

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Sít'ové aplikace a správa sítí  
Přenos souboru skrz skrztý kanál

# Obsah

<b>1</b>	<b>Zadání</b>	<b>2</b>
<b>2</b>	<b>Spuštění</b>	<b>2</b>
<b>3</b>	<b>Návrh</b>	<b>2</b>
<b>4</b>	<b>Argunemty</b>	<b>2</b>
<b>5</b>	<b>Klient</b>	<b>3</b>
5.1	Načtení vstupních dat . . . . .	3
5.2	Šifrování . . . . .	3
5.3	Skládání a odesílání paketu . . . . .	3
<b>6</b>	<b>Server</b>	<b>4</b>
6.1	Spuštění serveru, pthread . . . . .	4
6.2	Zpracování příchozích paketů . . . . .	4
6.3	Zpracování přijatých dat . . . . .	4
<b>7</b>	<b>Hlavička ICMP_FILE</b>	<b>4</b>

## 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 <iplhostname> -r <file> [-l]`

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

Příklad souboru s uloženými servery pro filtraci:

## 3 Návrh

Jako jazyk k implementaci 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 popsány v sekci *Spuštění*. Pokud nejsou některé požadované parametry uvedeny, program vypíše nápovědu a úspěšně se ukončí. V případě zadání parametru `-l` nejsou vyžadovány další argumenty, vzhledem k tomu že server nepotřebuje cílovou ip adresu ani specifikovat soubor pro přenos.

## 5 Klient

### 5.1 Načtení vstupních dat

Po spuštění aplikace s parametry klienta je zkontrolována daná ip addressa/hostname pomocí funkce *getaddrinfo()*, která vrátí 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, tudíž nezáleží na jeho příponě. Soubor je načítán najednou tudíž je jeho velikost je omezena velikostí dostupné RAM na kterém je aplikace spuštěna.

### 5.2 Šifrování

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. Šifrování probíhá pomocí knihovny *OpenSSL*, konkrétně pomocí šifry AES. Šifrování a rozšifrování souboru pak probíhá v daných funkcích v souboru *aes.c*.

### 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 jejích dat jsou vytvořené potřebné hlavičky a payload paketu, 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 *sendto()* a celý tento 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 vytvořeny dvě vlákna, aby mohly být přijímány současně IPv4 a IPv6 pakety. Jednak je potřeba pro každý typ paketů otevřít socket s jinými parametry, ale také aby se program nezdržoval např. při přenosu pomocí IPv4 paketů s náhodnými ICMPv6 pakety. Po vytvoření těchto vláken program čeká na příchozí pakety, které byly zachyceny.

### 6.2 Zpracování příchozích 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

## 7 Hlavička ICMP\_FILE