



Síťové aplikace a správa sítí

Projekt - Dokumentace

autor: Findra Michal (xfindr00)

Brno 2021

Obsah

1	Zadanie	2
2	Teoretická príprava	2
2.1	Zachytávanie paketov	2
2.2	Protokoly	2
2.2.1	ICMP	2
3	Implementácia riešenia	3
3.1	Použité knižnice	3
3.2	Inicializácia programu	4
3.3	MakeFile	4
3.4	Šifrovanie	4
3.5	Sender mode	4
3.6	Listener mode	5
3.7	Rozšírenia	5
3.8	Návod	6
3.9	Testovanie	6
4	Záver a poďakovanie	8
	Literatúra	9

1 Zadanie

Vytvořte klient/server aplikaci, která umožní přenést soubor skrz skrytý kanál, kde data jsou přenášena uvnitř ICMP Echo-Request/Response zpráv. Soubor musí být před přenosem zašifrován, aby nebyl přenášen v textové podobě.

Spuštění aplikace:

```
secret -r <file>-s <ip—hostname>[-l]
```

- -r <file>: specifikace souboru pro přenos
- -s <ip—hostname>: ip adresa/hostname na kterou se má soubor zaslat
- -l : pokud je program spuštěn s tímto parametrem, jedná se o server, který naslouchá příchozím ICMP zprávám a ukládá soubor do stejného adresáře, kde byl spuštěn.

2 Teoretická príprava

2.1 Zachytávanie paketov

Proces komunikácie a prenosu dát na internete prebieha pomocou správ, ktoré prenášajú nejakú časť informácie od odosielateľa ku klientovi. Dáta sú rozdelené na mnoho malých častí - paketov, ktoré odosielateľ podľa požiadavky odošle a klient ich prijme a získa z nich výslednú informáciu.

Na zachytávanie paketov v projekte je použitá doporučená C++ knižnica `libpcap`^[2].

2.2 Protokoly

Pakety sú vo väčšine prípadov zasielané klientovi pomocou konkrétného protokolu. Medzi najbežnejšie typy protokolov patrí UDP a TCP protokol.

2.2.1 ICMP

Internet Control Message Protocol^[1] je protokol sieťovej vrstvy založený na rodine protokolov TCP/IP. Protokol ICMP je využívaný v sieti napríklad na odosielanie chybových správ o nedosažiteľnosti routa alebo o nedostupnej požadovanej službe.

Medzi najpoužívanéjšie ICMP datagramy patrí:

- Echo request - požiadavka na odpoveď od cieľového klienta,
- Echo replay - odpoveď na *Echo request* od klienta,

- Destination Unreachable - informácie o nedostupnosti siete
- Time exceeded - vypršanie časového limitu,

a ďalšie.

Na obrázku 1¹ vidno zloženie ICMP paketu.

Checksum sa počíta funkciou `icmp_packet_checksum` a ostatné polia sú v implementácii nastavené na 0.

Type (0)	Code (0)	Checksum
Identifier		Sequence number
Optional data (ICMP payload)		

Obr. 1: Štruktúra ICMP paketu

3 Implementácia riešenia

3.1 Použité knižnice

Okrem štandardných knižníc boli použité knižnice *netinet* a to:

- ether.h,
- ip.h,
- ip_icmp.h,
- icmp6.h,
- ip6.h.

Ďalšie sieťové knižnice:

- netdb.h,
- arpa/inet.h.

Knižnice na zachytávanie paketov:

- pcap/sll.h,
- pcap.h.

¹<https://www.blogarama.com/technology-blogs/1287679-ictshorecom-blog/26524756-all-you-need-know-about-ping-icmp>

3.2 Inicializácia programu

V inicializačnej fáze programu sa spracujú argumenty pomocou funkcie `getopt`[4]. Argumenty sa skontrolujú podľa zvoleného režimu a začne sa vykonávať program podľa užívateľom zadaného vstupu buď v režime *Sender*(server) alebo v režime *Listener*(klient).

Ak v behu programu nastane chyba pri niektorej z funkcií, program sa ukončí a na chybový výstup sa vypíše informáciu o chybe.

3.3 MakeFile

Pomocou príkazu `make` sa zostaví a preloží program s potrebnými prepínačmi použitím `g++`.

MakeFile podpruje aj ďalšie dva argumenty:

- `clean` : odstráni binárne súbory,
- `pack` : použitím `tar` zabalí súbory potrebné na odovzdanie.

3.4 Šifrovanie

Zasielané dáta sú šifrované a dešifrované 128-bitovou AES šifrou. Ako šifrovací a dešifrovací kľúč je použitý študentský login (`xfindr00`). Dáta sú šifrované po 16B blokoch. Na šifrovanie sa použili funkcie z knižnice `openssl`[3] `AES_encrypt` a `AES_decrypt`.

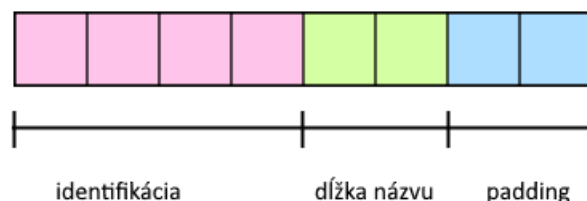
3.5 Sender mode

Skontroluje sa existencia súboru a odstráni sa prípadná cesta ku súboru, keďže má byť prijatý súbor podľa zadania uložený v aktuálnom adresári a zároveň aj preto, lebo celá dlhá cesta ku súboru by mohla spôsobiť pretečenie. Načítajú sa dáta do `stringstream` štruktúry, odkiaľ sú následne pri vytváraní paketu načítané, podľa požadovanej dĺžky dát.

Skontroluje sa v argumente programu zadaná ip adresa, resp. preloží sa *hostname* na príslušnú ip adresu. Určí sa či sa jedná o *IPv4* alebo *IPv6* adresu a podľa toho sa bude zostavovať príslušná hlavička ICMP paketu.

Zistí sa voľné miesto na dáta v pakete a z dát načítaných zo súboru sa zoberie počet Bytov deliteľný 16 a maximálne `PACKET_DATA_SIZE`. Ak počet bitov nie je dostatočný, doplnia sa ku dátam 0 a uloží sa ich počet do premennej `padding`. Tento `padding`, bude zaslaný v pakete a následne bude po prijatí odstránený.

Na začiatku dát sa prenáša 32b informácií o dátach ako je identifikácia paketu, dĺžka názvu súboru a *padding*. Ukážka uloženia dát je na obrázku 2. Podľa identifikácie sa zistí či sa má spracovať zachytený paket.



Obr. 2: Štruktúra informácií o dátach

3.6 Listener mode

Program zachytáva použitím knižnice `libpcap`^[2] na *any* rozhraní zasielané pakety. Pri zachytávaní sa používa filter, ktorý zachytáva len ICMPv4 a ICMPv6 echo pakety:

```
icmp[icmptype] = icmp-echo or icmp6[icmp6type] = icmp6-echo.
```

Zachytené ICMP pakety sú spracované podľa toho či sú ICMPv6 alebo ICMPv4. Postupne sú odstránené príslušné hlavičky. Z 32 bitov na začiatku prijatých dát je vyčítaná dĺžka názvu súboru, *padding* a identifikácia. Pred ďalším spracovaním sa najprv skontroluje identifikácia paketu. Ak je identifikácia nesprávna paket sa zahodí a nepokračuje sa v jeho spracovaní.

Ak je kontrola identifikácie úspešná, tak sa dáta dešifrujú a odstráni sa zo začiatku dešifrovaných dát názov súboru a dáta sa zapíšu na koniec súboru. Podľa zadania sa pakety nestrácajú, takže sa predpokladá, že prídu všetky.

3.7 Rozšírenia

Okrem zadaním špecifikovaných prepínačov, je podporovaný aj prepínač

- `-h` : vypísanie nápovede na štandardný výstup.

Program podporuje zasielanie a zachytávanie ICMPv6 paketov na IPv6 adresy.

Program podporuje zasielanie viacerých súborov za sebou. Program sa ukončí zachytením užívateľom zaslaným signálom SIGINT (CTRL+C) po odoslaní a prijatí všetkých požadovaných súborov. Program jednotlivé súbory spracuje a uloží do adresára z ktorého je spúšťaný skript.

3.8 Návod

Predpokladajme modelovú situáciu kde je zasielaný súbor textfile.txt na adresu 147.229.228.188.

- Stiahnuť a rozbaľiť .tar súbor,
- zostaviť program použitím príkazu `make`,
- na klientovi, ktorý prijíma súbory spustiť: `sudo ./secret -l`,
- na klientovi, ktorý zasiela súbory spustiť: `sudo ./secret -r textfile.txt -s 147.229.228.188`,
- zastaviť prijímanie Ctrl+C,
- súbor je uložený v koreňovom adresári.

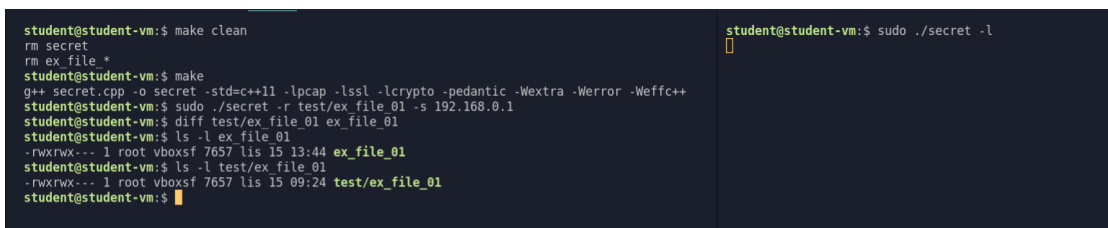
3.9 Testovanie

Testovanie prebiehalo v priebehu vývoja podľa jednotlivých implementačných krokov.

Preklad a zostavenie bolo úspešne otestované aj na školskom servere Merlin.

Pred behom zobrazených testov boli spustené príkazy `make clean`.

Na obrázku 3 je ukázaný prenos textového súboru použitím ICMPv4 paketov. Veľkosť súboru je väčšia ako veľkosť dátovej časti jedného paketu, takže na prenos bolo použitých viac ICMPv4 paketov.



```
student@student-vm:~$ make clean
rm secret
rm ex_file.*
student@student-vm:~$ make
g++ secret.cpp -o secret -std=c++11 -lpcap -lssl -lcrypto -pedantic -Wextra -Werror -Weffc++
student@student-vm:~$ sudo ./secret -r test/ex_file_01 -s 192.168.0.1
student@student-vm:~$ diff test/ex_file_01 ex_file_01
student@student-vm:~$ ls -l ex_file_01
-rwxrwx--- 1 root vboxsf 7657 lis 15 13:44 ex_file_01
student@student-vm:~$ ls -l test/ex_file_01
-rwxrwx--- 1 root vboxsf 7657 lis 15 09:24 test/ex_file_01
student@student-vm:~$
```

```
student@student-vm:~$ sudo ./secret -l
[]
```

Obr. 3: Testovanie zasielania paketu použitím ICMPv4 paketu

Na obrázku 4 je ukázaný prenos obrázku použitím ICMPv6 paketov. Veľkosť súboru je väčšia ako veľkosť dátovej časti jedného paketu, takže na prenos bolo použitých viac ICMPv6 paketov.

Po získaní súborov boli súbory porovnané.



```
student@student-vm:~$ make clean
rm secret
rm ex_file.*
student@student-vm:~$ make
g++ secret.cpp -o secret -std=c++11 -lpcap -lssl -lcrypto -lpthread -Wextra -Werror -Wfloat-equal
student@student-vm:~$ sudo ./secret -r test/ex_file_02.webp -s fc00::
student@student-vm:~$ git diff --no-index ex_file_02.webp test/ex_file_02.webp
student@student-vm:~$ ls -l test/ex_file_02.webp
-rwxrwx--- 1 root vboxsf 6892 lis 14 17:59 test/ex_file_02.webp
student@student-vm:~$ ls -l ex_file_02.webp
-rwxrwx--- 1 root vboxsf 6892 lis 15 14:13 ex_file_02.webp
student@student-vm:~$
```

Obr. 4: Testovanie zasielania paketu použitím ICMPv6 paketu

4 Záver a poďakovanie

V doterajšom priebehu semestra som sa na cvičení a na prednáškach vďaka vyučujúcim dobre oboznámil s danou problematikou. Pri tvorení projektu som si taktiež prakticky odskúšal získané vedomosti. Som si istý, že získané znalosti se mi hodia aj v nasledovnej praxi.

Literatúra

- [1] J. POSTEL.: RFC 792 INTERNET CONTROL MESSAGE PROTOCOL [online]
[cit. 13.11.2021]
<https://datatracker.ietf.org/doc/html/rfc792>
- [2] Manual page of PCAP [online] [cit. 13.11.2021]
<https://www.tcpdump.org/manpages/pcap.3pcap.html>
- [3] Manual page of AES_encrypt [online] [cit. 13.11.2021]
https://man.openbsd.org/AES_encrypt.3
- [4] Manual page of getopt [online] [cit. 13.11.2021]
<https://www.man7.org/linux/man-pages/man3/getopt.3.html>