Vysoké učení technické v Brně

Fakulta informačních technologií

Síťové aplikace a zpráva sítí

2022/2023

Dokumentace k projektu

Generování NetFlow dat ze zachycené síťové komunikace

Samuel Kuchta (xkucht11) 13.11.2021

1. Úvod

Cílem tohoto projektu bylo implementovat Netflow exportér, který bude přijímat data ve formátu .pcap, tvořit NetFlow, a odesílat ho na server/kolektor.

1.1 Popis problematiky

pcap soubory: jsou vstupem programu, a je třeba umět z nich získávat potřebné informace pro tvorbu NetFlow struktur.

Netflow v5 struktura: skládá se z hlavičky, a záznamu. Odesílá se na kolektor.

Časové informace: je potřeba získat různé časy jak pro hlavičku, tak záznamy. Časy se získávají jak z dat z paketů, tak i pomocí informací, které si vytváří samotný běžící program.

Převod na network big endian: na sítí se používá jiné pořadí MSB než na většině architektur. Proto je potřeba použít funkce jako jsou ntohs(), htons() a podobně.

Struktura cache paměti: Je potřebné uchovávat NetFlow záznamy ve strukturované formě, aby jsme v nich věděli hledat, upravovat je, a přidávat nové záznamy.

Odesíláni na udp server: je potřeba navázat spojení s kolektorem, aby jsme mu mohli odesílat záznamy.

1.2. Nefunkčné časti (dle Nfdump):

Nesprávny výpis IN BYTE (příliš velké čísla)

Nesprávny výpis času

Chyby ve flow sequence

1.3 Použití

Nejprve je nutné projekt zkompilovat pomocí příkazu make, který zavolá přiložený Makefile. Další doplňující informace jsou obsaženy v souboru README.

2. Implementace

2.1 Základní údaje

K implementaci jsem použil knihovnu „libpcap“. Jedná se o open source knižnicu jazyka C, která poskytuje API pro zachytávání paketů z dátových linek, nebo pro jejich čtení ze souboru. Funguje na všech Unixových systémech.

Při implementaci byly použité informace a příklady, které jsou dostupné na e-learning stránkách předmětu ISA (např. udp_echo_klient2), a informace z přednášek.

2.2 Struktura projektu

+ flow.cpp

+ lib.hpp

+ lib.cpp

+ readme.cpp

2.3 Tok Programu

1. Parsování argumentů

2. Otevření pcap soketu pro čtení ze souboru/stdin

3. Cyklické čtení jednotlivých paketů, kde se pro každý paket volá funkce flow_analyze(), která naplní netflow struktury příslušnými daty z paketu. A to tak, že buď vytvoří nový záznam, nebo aktualizuje hodnoty v již existujícím.

4. Záznamy jsou ukládány do cache paměti ve formě pole ukazatelů na netflow struktury.

5. v případě uplynutí času (active/ inactive) se určité struktury exportují na kolektor, a vymažou se z cache paměti. To stejné platí při maximálním naplnění cache paměti, nebo při přečtení posledního paketu.

6. samotné odesílání netflow struktur je prováděno pomocí udp soketu, který odesílá na kolektor vždy jednu flow hlavičku a jeden záznam (24B+48B)

Celý program je rozdělen na deset hlavních funkcí, které jsou popsány níže.

main()
parsování argumentů, alokace cache paměti, inicializace pcap, volá netFlow_analyze.

netFlow_analyze();
Volaný pomocí funkce pcap_loop().

Získá informace z hlaviček paketů, naplní NetFlow struktury, volá níže popsané funkce a odešle data na kolektor.

check_full();
Kontroluje, jestli není cache pamět NetFlow záznamů plná. Jestli ano, tak odešle všechny záznamy, a paměť vyprázdní.

export_timeout();
Kntroluje, jestli nedošlo k uplynutí času v active(TCP) a inactive(TCP) časovači. Ještě kontroluje jestli TCP spojeni nebylo ukončeno.

check_key();
Kontroluje, jestli se jedná o nový, nebo už inicializovaný záznam.

add_new_flow();
Přidá nový záznam.

update_key();
Aktualizuje záznam.

delete_flow();
Smaže záznam.

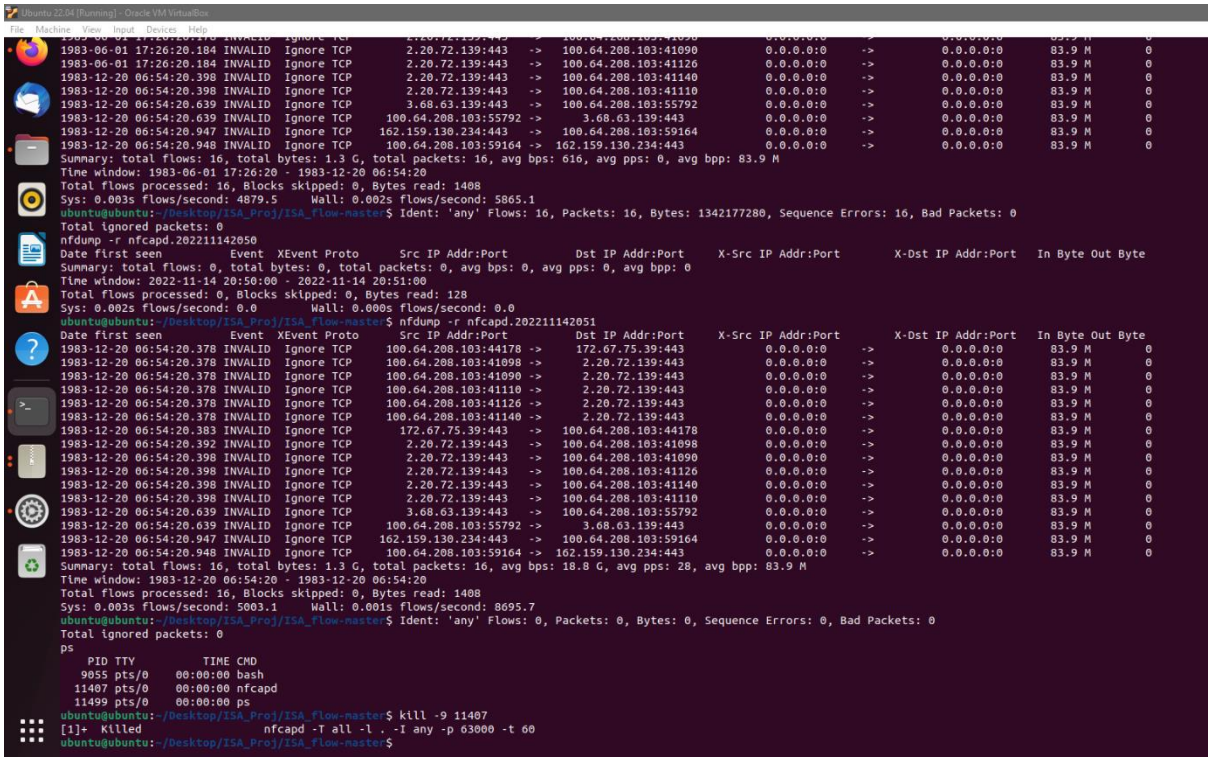
check_remaining();
Zkontroluje, jestli se v cache paměti nenachází nějaké záznamy.

send_netflow_to_collector();
Odešle záznam na kolektor.

3. Testování programu

Program jsem testoval na Virtuálním stroji s OS Ubuntu 22.4 metodou porovnávání výstupu s ověřeným zdrojem.

Pro testování funkčnosti jsem používal programy nfcapd a nfdump(viz. obrázek).



4. Závěr

V rámci projektu jsme si vyzkoušel implementovat v jazyce C++ exportér NetFlow záznamů verze 5 na kolektor.

Projekt mi zabral až 4 dny, protože jsem prvně nepochopil zadání, tak jsem musel předělávat půlku kódu (místo odesílání struktury na kolektor, jsem vypisoval textová data hodnot struktur na stdout. Takže jsem vlastně dělal navíc i kolektor..)

Projekt mě naučil, že je skutečně důležité prvně pořádně pochopit zadání, a udělat si návrh, než začneme implementovat kód.

5. Zdroje

Internetové zdroje z kterých jsem čerpal:

1. TCPDUMP/LIBPCAP public repository. *TCPDUMP/LIBPCAP public repository* [online]. Dostupné z: https://www.tcpdump.org/
2. pcap_compile(3) - Linux man page. *Linux Documentation* [online]. Dostupné z: https://linux.die.net/man/3/pcap_compile
3. pcap-filter(7) - Linux man page. *Linux Documentation* [online]. Dostupné z: https://linux.die.net/man/7/pcap-filter
4. inet_ntop(3) - Linux manual page. *Michael Kerrisk - man7.org* [online]. Dostupné z: http://man7.org/linux/man-pages/man3/inet_ntop.3.html
5. arpa_inet.h.0p - Linux manual page. *Michael Kerrisk - man7.org* [online]. Dostupné z: http://man7.org/linux/man-pages/man0/arpa_inet.h.0p.html
6. <netinet/in.h>. *The Open Group Publications Catalog* [online]. Copyright © 1997 The Open Group. Dostupné z: https://pubs.opengroup.org/onlinepubs/007908799/xns/netinetin.h.html
7. CARSTENS, T.: Programming with pcap. [online], rev. 25. október 2012, [vid. 2020-10-14]. Dostupné z: https://www.tcpdump.org/pcap.html
8. MATOUŠEK, P Monitorování toků NetFlow. [Univerzitní prednáška], 2020.
9. Zadání projektu. Dostupné z: https://www.vut.cz/studis/student.phtml?script_name=zadani_detail&apid=231021&zid=50009&armsgt=uwOo9yuc5C
10. https://www.cisco.com/c/en/us/td/docs/net_mgmt/netflow_collection_engine/3-6/user/guide/format.html
11. https://en.wikipedia.org/wiki/NetFlow