ISA PROJEKT

Přenos souboru skrz skrytý kanál

Richard Hrmo

OBSAH

1.Šifrovanie súboru

Ako šifra je použitá AES podľa zadania ktorá sa nachádza v knihovni openssl.

1.1. Vytvorenie kľúča

Na vytvorenie kľúča na zašifrovanie súboru je využitá funkcia AES_set_encrypt_key() a na dešifrovanie AES_set_decrypt_key(). Ako kľúč je použitý login „xhrmor00".

1.2. Šifrovanie a dešifrovanie dát

Pomocou funkcie AES_encrypt() sú dáta šifrované a pomocou funkcie AES_decrypt() sú dáta dešifrované. Tieto funkcie dokážu ale šifrovať a dešifrovať iba v úsekoch 16b. Tým pádom bolo potrebné rozdeliť dané dáta na úseky po 16B a až potom ich šifrovať (dešifrovať). Ak sú dáta menšie ako úsek 16b, tak funkcia automaticky doplní zvyšné bity, ale po dešifrovaní musí program tieto zvyšné bity odstrániť, inak by do súboru zapísal „bordel" čo tam nepatrí.

2. Odosielanie dát súboru

2.1. Socket

Pri odosielaní dát je využitá knižnica netdb.h, v ktorej sa nachádza aj knižnica socket.h. Informácie potrebné pre socket sú získané pomocou funkcie getaddrinfo(). Potom ako sú uložené do socketu dáta sú odosielané pomocou funkcie sendto(). Ak sú dáta posielané príliš rýchlo môže sa stať že táto funckia vyhodí error pretože sa ešte nestihol odoslať posledný paket. Tento problém je vyriešený pomocou funckie poll(), ktorá sa nachádza v knižnici poll.h.

2.2. ICMP a secret prokoly

Podľa zadania sú využité len ICMP pakety. Ako kód pre icmp je použité ICMP_ECHO. Type ani Checksum nijak kód nerieši. Po ICMP protokole je do paketu uložený secret protokol, v ktorom sa nachádza číslo pomocou ktorého sa určí ktoré pakety zahodiť a ktoré spracovať. Názov súboru a veľkosť prenášaných dát ktoré sú ďalej v súbore mali byť tiež súčasťou tohto protokolu ale to už som nestihol upraviť. Kvôli tomuto nedostatku sa môže stať že ak je klient rýchlejší ako server a posiela pakety rýchlejšie ako ich stíha server spracovať nespracujú sa všetky (nové pakety vyhodia tie staré z bufferu). Tento problém sa mi nepodarilo vyriešiť, jediné čo mi napadlo je použiť usleep() pri odosielaní paketov, ale to podstatu problému nevyrieši, takže som ho v kóde zakomentoval.

3. Prijímanie súboru

Serverová časť projektu bola inšpirovaná ISA examples – pcap-filter.c, autor Matoušek Petr, Ing., Ph.D., M.A..

3.1. PCAP

Pri prijímaní paketov je využitá knižnica pcap.h. Prijímanie packetov sa spustí pomocou funkcie pcap_open_live(). Pomocou funkcie pcap_setdirection() je nastavené spracovávanie iba packetov ktoré sú prijaté. Filter pre spracovávanie icmp alebo icmpv6 paketov je vytvorený a nastavený pomocou funkcií pcap_compile() a pcap_setfilter(). Funckia pcap_loop() posiela packety do pcap_handler() kde sú spracované.

3.2. Spracovanie paketov

Pri odosielaní paketov sú do nich pridávané informácie o mene a o dĺžke dát. Pri spracovaní prvého paketu sa nastavia globálne premenné pre meno súboru a jeho dĺžku. Podľa toho a protokolu secret sa určí či paket „patrí“ serveru alebo nie. Potom sa prijímajú pakety a ukladajú sa dáta z nich do globálneho vektoru global_data až pokiaľ nepríde konečný paket. Keď príde konečný paket, tak sa dáta v globálnom vektore global_data dešifrujú a uložia sa do súboru, ktorého názov je uložený v global name. Po uložení dát sa globálne premenné vynulujú.

4. Potrebné knižnice a ich minimálne verzie

unistd.h – verzia 2.1
string.h – verzia 2.1
fstream – verzia 3.1
openssl/aes.h
netdb.h – verzia 2.1
netinet/ip_icmp.h – verzia 2.1
vector – verzia 3.1
pcap.h
netinet/ether.h – verzia 2.1
sys/poll.h – verzia 2.1