VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Síťové aplikace a správa sítí

Přenos souboru skrz skrztý kanál

12. listopadu 2021

Marek Hlávka (xhlavk09)

Obsah

1 Zadání

Zadáním projektu bylo vytvořit klient/server aplikaci, pomocí které bude možno přenést soubor skrz skrytý kanál, kde jsou data přenášena pomocí ICMP protokolu, resp. uvnitř ICMP Echo-Request/Response zpráv. Kvůli zabezpečení musí být soubor zašifrován pomocí šifry AES. Serverová strana aplikace pak bude ukládat soubor do složky, ve které je spuštěn.

2 Spuštění

./secret -s <ip|hostname> -r <file> [-l]

- -s <ip|hostname> : adresa nebo hostname serveru na který bude odeslaný soubor
- -r <file> : specifikace souboru pro přenos
- -l : program spuštěný s tímto parametrem se chová jako server, který přijímá soubory posílané od klientů

3 Návrh

Jako jazyk k implemntaci byl zvolen jazyk C. Celý program je rozdělen do dvou větších celků: 1. Strana klienta a 2. Strana serveru

4 Argunemty

V této části je kontrolováno spuštění programu, resp. kontrola argumentů při spuštění. K tomuto účelu je využita funkce *getopt()* z knihovny *getopt.h*. Jsou postupně zkontrolovány parametry, které jsou blíže popsané v sekci *Spuštění*. Pokud nejsou některé požadované parametry uvedeny, program vypíše nápovědu a úspěšně se ukončí. V případě zadání parametru *-l* nesjou vyžadovány další argumenty, vzhledem k tomu že server nepotřebuje cílovou ip adresu ani specifikovat souor pro přenos.

5 Klient

5.1 Načtení vstupních dat

Po spuštění aplikace s paremetry klienta je zkontrolována daná ip addresa/hostname pomocí funkce *getaddrinfo()*, která vratí informace o vstupní adrese potřebné pro další běh programu. Následně je načten samotný soubor, který je určen k přenosu. Soubor je načtený v binární formě, tudiž nezáleží na jeho příponě. Soubor je načítán najednou tudíž je jeho velikost je omezena velikosti dostupné RAM na kterém je aplikace spuštěna.

5.2 Šifrování odesílaného paketu

Data vstupního souboru jsou šifrována naráz, jednak kvůli jednodušší implementaci, kvůli využití místa v paketu a také kvůli zvýšené bezpečnosti, aby nebyly samostatné pakety rozšifrovatelné.

5.3 Skládání a odesílání paketu

Potřebné data k složení výsledného paketu jsou uložena do struktury *icmp_packet*, definované v souboru *icmp_packet.h*. Tato struktura slouží jako mezikrok mezi získáváním všech potřebných dat a jejich zapisováním do paketu. Následně je před odesláním tato struktura přečtena a z jejich dat jsou utvořené potřebné hlavičky a payload packetu, tzn. *IPIPv6*, *ICMPICMPv6* a *ICMP_FILE* hlavičky. Hlavička *ICMP_FILE* je definovaná touto aplikací specificky pro přenos souboru pomocí icmp paketů. Výsledný paket je pak odeslán pomocí funkce *sentto()* a celý tenkto krok je zopakován pro každou část dat která vznikla rozdělením zašifrovaných dat na velikost, kterou je možné posílat po síti pomocí paketů.

6 Server

6.1 Spuštění serveru, pthread

Po spuštění serveru jsou okamžitě vztvořeny dvě vlákna, aby mohly být přijímány současně IPv4 a IPv6 pakety. Jednak je potřeba pro každý typ paketů otevřýt soket s jinými parametry, ale také aby se program nezdržoval např. při prenosu pomocí IPv4 paketů s náhodnými ICMPv6 pakety. Po vytvoření těchto vláken program čeká na příchozí pakety, keteŕe byly zachyceny.

6.2 Zpracování příchoyźích paketů

Po zachycení paktu funkcí *recvfrom* jsou data čtené z jednotlivých hlaviček. Při nesouhlasu důležitých informací jako např.: "icmp code", "filename"nebo "icmp type"je paket zahozen a čeká se příchod dalšího. Není tedy možné přenášet více souborů najednou, pokud není jeden přenášen pomocí IPv4 a druhý pomocí IPv6.

6.3 Zpracování přijatých dat

Po přijmutí všech paketů (aplikce počítá že nedochází ke ztrátám paketů), jsou jednotlivá data zkopírována dohromady, podle pořadí. Tento celek je poté rozšifrován pomocí klíče, který je zadaná přímó v programu definovaném v souboru *icmp_packet.h*.

7 Hlavička ICMP_FILE

```
struct s_icmp_file_info
{
    uint8_t type;
    uint32_t order;
    int cipher_len;
    int count;
    int part_size;
    int src_len;
    unsigned char iv[IV_SIZE];
    char filename[MAX_FILENAME];
};
```

- type ..... Typ icmp_file paketu, zejména pro případný bezpečný přenos
- order ..... Pořadí daného paketu v celkovém přenosu
- cipher_len . Délka celého zašifrovaného souboru
- count ..... Počet přenášených paketů
- part_size .. Velikost právě přenášených dat
- src_len .... Velikost původného souboru pro kontrolu po dešifrování
- iv ..... Startovací vektor pro šifru
- filename ... Název přenášeného souboru

8 Šifrování

Šifrování probíha pomocí knihovny *OpenSSL*, kontrétně pomocí šifry AES. Šifrování a rozšifrování souboru pak probíha v daných funkcích v souboru *aes.c*. Tato šifra je dostupná např. z [1]. Data jsou přesněji šifrována pomocí 256 bitové AES šifry v CBC módu. To vyžaduje aby klíč (key) byl dlouhý 256 bitů a startovaní vektor (iv) byl dlouhý 128 bitů.

Reference

[1] https://wiki.openssl.org/index.php/EVP_Symmetric_Encryption_and_Decryption