Reverse Design Plan

170D WOBC: Module K Exam II-A

CW2 Kyle Spicer

Due Date: 21 October 2022

1 Project Summary

Given a various amount of .pcap files that have been encrypted with top-secret alien messages. The task is to create a command-line utility that will decrypt the message, reverse the direction of command, re-encrypt the new message and write the result to a out.pcap file.

If the encrypted file contains a message, we are to print the original message to std-out, then prepend the message with Don't and replace the first letter of the original message with a lowercase letter.

Usage: ./reverse (128-bit key) (input file)

2 Architecture

2.1 Directories

2.1.1 reverse (top level directory)

- Makefile
- tls_crypto.h
- libtls_crypto.a

2.1.2 src

- reverse.c (main program)
- reverse_funcs.c (holds function declarations)
- reverse.h (header file for program)

2.1.3 test

- check_reverse.c (unit test for program)

2.1.4 doc

- design.pdf
- writup.pdf
- testplan.pdf
- reverse.1

2.2 Structures

2.2.1 Alien Encrypted Message Structs

- struct file_header_t;
- struct packet_header_t;
- struct ethernet_frame_t;
- struct ip_header_t;
- struct tcp_header_t;
- struct tls_header_t;

2.2.2 Alien Message / Movement Structs

- struct alien_msg_pyld_t;
- struct alien_mvmt_pyld_t;

3 Program Flow

1. receive the command line arguments
2. open/read the binary file
3. open the out.pcap file
4. read in the .pcap file contents and pack structures
5. decrypt the key provided
6. decrypt message with newly decrypted key
7. pack additional structure for message contents
8. check payload type (message or movement)
9. complete appropriate logic for each
10. re-encrypt message
11. write updated contents to out.pcap
12. free all memory, close all files, verify with valgrind

4 Timeline to Completion

4.1 Review Rubric and Project Instructions : NLT T+1

- Read thoroughly to understand entire scope of project.
- Prepare questions for any items that need clarification.

4.2 Create GitLab Repository, Create Directories and Files : NLT T+1

- create working directory
- initialize repository, using proper naming for all directories/files.
- prepare Makefile

4.3 Prepare Design Plan : NLT T+2

- create Overleaf documentation
- conceptualize steps to complete project

4.4 Start Writing Program : NLT T+2

- read in .pcap file
- parse arguments
- validate file type and implement error messages

4.5 Main Programming : NLT T+3 to T+9

- build structures and shells of known functions
- fine tune logic and program flow

4.6 Documentation and Finalization

- last few days will be completing documentation and unit testing - review rubric and ensure all requirements are met - verify valgrind is leak and error free

5 Topics to Research

- .pcap files (how to read, organize, and manipulate)
- wireshark (for assisting with pcap files)
- encapsulation
- packing structures properly

6 Extra Credit Items

- create manpage reverse.1