POČÍTAČOVÉ A KOMUNIKAČNÉ SIETE

2023/2024

Zadanie č. 1:

Analyzátor sieťovej komunikácie

Vypracoval: Viliam Páleník

Obsah

1. Štruktúra programu

Program je rozdelený do 3 súborov:

1. assignment1.py
2. packet.py
3. connection.py

1.1 assignment1.py

V tomto súbore pomocné funkcie:

- extract_hexadecimal_data – funkcia na načítanie packetov. Vytvára z nich objekty s ktorými pracujem ďalej pri analýze
- load_protocols – funkcia na načítanie protokolov zo súboru protocols.yaml
- print_packet – funkcia na správnu štruktúru na zápis do output.yaml súboru

A funkcie na filtrovanie podľa zadaného protokolu:

- case_arp – filter na arp protokol
- case_tcp – filter na tcp protokoly

Ešte tu je aj funkcia na výpis všetkých packetov pokiaľ na vstupe nie je zadaný protokol:

- case_print_packets – táto funkcia vypíše všetky packety (úlohy 1 - 3)

1.2 packet.py

Táto trieda slúži na reprezentáciu packetov, potom pri filtroch pre dané protokoly pristupujem k jednotlivým atribútom.

Atribúty objektu môžu byť:

- frame: bytová reprezentácia packetu
- frame_number: int
- hexa_frame: string – použitý potom vo výpise do yamlu
- dst_mac: string
- src_mac: string
- frame_length: int
- wire_length: int
- frame_type: string (napr. 'ETHERNET II', 'IEEE 802.3 LLC', 'IEEE 802.3 RAW').
- ether_type: string (napr . 'IPv4', 'ARP'...).
- src_ip: string
- dst_ip: string
- protocol: string reprezentujúci protokol na sieťovej vrstve (napr. 'TCP', 'UDP')
- src_port: int
- dst_port: int
- app_protocol: string reprezentujúci protokol na aplikačnej vrstve
- arp_opcode: string (REQUEST/REPLY)
- pid: string (pre IEEE 802.3 LLC & SNAP).

- sap: string (pre IEEE 802.3 LLC).
- flags: zoznam flagov pre tcp protokol

1.3 connection.py

Trieda reprezentuje jednu komunikáciu pre tcp protokol

Atribúty:

- src_ip: string
- src_port: int
- dst_ip: string
- dst_port: int
- state: string - stav v akom sa komunikácia nachádza
- packets: list packetov

2. Úlohy 1-3

V tomto prípade program načíta packety do pamäte (reprezentujem ich ako pole packet objektov). Pri vytváraní objektu program určí všetky potrebné protokoly a atribúty podľa jednotlivých bytov packetu.

Následne prechádza všetky packety v liste a vytvorí dictionary pre správnu štruktúru v output.yaml súbore. Na konci tohto výpisu packetov vypíše ešte štatistiku pre ipv4 packety. Teda zoznam IP adries všetkých odosielajúcich uzlov a koľko paketov odoslali + ip adresu uzla s najväčším počtom odoslaných packetov

3. Úloha 4(c-e) – analýza protokolov s komunikáciou so spojením

Úlohu som si rozdelil na dve časti. V prvej časti priraďujem packety do jednotlivých komunikácií, v druhej časti korigujem štruktúru výpisu a vypíšem všetky kompletné komunikácie a prvú nekompletnú ktorú som zaznamenal

3.1 Priraďovanie do komunikácií

Na komunikácie som vytvoril dictionary do ktorého zapisujem komunikácie. Kľúč pre každú komunikáciu sú: zdrojová a cieľová ip adresa, zdrojový a cieľový port. Ak prídem na packet, ktorý nepatrí do žiadnej komunikácie, tak vytvorím nový objekt – communication, kde uchovávam aj stav v akom sa komunikácia nachádza.

Komunikácia sa začína pomocou 3-way handshaku, končí sa týmito spôsobmi (FIN,ACK,FIN,ACK), (FIN-ACK,ACK,FIN-ACK,ACK), (FIN-ACK, FIN-ACK), (FIN-ACK, ACK, FIN-ACK) alebo RST. Ak nenastane korektný hanshake alebo ukončenie, komunikácia bude v inom stave ako LAST_ACK alebo RESET. Tým budem ďalej určovať, či je komunikácia kompletná alebo nekompletná.

3.2 Rozdelenie komunikácií

V poslednej časti vypíšem kompletné komunikácie (musia byť v stave LAST_ACK alebo RESET), a prvú nekompletnú komunikáciu

4. Úloha 4(i) – analýza ARP

Postupne prechádzam všetky arp packety. Keď narazím na arp request, tak ho zapíšem do dictionary kde kľúč je zdrojová ip adresa packetu.

Keď je packet arp reply, tak prechádzam postupne všetky arp requesty. Ak k nemu nájdem pár:

- cieľová ip adresa reply = zdrojová ip adresa request
- zdrojová ip adresa reply = cieľová ip adresa request
- cieľová fyzická adresa reply = cieľová fyzická adresa request

tak tento pár pridám do poľa párov.

Na konci vypíšem všetky páry reques-reply do jednotlivých komunikácií a všetky arp requesty bez reply zaradím do jednej nekompletnej komunikácie a všetky reply do druhej.

5. Diagram