Síťové aplikace a správa sítí

Dokumentácia k projektu Přenos souboru skrz skrytý kanál

Autor: Samuel Olekšák (xoleks00)
Akademický rok: 2021/22

Obsah

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.

Upřesnění zadání: Program zpracuje vstupní argumenty, načte soubor, zašifruje ho a zašle skrz ICMP zprávy na zvolenou IP adresu, kde program, spuštěný v listen (`-l`) módu, tyto zprávy zachytí, dešifruje a soubor uloží na disk.

- Program může používat pouze ICMP zprávy Echo-request/reply.
- Pro správné chování bude třeba definovat protokol pro přenos dat. (např. je třeba zaslat jméno souboru, ověřit, že soubor byl přenesen celý, apod.) Tento protokol je na vašem uvážení a definujte ho v rámci dokumentace.
- Jako šifru použijte AES, dostupnou např. pomocí knihovny openssl[1]. Jako klíč použijte svůj login.
- Program se musí vypořádat se souborem větší, jak max. velikost paketu na standardní síti (1500B), tj. musí být schopen větší soubor rozdělit na více paketů.
- Můžete uvažovat, že v rámci přenosu nedojde ke ztrátám paketů. Pokud implementujete formu spolehlivého přenosu, uveďte to v dokumentaci.
- Při vytváření programu je povoleno použít hlavičkové soubory pro práci se sokety a další obvyklé funkce používané v síťovém prostředí (jako je `netinet/*`, `sys/*`, `arpa/*` apod.), knihovny pro práci s vlákny (`pthread`), pakety (`pcap`), signály, časem, stejně jako standardní knihovnu jazyka C (varianty ISO/ANSI i POSIX), C++ a STL a knihovnu SSL. Další knihovny nejsou povoleny.

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

2 Teória

2.1 Zachytávanie paketov

Zachytávač paketov (packet sniffer) je pasívny prijímač, ktorý zaznamenáva kópiu každého paketu, ktorý prechádza okolo. Zachytené pakety sa následne dajú analyzovať, aj na citlivé informácie. Software na zachytávanie paketov je voľne dostupný na rôznych webových stránkach a v komerčne dostupných produktoch. Keďže je zachytávanie paketov pasívne – čiže, zachytávače nevkladajú pakety do kanálu – sú zachytávače náročné na detekciu. Takže ak posielame pakety do bezdrôtového kanálu, musíme akceptovať možnosť, že pakety môžu byť zachytávané a ukladané niekým iným. Na zabránenie zneužitia sa používajú riešenia zahrňujúce kryptografiu.[5]

Na zachytávanie paketov v tomto projekte je využitá prenosná C/C++ knižnica na zachytávanie sieťovej premávky `libpcap`.[4]

2.2 Analýza paketov

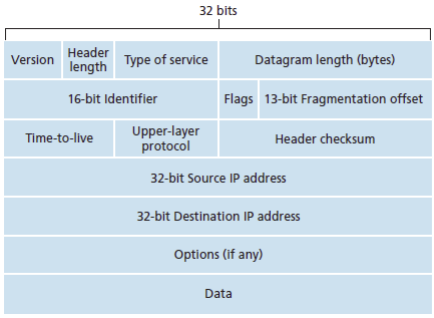
V projekte je uvažované iba zachytávanie ethernetových paketov. Knižnica `libpcap` zachytávané ethernetové pakety vydáva na 2. vrstve – čiže paket má 14 bytovú ethernetovú hlavičku.

2.2.1 Ethernetový rámec

V ethernetovom rámci má hlavička fixnú dĺžku 14 bytov – na prvých 12 bytoch je uložená zdrojová a cieľová MAC adresa a v nasledujúcich 2 bytoch je udaný Ethertype – identifikácia protokolu vyššej vrstvy.[1]

2.2.2 Hlavička IPv4 paketu

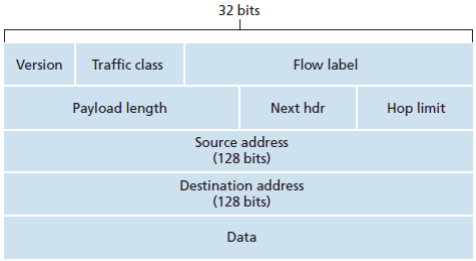
IPv4 datagram môže obsahovať variabilné množstvo nastavení, preto položka *Header length* obsahujúca 4 bity je potrebná na určenie, kde sa končí hlavička a začínajú dáta IPv4 datagramu. Väčšina IP datagramov neobsahuje nastavenia, preto je typická dĺžka IPv4 hlavičky 20 bytov.[6] Minimálna hodnota tohto poľa je 5, čo indikuje dĺžku 5×32 bitov $= 160$ bitov $= 20$ bytov.[3] Všetky položky hlavičky sú zobrazené na obr. 1.



Obr. 1: Štruktúra IPv4 paketu[5]

2.2.3 Hlavička IPv6 paketu

Formát IPv6 datagramu je ukázaný na obr. 2. Naproti IPv4 má IPv6 hlavička fixnú dĺžku 40 bytov. Položka *Next hdr* indikuje protokol vyššej vrstvy tohto datagramu. Zdrojová a cieľová adresa má 32 bitov jej formát je bližšie popísaný v RFC 4291[2].



Obr. 2: Štruktúra IPv6 paketu[5]

2.2.4 Hlavička ICMP paketu

2.2.5 Typy ICMP paketov

3 Implementácia

3.1 Využité knižnice

- `openssl/aes.h` – AES šifrovanie,

[2]https://tools.ietf.org/html/rfc4291

- `pcap/pcap.h` – vysokoúrovňové rozhranie na zachytávanie paketov[3],
- `getopt.h` – spracovanie argumentov príkazového riadka,
- `arpa/inet.h` – konvertovanie IP adres do čítateľnej podoby,
- `netinet/ether.h` – spracovanie ethernetovej hlavičky a prevod MAC adres do čítateľnej podoby,
- `netinet/ip.h` – spracovanie IPv4 hlavičky,
- `netinet/ip6.h` – spracovanie IPv6 hlavičky,
- `netinet/ip_icmp.h` – spracovanie ICMP hlavičky,
- `netinet/icmp6.h` – spracovanie ICMPv6 hlavičky
- `netdb.h` – preklad doménového mena na IP adresu.

3.2 Protokol

Namiesto ICMP echo hlavičky (32 bitov) sa zapisuje 12 fixných bitov `0xdac`, ktoré slúžia na odlíšenie od iných ICMP echo správ negenerovaných našim klientom, následne 4 bity reprezentujúce počet vyplňovacích (padding) bajtov, a potom 16 bajtov, ktoré značia poradie paketu v rámci prenosu jedného súboru. Ďalej je v každom pakete uložený názov súboru, ukončený znakom `NULL`, ktorý oddeľuje názov súboru od zvyšku paketu obsahujúceho segment obsahu prenášaného súboru.

3.3 Šifrovanie a dešifrovanie

Na šifrovanie a dešifrovanie je použitá 128-bitová CBC šifra s pomocou knižnice `openssl/aes.h`. Keďže veľkosť bloku je 16 bajtov, bolo potrebné veľkosť posielaných dát v bajtoch stále zaokrúhliť na najbližší vyšší násobok 16, pričom prázdne miesta boli vyplnené `NULL`. Pre každý paket sa posiela v hlavičke počet vyplňovacích bajtov, aby sa po dešifrovaní tieto `NULL` znaky nezapísali do preneseného súboru.

3.4 Nastavenie pcap

Knižnica pcap umožňuje zachytávať a vypisovať pakety v reálnom čase po zapnutí *immediate* módu pomocou funkcie `pcap_set_immediate_mode`. Buffer na príchodzie pakety je nastavený na veľkosť 256 MiB funkciou `pcap_set_buffer_size`, aby bol

[3]https://www.tcpdump.org/manpages/pcap.3pcap.html

schopný prijať dostatočne veľké súbory. Na odfiltrovanie paketov iných ako ICMP echo správy je použitý filter `"icmp[icmptype]=icmp-echo or icmp6[icmp6type]=icmp6-echo"`.

3.5 Spracovanie Ethernet hlavičky

Na jednoduchšie spracovanie Ethernet hlavičky poskytuje knižnica `net/ethernet.h` štruktúru `ether_header`. Po pretypovaní ukazovateľa na obsah paketu na ukazovateľ na túto štruktúru sa dá extrahovať typ paketu vyššej vrstvy, ktorý je dôležité poznať pred nasledujúcim krokom. Rovnaká knižnica taktiež poskytuje konštanty reprezentujúce hodnoty jednotlivých typov ethernetového paketu[2]:

- `ETHERTYPE_IP` (0x0800) – IPv4 paket,
- `ETHERTYPE_IPV6` (0x86DD) – IPv6 paket.

3.6 Spracovanie hlavičky IP protokolov

Podľa položky ethertype z predošlej hlavičky vieme určiť protokol transportnej vrstvy. Z hlavičky potrebujeme zistiť taktiež zdrojovú a cieľovú adresu. Keďže ethernetová hlavička má konštantnú dĺžku 14 bytov, presne vieme o koľko bytov je potrebné sa posunúť v pakete na nájdenie hlavičky nižšej vrstvy. Knižnice `netinet/ip` a `netinet/ip6` poskytujú štruktúry na jednoduché vybratie IP adresy z IP hlavičky `iphdr` a `ip6_hdr`.

4 Bonusové rozšírenia

4.1 Testovací skript

V archíve je priložený testovací skript `test.py` spolu s priečinkom `test/` obsahujúcim súbory na testovanie. Skript pre každý z testov spustí server a následne klienta spolu s testovacím súborom a testovacou IP adresou (odkazujúcou sa loopback alebo lokálnu adresu). Po uplynutí niekoľkých sekúnd sa server zastaví a prijatý súbor sa porovná so súborom, ktorý sa odoslal a ak sa zhodujú, test je úspešný.

Jednotlivé testy sa medzi sebou líšia v charaktere odoslaného súboru (zmestí sa do jedného paketu/viacerých paketov, typ súboru (plain text, binárne dáta), obsahuje Unicode znaky atp.) a v type adresy (IPv4, IPv6, doménové meno).

Zdroje

[1] *Ethernet frame*, online. URL: https://www.wikiwand.com/en/Ethernet_frame.

[2] *IEEE 802 Numbers*, online. URL: https://www.iana.org/assignments/ieee-802-numbers/ieee-802-numbers.xhtml.

[3] *IPv4*, online. URL: https://en.wikipedia.org/wiki/IPv4.

[4] *Man page of PCAP*, online. URL: https://www.tcpdump.org/manpages/pcap.3pcap.html.

[5] a. K. W. R. Kurose James F., *Computer networking: a top-down approach*, 7. vyd. Pearson, 2017, ISBN: 0133594149.

[6] J. Garud, *IPv4 Datagram Format*, online, máj 2016. URL: https://electronicspost.com/ipv4-datagram-format/.