 Standard query 0x0001 A dGV4dC50eHQ=.example.com
16 1.998261149   8.8.8.8         192.168.137.7     DNS     140 Standard query response 0x0001 No such name A dGV4dC50eHQ=.example.com SOA ns.icann.org
17 1.998610829   192.168.137.7   8.8.8.8           DNS     108 Standard query 0x0002 A TG9yZW0gaXBzdW0gZG9sb3Igc2l0IGFtZXQ=.example.com
18 2.013108237   8.8.8.8         192.168.137.7     DNS     164 Standard query response 0x0002 No such name A TG9yZW0gaXBzdW0gZG9sb3Igc2l0IGFtZXQ=.example.com SOA ns.icann.org
19 2.013397969   192.168.137.7   8.8.8.8           DNS     100 Standard query 0xffff A KiBBTExfUEFDS0VUU19TRU5EICo=.example.com
20 2.027169247   8.8.8.8         192.168.137.7     DNS     156 Standard query response 0xffff No such name A KiBBTExfUEFDS0VUU19TRU5EICo=.example.com SOA ns.icann.org
```

```
▸ User Datagram Protocol, Src Port: 54419, Dst Port: 53
▾ Domain Name System (query)
    Transaction ID: 0x0001
  ▾ Flags: 0x0100 Standard query
      0... .... .... .... = Response: Message is a query
      .000 0... .... .... = Opcode: Standard query (0)
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... .0.. .... = Z: reserved (0)
      .... .... ...0 .... = Non-authenticated data: Unacceptable
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ▾ Queries
    ▸ dGV4dC50eHQ=.example.com: type A, class IN
    [Response In: 16]
```

```
0000  f6 aa ea be fd dd 4c 1d  96 a1 b8 00 08 00 45 00   ·····L· ·····E·
0010  00 46 2d ac 40 00 40 11  b3 3b c0 a8 89 07 08 08   ·F-·@·@· ·;·····
0020  08 08 d4 93 00 35 00 32  5a 03 00 01 01 00 00 01   ·····5·2 Z·····
0030  00 00 00 00 00 00 0c 64  47 56 34 64 43 35 30 65   ·······d GV4dC50e
0040  48 51 3d 07 65 78 61 6d  70 6c 65 03 63 6f 6d 00   HQ=·exam ple·com·
0050  00 01 00 01                                       ····
```