Vysoké učení technické
Fakulta informačních technologií

Síťové aplikace a správa sítí

Tunelování datových přenosů přes DNS dotazy

Vojtěch Vašíček

xvasic31

Mechanismus tunelování DNS dotazů

Mechanismus spočívá v odesílání DNS paketů, obsahujících zakódovanou zprávu. Zároveň tyto pakety mají validní strukturu DNS paketu, takže jsou automatickým systémem nerozeznatelné od standardních DNS paketů. Námi vytvořený DNS server poté paket zachytí, dekóduje a uloží klientem poskytnutá data do zadaného souboru.

Návrh a implementace

Klient

Klient se vyskytuje v souboru dns_sender.c a skládá se z několika funkcí. Dále se také vyskytuje několik funkcí v souboru packet.c, které zde také popíšu.

Funkce sendData

Funkce sendData zastává hlavní operační funkci programu. Je volána přímo z mainu. Funkce inicializuje socket, zavolá funkci na získání dat k odeslání, odešle informace o těchto datech a cestu k servorému souboru. Následně tato funkce cyklí a odesílá v jednotlivých cyklech data a přijímá odpovědi, kterými verifikuje že byla data přijata serverem a mají validní formát. Cyklus se zastaví jakmile se odešlou všechna data. Nakonec klient odešle ukončující paket, aby ukázal serveru že byla odeslána všechna data a vypne se.

Pomocné funkce dns_sender.c

Funkce loadData načítá data ze vstupní souboru, případně stdinu, a vrací je zpátky do funkce sendData. Dále se využívá funkce parse_args, která získá z příkazové řádky jednotlivé argumenty a vloží je do struktury args, která obsahuje všechny tyto argumenty a zajišťuje lepší manipulaci s těmito argumenty.

Funkce getQueryPacket

Tato funkce se vyskytuje v souboru packet.c a pomocí dat získaných dříve skládá dns packet, který se bude odesílat na server.

Server

Podobně jako klient, server se vyskytuje v souboru dns_receiver.c a skládá se z několika funkcí. Dále také využívá několik funkcí ze souboru packet.c.

Funkce runServer

Tato funkce je ekvivalentem funkce sendData. Funkce inicializuje soket a následně cyklí a čte příchozí pakety. Pakety rozlišuje podle toho jestli začínají přenos dat, přenášejí data nebo tento přenos ukončují. Po ukončení přenosu funkce zavolá funkci print_to_file která přijatá data dekóduje a zapíše do daného souboru.

Funkce getResponsePacket a extract_dns_query

Funkce getResponsePacket skládá paket s odpovědí, který se odešle klientovy. Tento paket obsahuje návratové kódy, značící zda byl přijatý paket validní. Kód 0 znamená že paket byl validní. Kód 1 znamená že klient odeslal paket se špatným host jménem a kód 2 znamená že měl paket špatnou strukturu. Funkce extractDnsQuery parsuje přijatý paket do struktury query. Tato funkce se také využívá v klientovy.

Komunikační protokol mezi klientem a serverem

Klient se serverem spolu komunikují pomocí protokolu UDP. Každý paket obsahující data obsahuje pouze jeden segment dat. Tedy každý paket přenese 60 bytů zakódovaných dat. Větší soubory tedy zaberou více paketů, ale nejsou tak nápadné. Pakety s odpovědí mají datovou část prázdnou, protože nejsou žádná relevantní data která bych chtěl poslat zpět klientovy. Potřebné informace se posílají v hlavičce.

Způsob kódování dat

Data se kódují do modifikované base32. Moje base32 místo číslic používá malá písmena, je tedy case sensitive. Použil jsem soubor s nemodifikovanou base32 který jsem následně upravil. Zdroj tohoto souboru je uveden v samotném souboru base32.h a bude uveden na konci této dokumentace.

Způsob ukládání dat na serveru

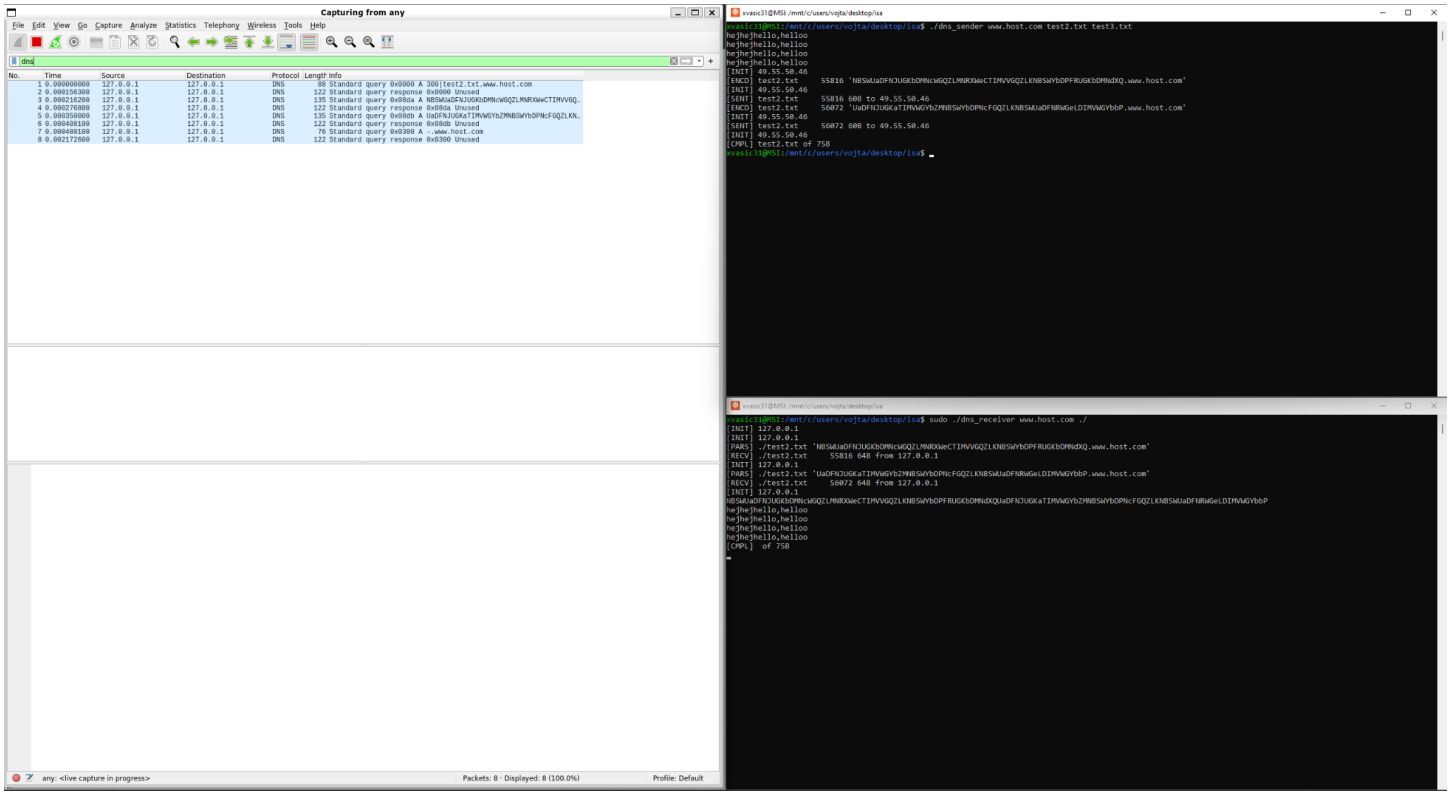
Server přijme v prvním paketu komunikace jméno souboru do kterého by se měla data ukládat. To přidá k cestě dané parametrem samotného serveru. Tímto vznikne cesta pomocí které volá funkci fwrite a dekódovaná data zapíše do souboru. Pokud již soubor existuje, server jej nepřepíše ale přidá data na konec souboru. Takto se zajistí aby uživatel nepřišel o data jež získal v minulém odeslání.

Omezení

Aplikace zvládne přeposílat pouze textová datai. Zároveň jsem nestihl naimplementovat zakódování prvního a posledního paketu komunikace. Tyto pakety jsou dle wiresharku stále validní DNS pakety, jsou ale poněkud nápadné a to i přesto že obsahují hostname.

Testování

Aplikace je testována zejména na platformě wsl2, která je v podstatě osekanou verzí ubuntu. Aplikace je také testována na vámi přiloženém virtuálním stroji. Testoval za pomocí menších i větších testových souborů pro různé adresáře. Největší testovaný soubor měl zhruba 150kb. Hlavní testovací pomůckou byly výpisy aplikace a wireshark.



Vpravo dole můžeme vidět výpis serveru a vpravo nahoře výpis klienta.

Zdroje

- google-authenticator-libpam/base32.c at master · google/google-authenticator-libpam (github.com)
- https://www.ietf.org/rfc/rfc1035.txt
- project1-primer.pdf (mislove.org)
- UDP Server-Client implementation in C - GeeksforGeeks
- DNS Query Code in C with Linux sockets - BinaryTides
- dns.h (apple.com)