Vysoké Učení Technické v Brně
Fakulta informačních technologií

Documentation for IPK - Project 2

Packet sniffer Implementation

Marek Sarvaš (xsarva00) Brno, 11.12.2019

Contents

1 About

ABOUT

2 Implementation

Sniffer is implemented in one file ipk-sniffer, whole program is divided into few functions and main function.
MAIN First part of main function is for parsing given arguments using check_args function.If interface was not given as argument all interfaces are printed in loop. If program got interface (or other optional arguments such as port, tcp, etc.) Firstly opens given interface for sniffing using pcap _open _live, on success compile given filter composed from given program arguments - tcp, udp, port. If compiled successfuly filter is applied on interface handler. For actual sniffing pcap _loop function is used with arguments interface handler, number of packets to be sniffed (stored int argument structure), callback function (documented below). There is no time limit in which packet has to be sniffed, because if user wants to sniffed eg. 2 packets program will run until 2 packets are sniffed. After wanted number of packets is sniffed programme closes interface handler and frees allocated memory. Otherwise if interface where tcp/udp packets could be sniffed were given sniffer will run infinitely until interuption (eg.: with CTRL+C).
CHECK _ARGS Verify arguments given to program using getopt, in case of -p (port filter) convert port number into integer and checks its correct value which has to be between 0 and 65535.
CREATE _FILTER Creates a string filter using tcp, udp, port number from values given as programme arguments eg.: "port 80", "tcp", "udp port 5353" etc. Because programme is sniffing only tcp or udp packets filter "tcp udp" is same as none.

CALLBACK
Function passed into pcaploop called for every packet sniffed. Is responsible for parsing packet to get necessary information such as: time, protocol of packet, source and destination ports and ip addresses; resolving ip addresses into names and printing these information and whole packet on standard output.

3 Testing

testing