Peter Kúdela

PKS

Zadanie 1

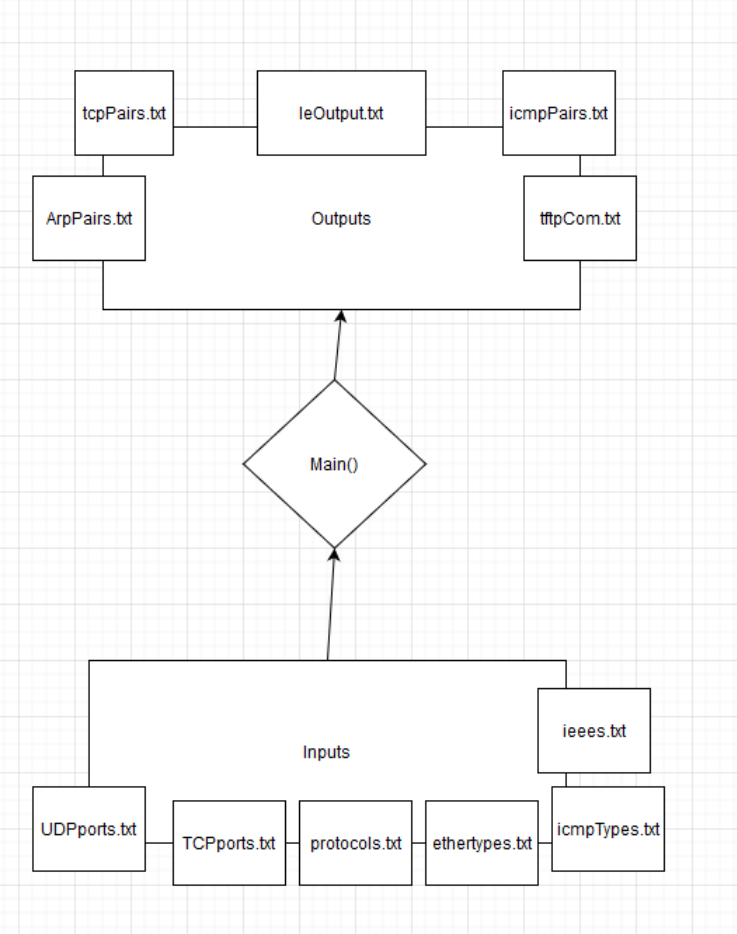
Analyzátor sieťovej komunikácie

21. 10. 2020

Zadanie úlohy

Navrhnite a implementujte programový analyzátor Ethernet siete, ktorý analyzuje komunikácie v sieti zaznamenané v .pcap súbore a poskytuje nasledujúce informácie o komunikáciách.

Blokový diagram súborov



Navrhnutý mechanizmus analyzovania protokolov na jednotlivých vrstvách

Vo funkcií main si najprv vyčistím súbor, do ktorého zapisujem výstup programu, nasledovne sa pokúsim načítať pcap súbor, pomocou pcap_loop na každý packet v pcap súbore spustím funkciu „forEveryPacket“. Po dokončení priebehu tejto funkcie s využitím globálneho počítadla vypíšeme všetky primajúce IP v packetoch typu IPv4 aj tú, ktorá prijala najviac packetov. Na konci voláme štyri funkcie plniace bod 4) zo zadania, arpPairs(), icmpPairs(), tftpPairs() a tcpPairs(), viac o nich nižšie.

```
void forEveryPacket(u_char* temp1, const struct pcap_pkthdr* header,      const u_char*
pkt_data)
```

Pomocou tejto funkcie analyzujeme každý packet, prvá vec ktorú funkcia urobí je zistí, či api poskytnutá dĺžka rámca je viac ako 60, ak nie vypíšeme dĺžku prenášanú po médiu 64, inak

vypisujeme dĺžku poskytnutú api + 4. Ďalej funkcia analyzuje, či je rámec typu IEEE alebo ethernet, posledná vec ktorú robí je vypisuje hex string rámca. Ak je rámec typu IEEE pomocou externého súboru ieees.txt zisťuje aký druh. Podobne ak je rámec typu ethernet zisťujeme aký ethertype, ďalej ak je ethertype IPv4 vnárame sa do funkcie workIPv4.

void workIPv4(const u_char* pkt_data, FILE** output)
Vo funkcií vypisujeme source a destination ip adresy packetu, destination adresy si pamätáme pre výpis na konci. Pomocou externého súboru protocols.txt analyzujeme s akým protocolom pracujeme, ak je protocolom TCP vnárame sa do funkcie workTCP(), ak UDP tak do funkcie workUDP() a ak ICMP tak source a destination IP ukladáme do globálnej štruktúry, nasledovne sa vnárame do workICMP()

void workICMP(const u_char* pkt_data, FILE** output, int IPv_size)
Funkcia zisťuje type name pre icmp rámec a ukladá ho do globálneho poľa štruktúr savedICMPs ktoré neskôr spracúvame funkciou icmpPairs().

void workARP(const u_char* pkt_data)
Ukladá ARP rámce do globálneho poľa savedArps ktoré neskôr spracúvame funkciou arpPairs()

void icmpPairs(), void tcpPairs(), arpPairs()
Prechádza všetkými uloženými informáciami o daných rámcoch, zgroupuje ich do komunikácií a zapisuje čísla rámcov do externého súboru {typRamca}Pairs.txt (Informácie o konkrétnych rámcoch sa nachádzajú v hlavnom výstupovom súbore leOutput.txt)

void workUDP(const u_char* pkt_data, FILE** output, int IPv_size)
Získava source a destination port z rámca, taktiež zisťuje či ide o známy port, ak ide o tftp ukladá čísla rámcov do globálneho poľa pre výpis v tftPairs().

void tftpPairs()
Vypisuje čísla rámcov nachádzajúcich sa v jednej komunikácií tftp. Tieto čísla berie z globálneho pola tftpOrd.

Príklad štruktúry externých súborov

0512 XEROX_PUP
0513 PUP_Addr_Trans
2048 IPv4
2049 X.75_Internet
2053 X.25_Level_3
2054 ARP

Porty, ethertypy etc. majú každý svoj vlastný externý súbor.

Používateľské rozhranie

Po spustení programu bude užívateľ vyzvaný na napísanie názvu pcap súboru (nachádzajúceho sa v adresári vzorky_pcap_na_analyzu.

Voľba implementačného prostredia

Program bol napísaný v jazyku C++ s použitím Visual studia.