Vysoké učení technické v brně

Fakulta Informačních technologií

ISA – Přenos souboru skrz skrytý kanál

Síťové aplikace a správa sítí

Ladislav Dokoupil

Obsah

Úvod

Cílem projektu bylo vytvořit komunikující aplikaci pro posílaní zašifrovaných souborů přes síť v jazyce C/C++. Data jsou posílána bez záruky doručení jak pomocí IPv4, tak IPv6 přes protokol ICMP. V případě jakékoli chyby při přenosu je celý soubor zahozen, v opačném případě je uložen v aktuálním adresáři. Server je schopen přijímat více souborů současně, přičemž každý stream dat odlišuje pomocí identifikátoru uloženého v ICMP hlavičce.

Implementace

Aplikace je rozdělena do několika zdrojových souborů a souborů dokumentace. Všechny zdrojové a hlavičkové soubory jsou v adresáři src. Dokumentace a soubor Makefile určený k překladu jsou ve zdrojovém adresáři.

Hlavní funkce

Hlavní funkce je určena ke zpracování argumentů příkazové řádky a určení primárního chodu programu. Nejdříve jsou zpracovány argumenty, v případě, že je zachycen argument h nebo l, je okamžitě proveden jejich úkon. V případě přepínače l se aplikace přepne na režim serveru. Pokud se objeví přepínač h nebo je program spuštěn bez jakéhokoliv přepínače, je vypsána nápověda a následně program ukončen. Pokud se vyskytují oba přepínače r a s, spustí se režim klienta a začne se odesílat určený soubor na stanici serveru.

Šifrování a vlastní protokol

K řádnému odeslání souboru bylo potřeba vyvinout vlastní hlavičku nad šifrovanými daty. Nejdříve bylo potřeba rozeznat o jaký typ komunikace se jedná, pro mé účely stačily kategorie NAME, DATA a END, které jsou pro posílaní názvu souboru, dat a oznámení ukončení komunikace. K realizaci tohoto úkonu jsem do vlastní hlavičky zavedl výčtový typ s těmito předdefinovanými hodnotami. Dále bylo nutné data šifrovat, přičemž implementace AES vyžaduje 16B bloky dat. V dané implementaci stačilo zaručit, že komunikace nebude kratší jak 16B. To jsem zaručil případným vyplněním nulami pro kratší datagramy a do hlavičky jsem přidal celočíselný atribut určující, kolik bajtů se má z doručeného datagramu odstranit po dešifrování. AES šifrování je řešeno 16B klíčem ve tvaru loginu, doplněného nulovými bity.

Server

Pro poslouchaní paketů je využita knihovna pcap. Pro server jsou využity funkce listen a packet_handler. Ve funkci listen je inicializována knihovna pcap, což zahrnuje nastavení rozhraní k naslouchaní komunikace, nastavení naslouchacího filtru na pakety typu ICMP, pro odfiltrování nedůležité komunikace a následně je zavolána funkce packet_handler, ve které jsou zpracována všechna přijatá data.

Ve funkci packet_handler je zpracován každý rámec, kde jsou zpracovávány hlavičky paketů. Pro umožnění příjímání komunikace více klientu a kontrolu chyb jsou navíc podle příslušného ID z hlavičky data ukládána do příslušného streamu a je kontrolována kontinuita posloupnosti paketů, zda nedošlo ke ztrátě paketu. Data jsou dešifrována a následně jsou zahozeny potenciální přebytečné bajty a dále zpracována podle předdefinovaného typu komunikace v hlavičce.

Implementace serveru umožňuje současný zápis více souborů, každý soubor je určen identifikátorem z ICMP hlavičky, který je na straně klienta generován náhodně. V případě kolize je ukládání souborů kolizních komunikací zastaveno.

Klient

Inicializace odesílaní je provedena funkcí send_icmp, kde je realizována resoluce adresy otevření souboru pro čtení, načtení potřebných dat a inicializace typu komunikace pro implementovaný protokol. Vlastní odesílání dat je realizováno ve funkcí send, která navíc provádí rutinní funkce pro přehlednější kód, jako počítaní kontrolního součtu a volání funkce šifrování paketů.

Network.h/.cpp

Soubor network.cpp obsahuje implementace pomocných funkcí pro zbytek programu. Jsou zde implementovány struktury hlaviček pro vlastní definovaný protokol.

Funkce checksum pro výpočet kontrolního součtu do hlavičky ICMP paketu přejata z [4]. A funkce Encrypt, decrypt pro a dešifrování dat dle [5].

Použití

Program je přeložen díky přiloženému Makefile souboru s příkazem make, výsledný soubor je uložen pod názvem ”secret”.

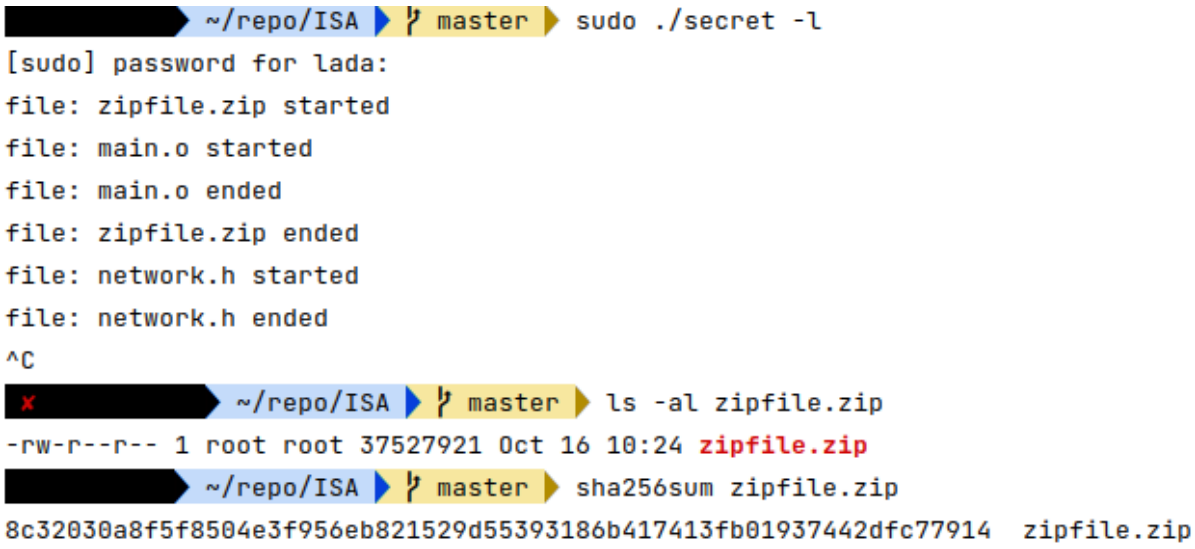
”./secret [-h] [-l] [-s HOSTNAME] [-r FILE]”

Pro vypsání nápovědy je program spuštěn pomocí parametru ”-h” nebo bez jakýchkoliv parametrů. Spuštění v režimu serveru je realizováno pomocí přepínače ”-l”. Spuštění v režimu klienta je realizováno pomocí specifikování kombinace přepínačů ”-s [HOSTNAME] -r [FILE] ”, kde HOSTNAME je IP adresa, nebo jméno hosta a FILE je název existujícího souboru.

Testování

Testování bylo primárně prováděno mezi dvěma počítači v lokální sítí s použitím IPv4 a IPv6 směrování. Pro kontrolu odolnosti vůči rušení bylo testováno současné posílaní více souborů.

Následná verifikace správného doručení dat byla prováděna programem ls pro kontrolu velikosti a s pomocí hashovací funkce pro kontrolu integrity.



Obrázek 1 ukázka příjímání souborů a ověření integrity

Závěr

Podařilo se mi implementovat všechny části aplikace dle zadání projektu, navíc je aplikace schopna přijímat více souborů současně.

Pro budoucí verze programu by bylo vhodné implementovat funkce pro spolehlivý přenos souborů, což by zahrnovalo odesílaní identifikátoru posledního zpracovaného paketu ze serveru a jeho zpracování na straně klienta, jelikož v současné implementaci je možné, že dojde ke ztrátě paketů a následnému zahození souboru.

Zdroje

[1] Tcpdump&libpcap. [online]. 2010 [cit.2021-10-16]. Dostupné z: https://www.tcpdump.org/pcap.html

[2] RFC 792 - INTERNET CONTROL MESSAGE PROTOCOL. Tools.ietf.org [online]. 1981 [cit.2021-10-16]. Dostupné z: https://tools.ietf.org/html/rfc792

[3] RFC 4443 - INTERNET CONTROL MESSAGE PROTOCOL (ICMPv6). Tools.ietf.org [online]. 2006 [cit.2021-10-16]. Dostupné z: https://tools.ietf.org/html/rfc4443

[4] RFC 1071 - Computing the Internet checksum. Tools.ietf.org [online]. 1988 [cit.2021-10-16]. Dostupné z: https://tools.ietf.org/html/rfc1071

[5] Opensl. *Wiki.openssl.org* [online]. [cit. 2021-10-16]. Dostupné z: https://wiki.openssl.org/index.php/EVP_Symmetric_Encryption_and_Decryption