Vysoké učení technické v Brně

Fakulta informačních technologií



Síťové aplikace a správa sítí

2022/2023

Projekt

**Generování NetFlow dat ze zachycené síťové komunikace**

Dominik Pop (xpopdo00) Brno, 14. listopadu 2022

Obsah

[Úvod 3](#_Toc119267000)

[Základní informace 3](#_Toc119267001)

[Návrh programu 4](#_Toc119267002)

[Popis implementace 5](#_Toc119267003)

[Návod na použití 7](#_Toc119267005)

[Testování 8](#_Toc119267006)

[Zdroje 9](#_Toc119267007)

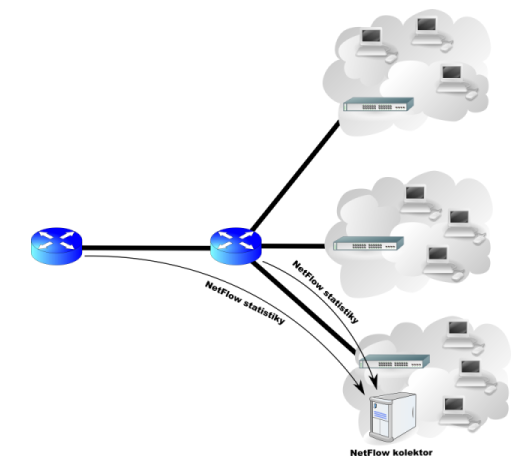
Úvod

Cílem projektu bylo navrhnout a implementovat NetFlow exportér. Jeho úkolem je analyzovat pakety a slučovat je do flows na základě jejich podobnosti. Tyto flows pak na základě zadaných vstupních parametrů odesílány na kolektor.

Základní informace

**Netflow [**[**1**](#z1)**]** je otevřený protokol od společnosti Cisco. Hlavním účelem je monitorování síťového provozu na základě IP toků a poskytovat tak podrobný pohled do provozu na síti. S pomocí NetFlow statistik lze odhalovat vnější i vnitřní incidenty, úzká místa v síti, dominantní zdroje provozu, efektivněji plánovat budoucí rozvoj sítě, sledovat, kdo komunikoval s kým, jak dlouho a s pomocí kterého protokolu.

Architektura NetFlow se skládá z exportéru a kolektoru. Exportér analyzuje příchozí pakety. Na základě IP toků generuje statistiky, které posílá na kolektor. Kolektor sbírá statistiky z exportérů a ukládá je do dlouhodobé databáze. Nad těmito daty pak může běžet nějaká aplikace, která může vizualizovat přehledy nasbíraných statistik uživateli.



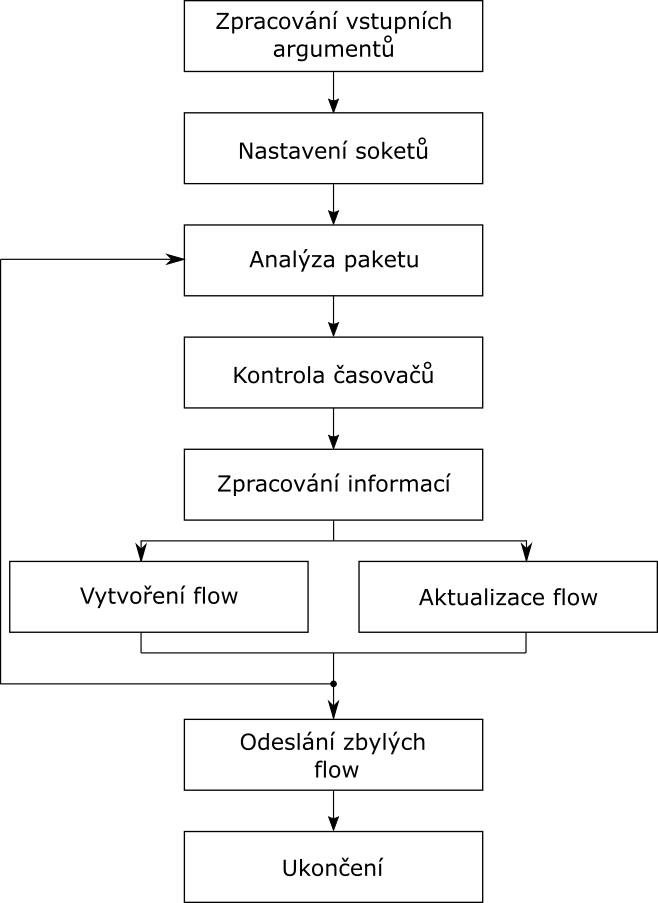
Obrázek : Typická architektura NetFlow

IP tok se skládá ze sekvence paketů se shodnou pěticí (sedmicí) údajů: zdrojová/cílová IP adresa, zdrojový/cílový port a typ protokolu. Každý tok o sobě nese informace, jako např. dobu vzniku, dobu trvání a další zobrazené [zde](#hlavička). [[2](#z2)]

NetFlow má několik verzí, nejvíce používanou je verze 5, ale v současnosti se začíná ve větším rozsahu využívat verze 9.

Návrh programu

K implementaci byl použit jazyk C a knihovna libpcap [[3](#z3)]. Struktura programu je rozdělena do několika zdrojových kódů a jednoho hlavičkového souboru. Program je navržen tak, aby analyzoval pakety v offline režimu buď z STDIN nebo souboru typu \*.pcap. Pakety se stejnými základními parametry slučuje do toků („flows“), které pak exportuje na kolektor. Pro jednoduchost je v jednom paketu exportována jeden tok.



Obrázek : Návrh programu

Popis implementace

Program se skládá z dvou hlavních funkcí a několika pomocných. Hlavní funkce jsou ***main***, ***callback*** a ***send\_flow*** popsané níže.

1. main

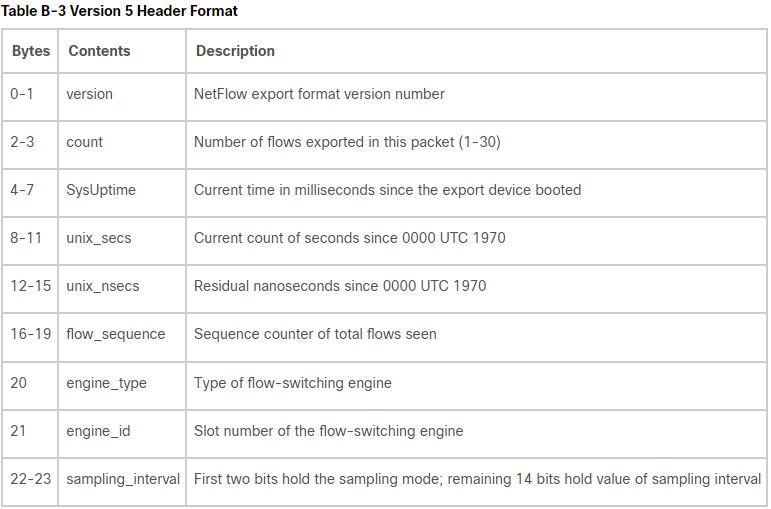
Funkce main se zabírá obstaráním všech potřebných komponent pro funkci programu. Nejdříve zpracováním vstupních argumentů pomocí funkce ***parse*\_*arguments*,** která využívá ***getopt*** a ukládá parametry do implementované struktury **t\_*Args***. Následně se main postará o založení listu ***t\_List*** pro flows, připravením soketů pomocí funkcí z knihovny ***libpcap***, nastavením filtru a následným voláním ***callback*** funkce pomocí knihovní funkce ***pcap*\_*loop***. Po skončení callback se provede export všech zbylých flows v listu, pomocí funkce ***send\_flow***, kdy čas odeslání je čas posledního příchozího paketu. V poslední části main probíhá patřičné uvolnění všech používaných komponent.

1. callback

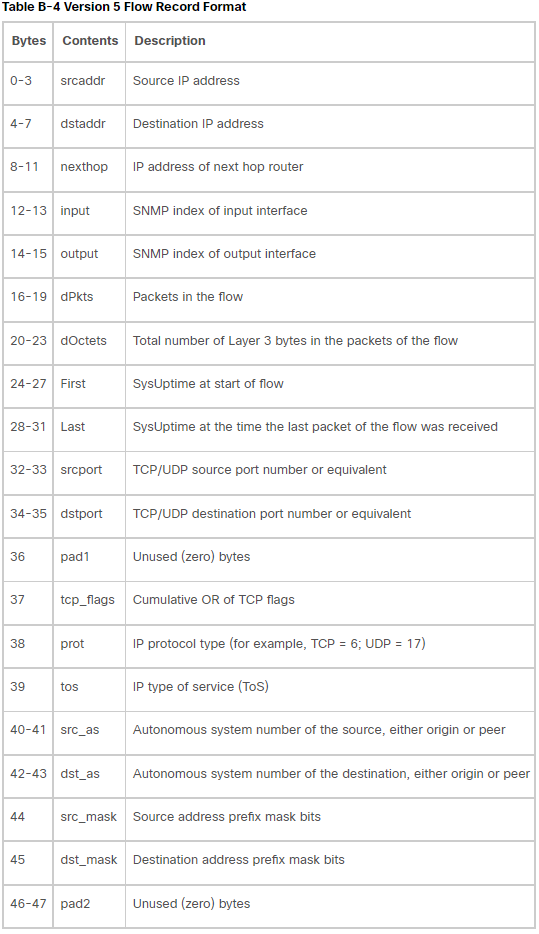
Funkce callback proběhne pro všechny přijaté pakety ze vstupu, které splňují podmínku filtru, tudíž pouze pakety s protokoly ICMP/TCP/UDP. Má za úkol analyzovat jednotlivé pakety. Vybírá z paketů informace a na základě těchto informací je sdružuje do toků („flows“). Pro uchování těchto informací slouží struktura ***t\_Flow***. První informace, kterou zpracovává je tzv. „epoch time“, jenž reprezentuje čas v sekundách, který uplynul od 1. 1. 1970 do doby příchodu paketu. Tento čas považujeme jako náš aktuální čas a je využíván k vypočítání intervalů Sysup. Sysup je interval v milisekundách, který uběhl od nabootování, ovšem v našem případě jakožto boot používáme příchod prvního paketu. Jestliže se jedná o první paket, pak je náš aktuální čas zároveň časem nabootování. Dále probíhá kontrola expirace časovačů již vytvořených toků pomocí funkce ***check\_timers***. Jestliže toku vypršel některý z časovačů je odeslán pomocí ***send\_flow***. Dále probíhá zjištění pěti informací, pomocí kterých identifikujeme jednotlivé toky. Jsou jimi zdrojová/cílová IP adresa, zdrojový/cílový port a typ protokolu. K těmto informacím zpracováváme ještě další, jako např. tos („type of service“) nebo tcp flags, které je zapotřebí zapisovat a někdy i aktualizovat k jednotlivým tokům. Po zpracování veškerých informací se buď vytváří nový tok anebo se aktualizuje již vytvořený tok s dříve zmíněnou pěticí. Pro vytvoření toku se používá funkce ***create\_flow*** a pro aktualizaci ***update\_flow***. Při vytváření toku je potřeba kontrolovat zaplnění cache-flow. Jestliže je maximálně zaplněna je před vytvořením nového toku nutno odeslat tok nejstarší, tudíž první tok v listu. U paketů, jež používají protokol TCP je zapotřebí kontrolovat, jestli se stav tcp flags nedostal do FIN/RST [[4](#z4)]. Jestliže ano je tok, do kterého daný paket spadá, exportován.

1. send\_flow

Funkce send\_flow se stará o odeslání toku na kolektor. V rámci tohoto řešení jeden paket odesílá pouze jeden tok. Funkce si nejdřív připraví proměnnou ***packet***, do které bude zapisovat informace o posílaném toku. Jako první se do paketu zapisuje hlavička daného toku, která obsahuje informace ukázané v [obrázku č. 3](#hlavička). Hlavička má celkovou velikost 24 B. Následně se zapisují informace o toku ukázané v [obrázku č. 4](#info). Jeden tok má velikost 48 B, celý paket má tedy velikost 72 B. Informace, které nejsme schopni z analyzovaných paketů zjistit, ale tok je vyžaduje (nexthop, pad1, …), mají připsanou hodnotu 0. Po zapsání veškerých informací je tok exportován a vymazán z listu.



Obrázek : Hlavička Flow



Obrázek : Data Flow

Návod na použití

Program byl implementován pro Unixová prostředí. K jeho přeložení je zapotřebí překladač gcc a nástroj GNU Make.

**Překlad programu** za pomocí Make:

$ make

**Překlad programu** bez Make:

$ gcc -std=gnu99 -Wall -o flow \*.c \*.h -lpcap

**Spuštění programu:**

$ ./flow [-f <file>] [-c <netflow\_collector>[:<port>]] [-a <active\_timer>] [-i <inactive\_timer>] [-m <count>] [-h]

Pokud spuštění předchozím příkladem není možné, je zapotřebí přidělit příkazu rootovská práva pomocí sudo (sudo ./flow …).

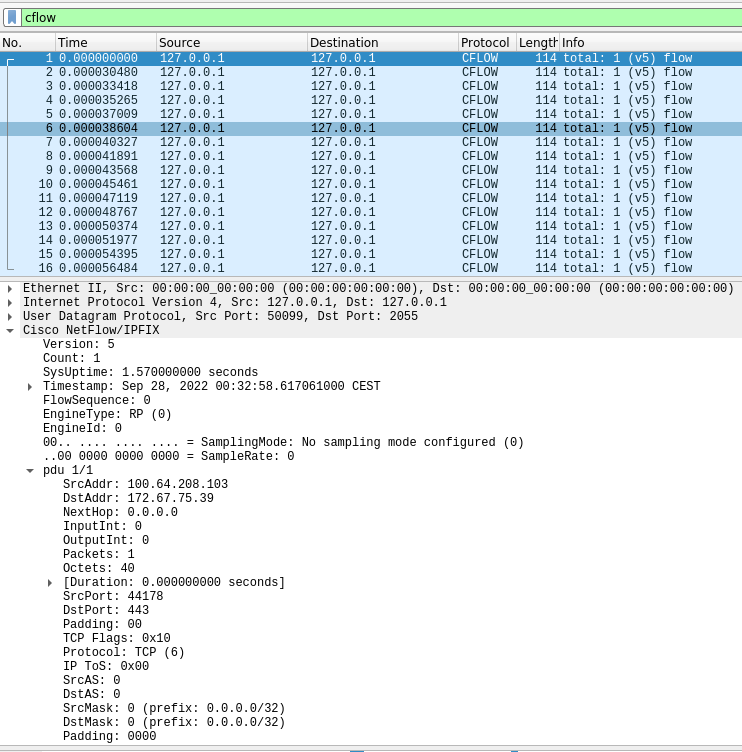
**Významy argumentů:**

* [-f <file>] – Jméno analyzovaného souboru nebo STDIN.
* [-c <netflow\_collector>[:<port>]] – IP adresa, nebo hostname NetFlow kolektoru. Volitelně i UDP port. Implicitně „127.0.0.1:2055“.
* [-a <active\_timer>] - Interval v sekundách, po kterém se exportují aktivní záznamy na kolektor. Implicitně 60 s.
* [-i <inactive\_timer>] – Interval v sekundách, po kterém se exportují aktivní záznamy na kolektor. Implicitně 10 s.
* [-m <count>] – Udává maximální velikost flow-cache. Při naplnění dojde k exportu nejstarší flow. Implicitně 1024.
* [-h] – Vypíše informace ohledně používání programu.

Testování

K testování projektu byly využity softwary Wireshark, Nfcapd [[5](#z5)], Nfdump [[6](#z6)], Softflowd [[7](#z7)]. Wireshark sloužil ke kontrole posílání toků. Dali se v něm zobrazit jednotlivé poslané toky a informace, které nesou.

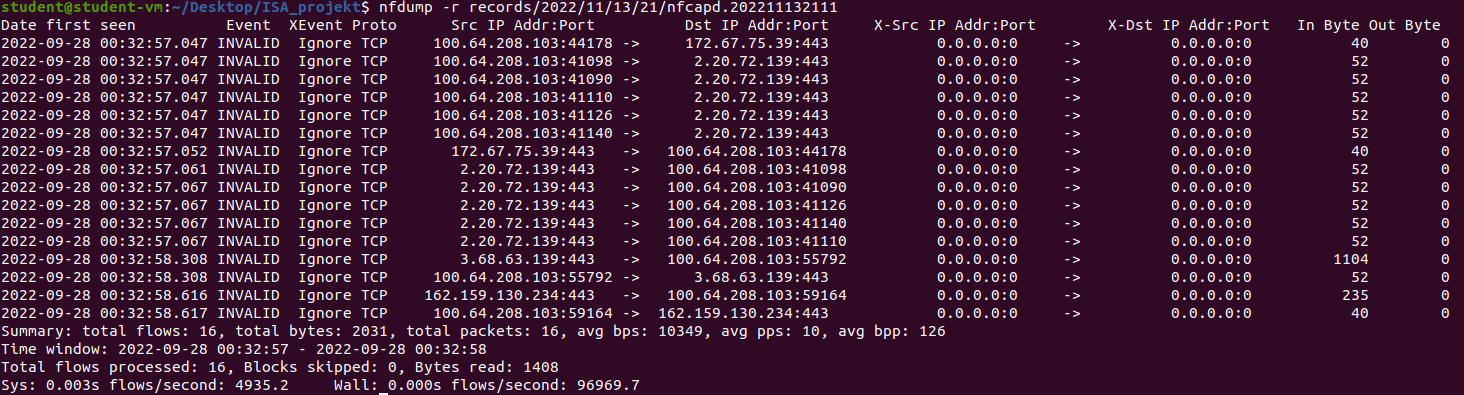
**$ ./flow <files/tcp.pcap**



Obrázek : Zobrazení odeslaných flows ve Wireshark

Pomocí Nfcapd bylo možné spustit kolektor toků, na který pak bylo možné dané toky posílat. Výstupy generoval do \*.nf souborů, které bylo možné zobrazit pomocí Nfdump.

**$ nfcapd -D -T all -l records -I any -S2 -p 2056 & ./flow -f files/tcp.pcap -c 0.0.0.0:2056**



Obrázek : Zobrazení statistik z kolektoru pomocí nfdump

Softflowd sloužil k poskytnutí referenčních výsledků.

Zdroje

* [1] NetFlow. *Wikipedia* [online]. 13 říjen 2022 [cit. 2022-11-13]. Dostupné z: <https://en.wikipedia.org/wiki/NetFlow>
* [2] NetFlow Export Datagram Format. *Cisco* [online]. 14. září 2007 [cit. 2022-11-13]. Dostupné z: <https://www.cisco.com/c/en/us/td/docs/net_mgmt/netflow_collection_engine/3-6/user/guide/format.html#wp1003394>
* [3] Manpage. *Tcpdump* [online]. [cit. 2022-11-13]. Dostupné z: <https://www.tcpdump.org/manpages/libpcap-1.5.3/>
* [4] TCP flags. *Pierky's Blog* [online]. [cit. 2022-11-13]. Dostupné z: <https://blog.pierky.com/netflow-weird-tcp-flags-in-flowviewer-and-flow-print/>
* [5] Nfcapd. *FreeBSD* [online]. 19. srpna 2005 [cit. 2022-11-13]. Dostupné z: <https://www.freebsd.org/cgi/man.cgi?query=nfcapd&apropos=0&sektion=1&manpath=FreeBSD+8.2-RELEASE+and+Ports&format=html>
* [6] Nfdump. *Ubuntu manuals* [online]. [cit. 2022-11-13]. Dostupné z: <https://manpages.ubuntu.com/manpages/xenial/man1/nfdump.1.html>
* [7] Softflowd. *Ubuntu manuals* [online]. [cit. 2022-11-13]. Dostupné z: <https://manpages.ubuntu.com/manpages/bionic/man8/softflowd.8.html>