NETWORK APPLICATIONS AND NETWORK ADMINISTRATION

2020/2021

Transfer encrypted files through hidden channel

Šesták Pavel(xsesta07)

Brno, November 13, 2021

Contents

1 Introduction

This is manual for TUI utility secret. This utility allow to open server and send encrypted files.

1.1 Usage

./secret -r <file> -s <ip|domain> | <-l>

Square brackets represent mandatory arguments, arguments in curly brackets are optional. In case of some error during executing program non zero return value is returned.

1.1.1 Parameters

-r <file> file which will be transfered
-s <ip|domain> Destination, can be specified by IPV6, IPV4, hostname and domain.
-l secret in server mode (sudo required)

1.2 Scenario of usage

We want to transfer file sample.pdf from client A to server B.

1.2.1 Server

#./secret -l

1.2.2 Client

$./secret -r sample.pdf -s B

2 Dependencies and makefile

This sniffer was implemented in c++ with libraries m, crypto, ssl, pcap.

2.1 Dependencies

For successfully build you need to install libraries.

2.2 Makefile

2.2.1 Build

This program was build with c++ compiler, libraries need to be linked.
Build command: make build
Program can be compiled in debug mode, which add debug prints to standard error output.
Debug command: make debug
Solution after build create object files. To delete binary files launch clean.
Clean command: make clean

3 Protocol design

First 8 bytes contains random generated protocol identifier 0xA7E0C13E9815EDF0, next 8 bytes containt length of message, following 16 byte containt initial vector for AES encryption. After this headers is encrypted message with lenght specified in header. Encrypted message contains name of file followed by zero byte. Residue of message is content of file.

4 Implementation

4.1 Parsing arguments

In main is called function argumentParse. Function has parameters argument count, array of arguments and reference to variables to fill info about arguments. Arguments are compared by if-tree with function compare. Depends on arguments program start sniffing packets or sending file to server.

4.2 Listening

Program first setup pcap for sniffing packets with function pcap_open_live. Then set pcap filter "(icmp or icmp6) and(icmp[icmptype] == icmp-echo or icmp6[icmp6type] == icmp6-echo)" to capture just icmp packets. Then set data link offset. Sniffing is processed by pcap_loop function and each packet is handle by function processPacket.

4.2.1 ProcessPacket

This function is called when packet is capture with pcap. First we need unwrap our ICMP packet. We need to check if the transfer is in progress. When server is in idle state, first packet contain headers about payload. We check protocol identifier. Initial vector is extracted and message size from our protocol header. Check if icmp identier is equal to current transfer id and if ip addresses match. When check fail, time is compared. When last succesfull packet was accept 15 seconds ago server will be reseted. Content is readed and append to whole message. When whole file was read message is decrypted and saved to file. Variables are reseted to default values and server may accept next file.

4.3 Sending

Program first resolute destination, which may be ip address, hostname or domain name. Destination is received from function getaddrinfo. File specified by param is readed into memory. Initial vector is generated and message encrypted. Encrypted text is send by function send_content_icmp.

4.3.1 send_content_icmp

This function fill our ICMP packet. First packet must start with headers of our protocol. While whole message is not send set type of icmp packet to echo and copy part of message into packet data. One packet is set up to 1000 bytes. Sequence number is set and packet send with function send. After sendign packet we wait to answer to prevent full buffer. After file is send memory is freed and program exit with zero exit code.

5 Testing

Program was tested in local wireless network between Ubuntu in virtual machine and Arch. While testing was send plain text file, pdf file, image and mp4 video file. Transfer of small files was without problem, but video af transfer cannot be decrypted.

6 Conclusion

Basic features are successfully implemented but there is lot of things to improve. Use function poll to check if its possible to write into buffer. Packet is classified as our on first 8 bytes, what happens when other ICMP packet match this prefix?