VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta informačných technológií

Síťové aplikace a správa sítí

2022/2023

**Projekt ISA**

**Tunelování datových přenosů přes DNS dotazy**

Králik Dalibor (xkrali20) Brno 8.12.2021

Obsah

[1. Problematika 3](#_Toc118534980)

[2. Návrh aplikácie 3](#_Toc118534981)

[2.1. DNS sender 3](#_Toc118534982)

[2.1.1. DNS header 4](#_Toc118534983)

[2.1.2. DNS question 4](#_Toc118534984)

[2.2. DNS server 5](#_Toc118534985)

[2.3. Komunikačný protokol 5](#_Toc118534986)

[3. Implementácia 5](#_Toc118534987)

[3.1. DNS Sender 6](#_Toc118534988)

[3.2. DNS Server 6](#_Toc118534989)

# Problematika

DNS (Domain name system) je decentralizovaný system doménových mien, ktorý je realizovaný DNS servermi a prokotolom DNS. Jeho hlavnou úlohou je preklad doménových mien na IP adresy a naopak.

DNS tunneling je metoda hackerského útoku, kedy útočník môže pomocou infikovaného počítača v internej sieti vynášať data a prijímať ich pomocou svojho servera. Hacker potrebuje ovládať doménu a server, ktorý bude pôsobiť ako autoritatívny DNS server, aby mohol spustiť programy tunelovania. Vzhľadom na to, že DNS dotazy sú vždy povolené prejsť skrz firewall, infikovaný počítač je schopný posielať na DNS dotazy na DNS resolver, ktorý tieto dotazy pošle na útočníkov server a prijíme ich. Jedná sa teda o tunelovanie dat – získavania a extraktovanie informácií.

# Návrh aplikácie

## DNS sender

Hlavnou úlohou DNS senderu je poslať UDP packet, ktorý obsahuje DNS packet s dátami pomocou DNS servera, ktorý bol zadaní ako parameter programu dns\_sender, alebo dns servera zo súboru resolv.conf. Následne sa tento UDP packet pošle na DNS port - port číslo 53.

Program musel vytvoriť korektne UDP packet viz [1] s DNS packetom a poslať ho.

DNS packet sa skladá z viacerých častí:

1. DNS header
2. DNS question
3. DNS answer
4. Authority
5. Additional

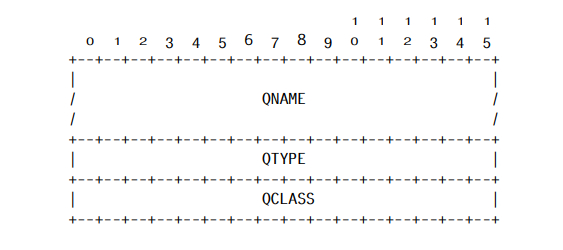
### DNS header

Obrázok 1: DNS header

DNS header popisuje typ packetu a to, čo daný DNS packet obsahuje viz [2].

Pre DNS header som v projekte vytvoril datovú štruktúru, ktorú som potom postupne napĺňal dátami.

### DNS question



Obrázok 2. DNS question

DNS question obsahuje 3 polia pre dáta (QName, QType, QClass).

Pole QType špecifikoval typ DNS požiadavku. V tomto projekte som využil hodnotu htons(1), ktorá reprezentovala A záznam.

Pole QClass špecifikovalo triedu DNS požiadavku. V tomto projekte som použil hodnotu htons(1), ktorá reprezentovala *Internetové adresy*.

QName je pole, ktoré ma maximálnu dĺžku 255 bytov a obsahuje data s bázovou doménou, na ktorú sa má DNS packet posielať. V QName som posielal zašifrované dáta spoločne s bázovou doménou. Avšak pre správne formátovanie dat v QName bolo potrebné previesť tieto data to tzv. DNS formátu. Tento format vyzerá tak, že data musia byť v substringoch dlhých maximálne 63 znakov a pred týmto substringom sa musí nachádzať číslo počtu znakov v substringu prevedené do hexadecimálneho formátu. Jedine v takomto prípade maju dáta v QName správny format a software ako Wireshark dokáže QName rozložiť a vyčátať z neho dáta.

Data v QName som šifroval pomocou šifry BASE32, ktorej knižnicu pre C napísal Markus Gutschke, ktorá podieha the Apache License, Version 2.0. a bola voľne dostupná na stiahnutie.

Vzhľadom na to, že DNS sender posielal len tzv. DNS Question, nebolo nutné sa v tomto prípade starať o DNS Answer.

Po odoslaní UDP packetu DNS sender čakal, kým sa mu vráti UDP packet od DNS receivru s DNS answer. DNS sender mal nastavený tzv. timeout, po ktorého uplynutí sa predpokladalo, že UDP packet nedorazil k DNS receivru a DNS sender zopakoval poslanie packetu s datami. Tento timeout mal DNS sender nastavený na 3 sekundy. Timeout som nastavil pomocou funkcie setsockopt().

## DNS receiver

Cieľom DNS receivru je prijať UDP packety na porte 53 z ľubovolnej zdrojovej adresy, vytiahnúť data z DNS QNAME, dešifrovať ich pomocou knižnice base32.c, data uložiť do súboru a poslať validnú odpoveď -> UDP packet s DNS packetom a DNS Answer.

DNS receiver beží v nekonečnom while cykle, ktorý prijíma UDP packety pomocou funkcie recvfrom(). Po prijatí UDP packetu tento packet rozloží a vytiahne z neho potrebné dáta. DNS receiver skontroluje, či prijal packet so správnou bázovou doménou. Ak táto bázová doména bola správna, začal spracovávať data, ktoré dešifruje a uloží do súboru. Následne vytvorí nový UDP packet, ktorý obsahuje DNS odpoveď. Náplní buffer štruktúrou DNS header, DNS question a následne DNS answer.

Po naplnení bufferu tento UDP packet pošle na adresu, z ktorej prijal UDP packet s DNS question.

Vzhľadom na to, že beží v nekonečnej smýčke, tak celý tento proces opakuje, až kým nie je jeho program ukončený pomocou CTRL+C.

## Komunikačný protokol

DNS sender neposielal len datové packety. Pre poslaním prvého datového packetu poslal DNS sender tzv. inicializačný packet, ktorý služil na inicializáciu spojenia a poslanie inicializačných dát(FILE\_PATH) DNS receivru. DNS receiver si z QNAME inicializačného packetu zobral FILE\_PATH (cestu k súboru, do ktorého sa majú ukladať dešifrované data). Datový obsah init packet bol nasledovky:

INITPATH[*FILE\_PATH*]

Kde za FILE\_PATH program doplnil cestu k súboru, ktorá bola zadaná ako parameter programu.

Následne sa po init packete posielali datové packety so zašifrovanými dátami v QNAME.

Po odoslaní všetkých packetov sa vytvoril a odoslal END packet, ktorý signalizoval, že DNS sender ukončil posielanie dat. Tento packet je veľmi dôležitý hlavne pre DNS receiver, pretože po prijatí tohto packetu korektne uzatvorí súbor, do ktorého sa zapisovali data. Data v END packete majú nasledujúci tvar:

[ENDPACKET]

# Implementácia

## DNS Sender

1. Prijatie a spracovanie parametrov program
2. Vytvorenie socketu (endpointu) a nastavenie socketu pomocu funkcie setsockopt()
3. Tvorba inicializačného packetu, tvorba a naplnenie DNS header a DNS question
4. Odoslanie inicializačného packetu
5. Čakanie a spracovanie answer packetu s DNS Answer, v prípade chýbajúcej odpovede sa program presunie na bod 3.
6. Priprava DNS Header pre datový packet
7. Zašifrovanie časti načítaných dát a vloženie do QNAME v správnom formácie + pridanie BASE\_HOST
8. Poslanie datového packet
9. Čakanie a spracovanie answer packet s DNS Answer, v prípade chýbajúcej odpovede sa program presunie na bod 6.
10. Ak DNS sender prijíme answer packet, zoberie dalšie data a presunie sa na bod 6., kým nespracuje všetky data.
11. Po odoslaní všetkých datových packetov program začne pripravovať DNS header pre END packet.
12. Pridanie dat do QNAME.
13. Odoslanie END packet
14. Čakanie a spracovanie answer packetu s DNS Answer, v prípade chýbajúcej odpovede sa program presunie na bod 10.

## DNS Receiver

1. Prijatie a spracovanie parametrov program
2. Vytvorenie socketu (endpointu) a nastavenie socketu pomocu funkcie setsockopt()
3. Prijímanie packetov v infinite while cykle
4. V prípade prijatia INIT packetu sa zistí z dát FILE\_PATH, spojí sa s DIR\_PATH, vytvorí sa a otvorí daný súbor
5. Odošle sa UDP packet DNS senderovi s DNS answer
6. V prípade datového packetu sa data spracujú, dešifrujú a uložia do súboru
7. Odošle sa UDP packet DNS senderovi s DNS answer
8. V prípade END packet sa uzatvorí korektne otvorený súbor pomocou funkcie fclose()
9. Odošle sa UDP packet DNS senderovi s DNS answer

# Návod na použitie

Po rozbalení balíčka tar.gz sa presuňte do hlavného priečinka projektu. Následne otvorte terminál, ktorý sa bude nachádzať v hlavnom priečinku projektu. Príkazom *$make*  preložíte projekt. Tento príkaz by Vám mal vygenerovať spustitelné súbory *dns\_sender* a *dns\_receiver.* Bližšie informácie o parametroch a spôsoboch spustenia programov *dns\_sender*  a *dns\_receiver* sa dočítate v súbore README.md, ktorý sa nachádza v hlavnom priečinku projektu.

Názorný príklad spustenia:

DNS receiver sme spustili príkazom:

*$sudo ./dns\_receiver example.com ./data*

DNS sender sme spustili s príkazom:

*$./dns\_sender -u 127.0.0.1 example.com output/data.txt ./data.txt*

1. <https://www.geeksforgeeks.org/udp-server-client-implementation-c/>
2. <https://mislove.org/teaching/cs4700/spring11/handouts/project1-primer.pdf>