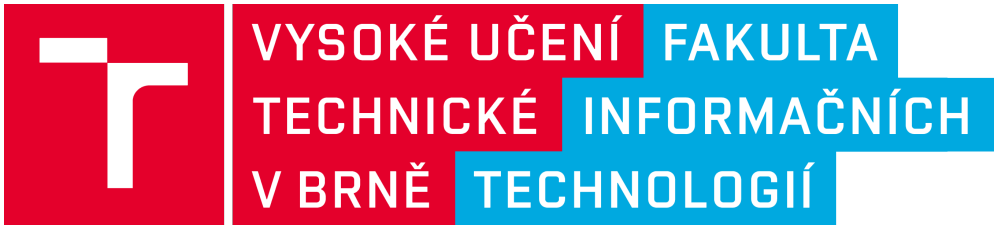

Projektová dokumentácia

Generování NetFlow dat ze zachycené síťové komunikace

ISA - Síťové aplikace a správa sítí

14. listopadu 2022

Adrián Bobola (xbobol00)

Obsah

1 Úvod

Úlohou projektu bolo naimplementovať NetFlow exportér, ktorý zo zachytených sieťových dát vo formáte pcap vytvorí záznamy Netflow. Tie následne odošle na kolektor.

Jednotlivé vstupné parametre a adresa výstupu pre kolektor je možné nastaviť užívateľom pomocou voliteľných vstupných argumentov. V prípade že užívateľ nezadá vstupné parametre, použijú sa predvolené nastavenia spustenia programu, viď 3.

2 Návrh a popis implementácie

Celá aplikácia je naimplementovaná v súbore `flow.cpp` spolu s hlavičkovým súborom `flow.hpp`. V tejto kapitole nájdete stručný popis jednotlivých časti implementácie.

2.1 Hlavičkový súbor, flow.hpp

Hlavičkový súbor `flow.hpp` obsahuje deklarácie pomocných funkcií a následovných štruktúr:

- `flow_v5_export` - Slúži na ukládanie všetkých potrebných parametrov "flowu" v5 pred tým, než bude odoslaný na kolektor. Jednotlivé položky sú dátového typu "u_int_t" podľa potrebného počtu bajtov danej položky.
- `run_params` - Slúži na uloženie všetkych vstupných argumentov pri spustení programu.
- `One_flow_params` - Slúži na uloženie informácii o jednej flow pri uschovávani v pomocnej mape.
- `timeval` - Slúži na ukládanie časových informácii v Unix formáte, tzv. "timestamp".

2.2 Štart programu, funkcia int main()

Jedná sa o prvú automaticky spúšťanú funkciu po spustení aplikácie.

Po spustení zavolá pomocnú funkciu `params_parser()`, viď 2.3. Následne dôjde k pokusu o načítanie uloženého .pcap súbora so zachytenými packetmi pomocou funkcie `pcap_open_offline`[5] a použitie filtra na pakety. Pre túto aplikáciu je filter nastavený na protokoly TCP, UDP a ICMP. Ostatné protokoly sa zahadzujú.

Potom následuje funkcia `pcap_loop()`[5], ktorá predá každý načítaný packet ďalšej pomocnej funkcii `packet_parser()`, viď 2.4.

Po sprácuvaní všetkých zachytených paketov zo súboru dôjde k ukončeniu čítania súboru pomocou funkcie `pcap_close()`[5] a k exportu možných packetov uložených vo "flow-cashi" - funkcia `export_full_flows_cache()`, viď 2.8.

2.3 Načítanie vstupných argumentov, funkcia params_parser()

Táto funkcia načíta zadané argumenty pri spustení programu. K tomu využíva pomocnú funkciu `getopt()`[6]. Pri parametri -c sa používa funkcia `gethostbyname()`[3], ktorá preloží možný zadaný hostname na IP adresu. Pri ostatných parametroch je použitá funkcia `atoi()`[2] pre prevod zadaných čísel z dátového typu string na typ int. Pri chybne zadaných parametroch je vypísaná chybová správa a beh programu je ukončený chybovým kódom.

Celá aplikácia počíta čas v mikrosekundách z dôvodu možných strát niektorých flows. Práve z tohto dôvodu sa zadané parametre -a, -i a -m ešte násobia číslom 1 000 000. Tým dostaneme zadané parametre v mikrosekundách.

2.4 Spracúvanie paketov, funkcia packet_parser()

Pri načítaní prvého paketu nastaví funkcia globálnu premennú "first_flow_SysTime" na čas z daného paketu. Tento čas bude následne používaný ako čas spustenia systému tzv. "SysUptime".

Následne funckia zistí použitý protokol z hlavičky paketu. Podľa neho sa táto funkcia rovetví na jeden z troch možných prípadov: ICMP, TCP alebo UDP. Pri použití ICMP protokolu sa porty a tcp flagy nastavia na hodnoty 0. Pri UDP protokole sa obdobne nastaví položka tcp flags na 0. Pred spracúvaním samotného protokolu sa spustí pomocná funkcia `check_timers()`, viď 2.5.

Po zistení protokolu sa vytvorí kľúč mapy, ktorá obsahuje všetky aktuálne spracúvané pakety tzv. "flow-cache". Pri tejto implementácii kľúč pozostáva zo 6 parametrov, ktorými sú:

- Zdrojová IP adresa
- Cieľová IP adresa
- Zdrojový port
- Cieľový port
- Protokol
- ToS

Následne sa vyhľadáva tento kľúč v pomocnej mape, ktorá reprezentujuje "flow-cache". Ak sa tento kľúč už nachádza v danej mape, dôjde iba k aktualizácii parametrov položky s týmto kľúčom. Ak sa ale tento kľúč nenachádza v mape, dôjde k vytvoreniu novej položky v pomocnej mape. Ešte pred samotným vložením nového záznamu dôjde ku kontrole veľkosti "flow-cache". O to sa stará pomocná funkcia `check_cache_size()`, viď 2.6.

Pri TCP protokole sa pred vložením/aktualizáciou záznamu navyše kontrolujú aj TCP flagy. Ak aktuálny paket obsahuje flag "FIN" alebo "RST', dôjde k exportu daného paketu/celej flow s rovnakým kľúčom na kolektor a dôjde k odtráneniu záznamu z flow-cash viď `export_flow()`2.7.

2.5 Kontrola active/inactive času, funkcia check_timers()

Vstupný parameter funkcie je aktuálny čas systému vo formáte timeval. Funkcia prechádza každú položku v mape (flow-cache). Pre každú položku sa vypočítajú 2 časy, jeden pre active časovač a druhý pre inactive časovač.

- Pre active časovač:
  abs[(Flow.Last čas) - (Flow.First čas)]
- Pre inactive časovač:
  abs[(aktuálny čas) - (Flow.Last čas)]

Z dôvodu zvyšenia presnosti sa pripočítavajú pri výpoočtoch sekundy prevedené na mikrosekundy spolu so zbytkovými mikrosekundami. Oba tieto vypočítané časy sa porovnávajú so vstupným parametrom -a alebo -i. V implementácii som zvolil porovnanie pomocou ostrej nerovnosti $\geq$ medzi výpočtom a parametrami, takže napr. pri SysUptime 10s dôjde k odstráneniu záznamu, ktorý je neaktívny 10s.

2.6 Kontrola počtu záznamov v cashi, funkcia check_cache_size()

Funkcia sa spustí iba v prípade, že počet položiek vo flow cashi je ostro väčší než zadaný parameter -m (počer položiek v mape $\geq$ parameter -m).

Na začiatku volania musí nájsť funkcia najstaršiu flow v cashi. Keďže použitý dátový typ std::map neumožňuje jednoducho zoradiť položky v mape podľa jednej hodnoty z kľúča mapy, musím prejsť celú mapu položku po položke. Ak má aktuálna položka čas v parametri Last menší ako najmenší zistený čas, aktualizuje sa na nový. Taktiež sa uloží aj kľúč ktorý má nižšiu hodnotu.

Z dôvodu zvyšenia presnosti sa pripočítavajú pri výpoočtoch sekundy prevedené na mikrosekundy spolu so zbytkovými mikrosekundami danej flow.

Po skončení iterovania položiek v mape máme kľúč záznamu, ktorý je najdlhšie neaktívny. Následne dôjde k odstráneniu položky s daným kľúčom z mapy a odoslanie na kolektor viď `export_flow()`2.7.

2.7 Export záznamu na kolektor, funkcia export_flow()

Funkcia uloží všetky potrebné parametre pre NetFlow v5 do pomocnej štruktúry. Parametre sú vkladané pomocou funkcie "htnol()"[4] - ak sa jedná o 32-bajtové parametre, alebo pomocou "htnos()"[4] pre 16-bajtové parametre.

Následne dôjde k vytvoreniu spojovaného UDP socketu. Pri jeho odosielaní na adresu kolektora je buffer naplnený vytvorenou štruktúrou dat pre NetFlow v5. Po odoslaní socketu dôjde k uzavretiu UDP spojenia.

2.8 Export celej cashe na kolektor, funkcia export_flow()

Funkcia prechádza celú flow cash až dokým nedôjde k jej vyprádzneniu. Jednotlivé záznamy posiela na kolektor pomocou funkcie `export_full_flows_cache()`, viď 2.8 a následne zmaže danú položku z cashe.

3 Návod na použitie

3.1 Kompilácia programu

Pre kompiláciu je potrebné zadať príkaz `make`. Po úspešnej kompilácii dôjde k vytvoreniu spustiteľného súboru `flow`.

Aplikácia je písaná v jazyku C++ a požaduje std=C++17.

3.2 Spúšťanie programu

Aplikáciu je možné spustiť pomocou príkazu `./flow [options]`

`[options]` sú voliteľné parametre:

- -f <file>
  Meno analyzovaného súboru v aktuálnom adresári.
  Bez zadaného parametra príjma súbor zo STDIN.
- -c <netflow_collector:port>
  IP adresa alebo hostname NetFlow kolektora. Voliteľne aj UDP port.
  Bez zadaného parametra je táto hodnota: 127.0.0.1:2055.
- -a <active_timer>
  Iterval v sekundách po ktorom sa exportujú aktívne záznamy na kolektor.
  Jedná sa o celé nezáporné číslo.
  Bez zadaného parametra je táto hodnota: 60.
- -i <inactive_timer>
  Iterval v sekundách po ktorom sa exportujú neaktívne záznamy na kolektor.
  Jedná sa o celé nezáporné číslo.
  Bez zadaného parametra je táto hodnota: 10.
- -m <count>
  Veľkosť flow cahce. Pri dosiahnutí max veľkosti dôjde k exportu najstaršieho záznamu v cashi na kolektor.
  Jedná sa o celé číslo väčšie ako 0.
  Bez zadaného parametra je táto hodnota: 1024.

4 Záver

Implementované je celé zadanie projektu. Používam NetFlow v5. Pri testovaní projektu som nezaznamenal žiadne problémy ani chyby implementácie.

Kompilácia aplikácie testovaná na servri Merlin.fit.vutbr.cz. Na spustenie nfcapd aplikácie nemáme sudo oprávnenia.

5 Použitá literatúra

[1] Cisco.com: NetFlow Export Datagram Format. [online], [vid. 2022-11-14]. Dostupné z: https://www.cisco.com/c/en/us/td/docs/net_mgmt/netflow_collection_engine/3-6/user/guide/format.html

[2] Cplusplus.com: Reference <cstdlib> atoi. [online], [vid. 2022-11-14]. Dostupné z: https://cplusplus.com/reference/cstdlib/atoi/

[3] Die.net: gethostbyname(3) - Linux man page. [online], [vid. 2022-11-14]. Dostupné z: https://linux.die.net/man/3/gethostbyname

[4] Die.net: htonl(3) - Linux man page. [online], [vid. 2022-11-14]. Dostupné z: https://linux.die.net/man/3/htonl

[5] WinPcap Documentation: pcap.h File Reference. [online], [vid. 2022-11-14]. Dostupné z: https://www.winpcap.org/docs/docs_412/html/funcs_2pcap_8h.html

[6] Kerrisk, M.: Man7.org. [online], [vid. 2022-11-14]. Dostupné z: https://man7.org/linux/man-pages/man3/getopt.3.html