FIIT STU

Analyzátor sieťovej komunikácie

Dokumentácia

Meno: Martin Šváb
Študijný program: Informatika
Ročník: 2, cvičenie: štvrtok 8:00
Predmet: Počítačové a komunikačné siete
Cvičiaci: Ing. Rastislav Bencel, PhD.
Akademický rok: 2020/2021

Zadanie úlohy

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

Vypracované zadanie musí spĺňať nasledujúce body:

1) **Výpis všetkých rámcov v hexadecimálnom tvare** postupne tak, ako boli zaznamenané v súbore.
2) Pre rámce typu **Ethernet II a IEEE 802.3 vypíšte vnorený protokol**. Študent musí vedieť vysvetliť, aké informácie sú uvedené v jednotlivých rámcoch Ethernet II, t.j. vnáranie protokolov ako aj ozrejmiť dĺžky týchto rámcov.
3) Analýzu cez vrstvy vykonajte pre rámce Ethernet II a protokoly rodiny TCP/IPv4: **Na konci výpisu z bodu 1)** uveďte pre IPv4 pakety:
    a) Zoznam IP adries všetkých prijímajúcich uzlov,
    b) IP adresu uzla, ktorý sumárne prijal (bez ohľadu na odosielateľa) najväčší počet paketov a koľko paketov prijal (berte do úvahy iba IPv4 pakety).
4) V danom súbore analyzujte komunikácie pre zadané protokoly:
    a) http
    b) HTTPS
    c) TELNET
    d) SSH
    e) FTP riadiace
    f) FTP dátové
    g) TFTP, **uveďte všetky rámce komunikácie**, nielen prvý rámec na UDP port 69
    h) ICMP, uveďte aj typ ICMP správy (pole Type v hlavičke ICMP), napr. Echo request, Echo reply, Time exceeded, a pod.
    i) **Všetky** ARP dvojice (request – reply)
5) v IP pakete (pole Protocol), ako aj čísla portov v transportných protokoloch boli programom **načítané z jedného alebo viacerých externých textových súborov.** Pre známe protokoly a porty (minimálne protokoly v bodoch 1) a 4) budú uvedené aj ich názvy. Program bude schopný uviesť k rámcu názov vnoreného protokolu po doplnení názvu k číslu protokolu, resp. portu do externého súboru. Za externý súbor sa nepovažuje súbor knižnice, ktorá je vložená do programu.
6) V procese analýzy rámcov pri identifikovaní jednotlivých polí rámca ako aj polí hlavičiek vnorených protokolov nie je povolené použiť funkcie poskytované použitým programovacím jazykom alebo knižnicou. **Celý rámec je potrebné spracovať postupne po bajtoch.**
7) Program musí byť organizovaný tak, aby bolo možné jednoducho rozširovať jeho funkčnosť výpisu rámcov pri doimplementovaní jednoduchej funkčnosti na cvičení.
8) Študent musí byť schopný preložiť a spustiť program v miestnosti, v ktorej má cvičenia. V prípade dištančnej výučby musí byť študent schopný prezentovať podľa pokynov cvičiaceho program online, napr. cez Webex, Meet, etc.

Riešenie úlohy

Blokový návrh

V tomto projekte sa pri spustení programu zanalyzujú vstupné pakety a uložia do listu triedy PacketInfo. V tejto triede sa nachádzajú všetky údaje ktoré potrebujem na výpis ľubovľného vstupu z tejto úlohy. Po tomto kroku môžem vypísať pakety pomocou funkcie určenej na výpis paketov. V prípade potreby spracovania konkrétnej komunikácie sa tieto údaje prefiltrujú do druhého listu, kde sa budú nachádzať iba pakety s hľadaným protokolom. Na vypís spojenia medzi 2 uzlami môžem použit funkciu group_communications, ktorá navráti slovník, ktorý ukladá packety do kľúčov – (src_socket, dst_socket) pre TCP protokoly alebo (src_ip, dst_ip} pre ARP protokol. To mi uľahčí prácu nasledovnej funkcii, ktorá vytvorí list komunikácii (triedy TCPCommunication, ARPCommunication) a tento list sa už môže poslať do funkcie na výpis komunikácie.

Mechanizmus analyzovania protokolov

Analýza paketov sa robí vo funkcii analyze_packet. Tá prebieha v tomto poradí:

- Základné informácie
  - číslo rámca
  - dĺžka rámca poskytnutú pcap API
  - dĺžka rámca prenášaného po médiu
  - celý rámec
- Linková vstva
  - typ rámca (podľa dĺžky rámca v ethertyp zložke)
  - zdrojová MAC adresa
  - cieľová MAC adresa
  - SAP – ak je IEEE 802.3 Raw, IEEE 802.3 SNAP alebo IEEE 802.3 LLC
  - ethertyp – ak je Ethernet II alebo IEEE 802.3 SNAP
- Sieťová vrstva
  - IPv4
    - zdrojová a cieľová IP adresa
    - transportný protokol pre IP
  - ARP
    - zdrojová a cieľová IP adresa
- Transportná vrstva
  - TCP
    - zdrojový a cieľový port
    - zdrojový a cieľový socket
    - flagy
    - aplikačný protokol pre TCP
  - UDP
    - zdrojový a cieľový port
    - aplikačný protokol pre UDP
- Aplikačná vrstva

Po analýze vrstvy rámca v tejto funkcii sa z neho tento rámec vymaže aby sa bajty vo funkciach nepočítali od začiatku rámca.

Externé súbory

Na preklad hexadecimálnych kódov, ktoré reprezentujú Ethertypy, SAP, transportné a aplikačné protokoly som použil súbory nachádzajúce sa v priečinku „Data". Použil som na to viac súborov, každý súbor prekladá len 1 vec. Názvy týchto súborov sú pre jednoduchosť zavolania uložené ako globálne premenné v kóde.

Patria tam súbory:

- EtherTypes.txt
- ICMP_Types.txt
- IP_Protocols.txt
- SAP.txt
- TCP_Ports.txt
- UDP_Ports.txt

V týchto súboroch sú v každom riadku 2 slová. To prvé je hexadekadický kód a druhé je jeho preklad. Tieto slová musia byť oddelené od seba 1 medzerou pričom v nich samotných musí byť namiesto medzier použitý iný vhodný znak ako napr. podčiarkovník - "_". V súbore by nemali byť prázdne riadky.

Napr.:
0200 XEROX_PUP
0201 PUP_Addr Trans
0800 IPv4
0801 X.75_Internet
...

Používateľské rozhranie:

Tento projekt neobsahuje používateľské rozhranie. Program pracuje v konzole. Prebieha tam len načítavanie názvu vstupného súboru a kódu výstupného súboru. Podrobnejšie inštrukcie na prácu v programe sa nachádzajú v súbore README.txt.

Implementačné prostredie

Projekt bol vypracovaný v jazyku Python. Na načítanie pcap súborov používam funkciu rdpcap a raw z knižnice Scapy. Tento jazyk som si vybral pre jednoduchšiu syntax a intuitívnejšiu inštaláciu nových knižníc.