Síťové aplikace a správa sítí

2022/2023

Dokumentace k projektu
Generování NetFlow dat ze zachycené síťové komunikace

Autor:

David Drtil (xdrtil03)

Brno, 11. listopadu 2022

Obsah

Úvod

Cílem bylo navrhnout a naimplementovat NetFlow exportér, který je schopný zpracovat zachycený síťový provoz ve formátu pcap, z nich vytvořit NetFlow záznamy a ty následně odeslat na kolektor. Tento proces se používá za účelem identifikování provozu v síti pro další analýzu jako je zjištění, zda je síťová infrastruktura dostatečně robustní, nebo také pro rozpoznání nekalých činností (útoků), které někdo může v síti páchat.

Detailní popis implementace

NetFlow analyzátor byl naimplementován v jazyce C s využitím síťových knihoven pcap a netinet a podporuje export pouze NetFlow v5 datagramů [1]. Pro definování NetFlows je zvolen jako klíč pětice (5-tuple) <protocol, src_ip, src_port, dst _ip, dst_port> podle Cisco ASA [2] na místo sedmice (7-tuple) <protocol, src_ip, src_port, dst _ip, dst_port, ip_tos, interface_ifIndex>.

Po spuštění programu dojde ve funkci `main()` k volání funkce `handle_sigint()` zachytávající systémové přerušení (např. Ctrl + C) pro bezpečné ukončení programu a uvolnění všech alokovaných zdrojů.

Následně probíhá načtení argumentů do struktury `args_t` a jejich kontrola ve funkci `parse_arguments()`, zda jsou syntakticky správné. Vytvoří se cache implementovaná pro svou jednoduchost dvousměrně vázaným seznamem. Otevře se udp socket pro odesílání exportovaných NetFlow na kolektor, otevře se pcap soubor a v hlavní smyčce programu, funkci `process_pcap_file()`, začne zpracování zachyceného síťového provozu. Brány v potaz jsou pouze pakety s protokolem TCP, UDP a ICMP, ostatní jsou odfiltrovány display filtrem pomocí funkce `pcap_setfilter()`

Původní časové značky paketů ze souboru jsou ukládány do 64bitového čísla. Důležitý je časový údaj prvního příchozího paketu, který se považuje za čas 0, a pomocí něj se poté počítá čas příchodu dalších paketů. Po načtení časového údaje je zkontrolováno, zda nešlo k překročení doby aktivního a neaktivního intervalu NetFlow v cache funkcí `check_timers()`.

Po načtení klíče Netflow z hlaviček příchozího paketu se rozhoduje, zda má být paket agregován do existujícího NetFlow nebo má být založen nový NetFlow. Při agregaci k existujícímu NetFlow jsou kontrolovány „tcp flags“, pokud je nastaven flag FIN nebo RST [3] je NetFlow exportován a nečeká se na vypršení neaktivního časovače.

Po zpracování posledního paketu z pcap souboru, dojde k ukončení hlavní smyčky programu, exportování všech zbývajících NetFlows z cache, uvolní se paměť všech alokovaných struktur, uzavře se socket a soubor zavoláním funkce `cleanup_on_exit()`.

Kontrolní výpisy o exportování NetFlows

Program je navržený tak, aby se pomocí logů vypisovaných na příkazovou řádku dalo zjistit, jak zpracování souboru a export NetFlow probíhalo. Tyto výpisy lze jednoduše vypnout pomocí nastavení preprocesoru `LOG_NETFLOWS_PROCESSING_ENABLED` na hodnotu false. V odevzdané verzi je výpis povolen.

Pomocná funkce

Funkce `get_readable_ipv4_address()` zpracovává v jednoduchém cyklu 32bitové číslo extrahované ze struktury `ip_header` a toto číslo načte ve správném formátu pro výpis do předaného pole.

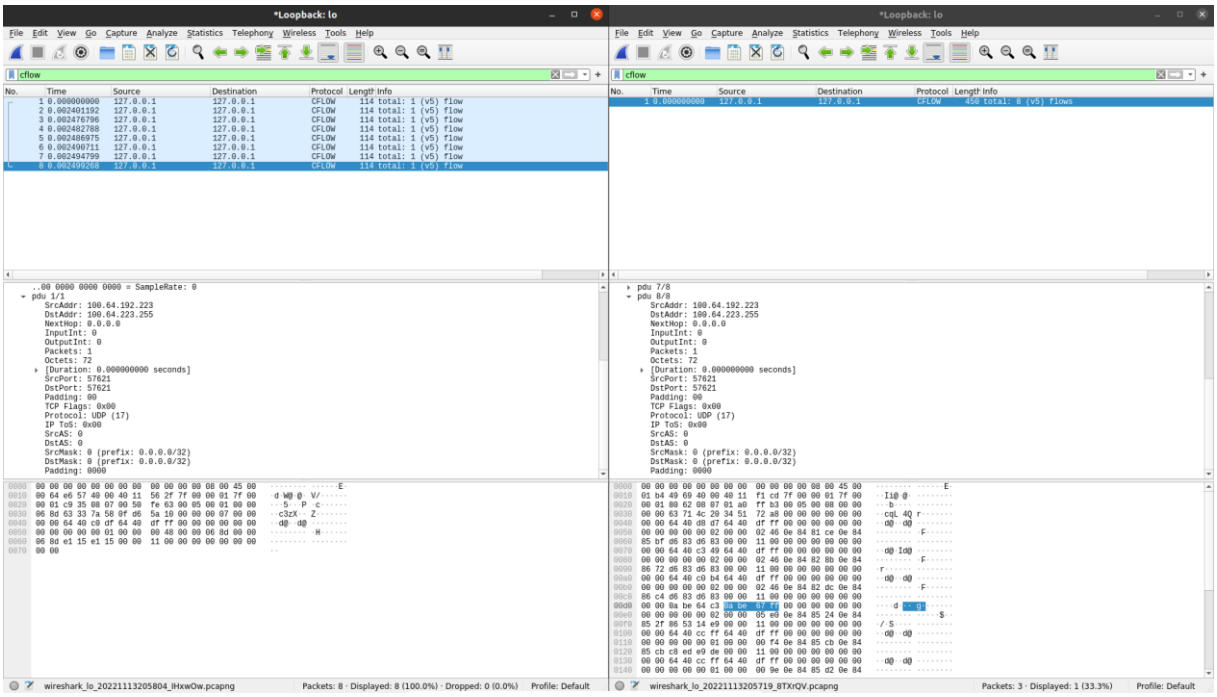
Ukázky použití exportéru a testování

Pomocí programu tcpdump jsem zachycoval síťovou komunikaci a vytvořil testovací soubory s formátem pcap (příponou .pcap).

Testovací metody:

1) Vlastní NetFlow exportér jsem spouštěl s testovacími soubory a posílal jsem vytvořené NetFlow datagramy verze 5 na adresu 0.0.0.0:2055, což je adresa pro zachytávání Cisco NetFlow [4]. Exportované NetFlow jsem zachytil aplikací Wireshark, která odposlouchávala na rozhraní „lo“ (loopback) a měla nastavený display filtr na „cflow“. Následně jsem zachytil v aplikaci Wireshark také referenční výstup z veřejně dostupného exportéru „softflowd“ a srovnal jsem výstupy s mými exportovanými NetFlow pro ověření správnosti.

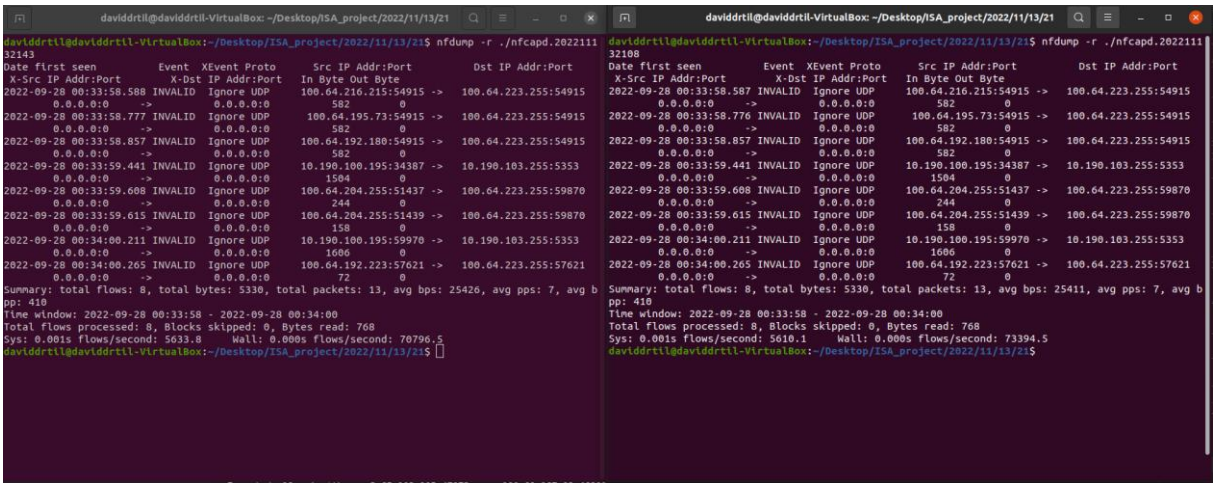
   Příkaz – „sudo softflowd -v 5 -n 0.0.0.0:2055 -r ./pcap_test_files/captured_traffic.pcap”



Obrázek č. 1 – Srovnání poslaných datagramů vlastního NetFlow exportéru s výstupem softflowd sw

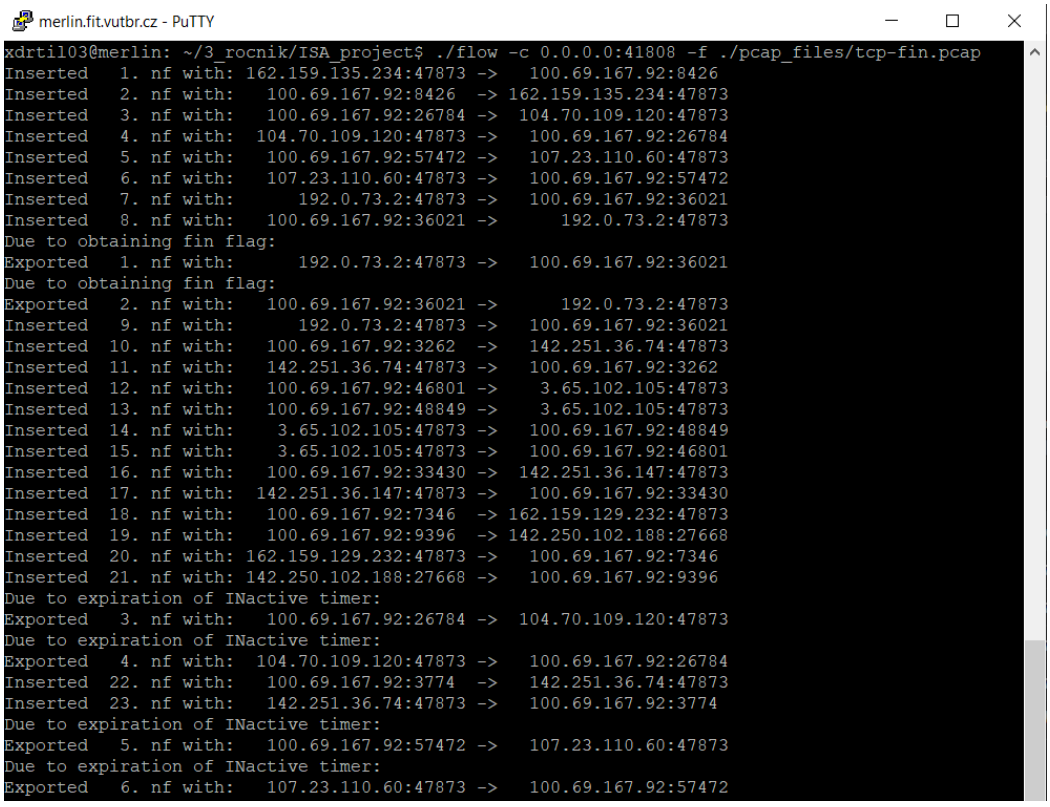
2) Pomocí programu „nfcapd“ jsem vytvořil kolektor neboli udp server běžící na zadaném portu. Na kolektor jsem posílal vlastní NetFlow datagramy a referenční NetFlow datagramy pomocí již zmiňovaného programu „softflowd“. Kolektor každých 5 minut vytvořil ze zachycených NetFlow datagramů soubor čitelný programem nfdump a vytvořit souhrny, které jsem následně pomocí programu diff porovnával.

   Příkaz – „nfcapd -D -T all -l . -I any -S 2 -p 20549 & sudo sudo softflowd -v 5 -n 0.0.0.0:20549 -r ./pcap_test_files/captured_traffic.pcap“

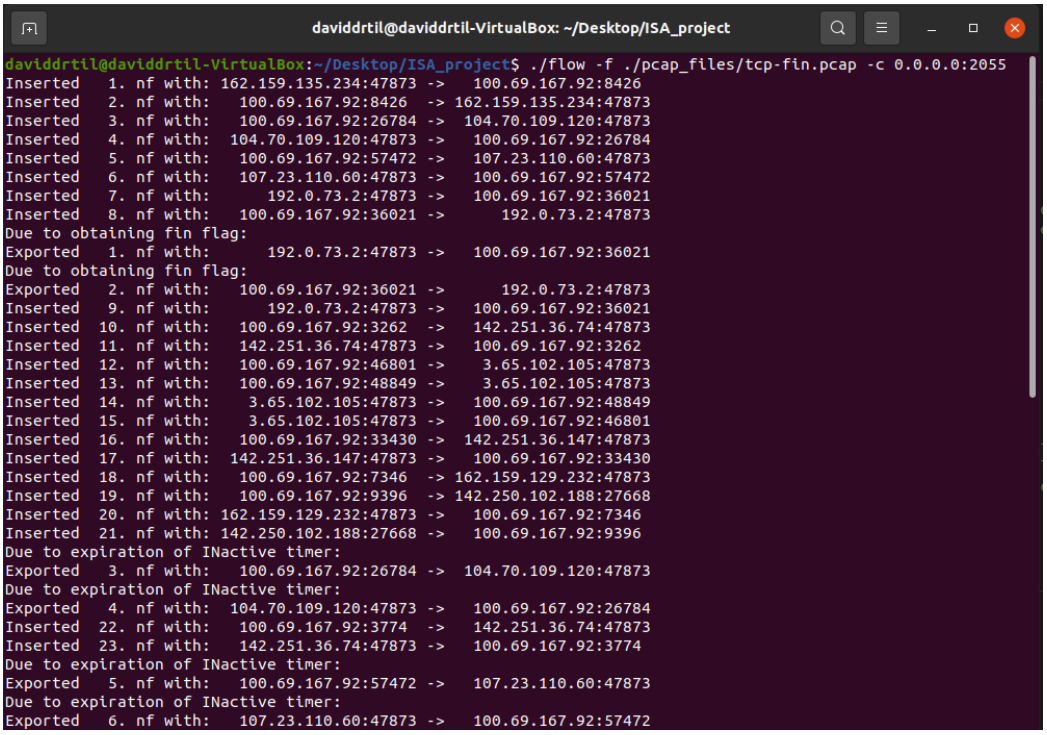


Obrázek č. 2 – Srovnání výstupů programu nfdump

3) Testování na školním serveru Merlin probíhalo tím způsobem, že jsem zde otestoval, zda je program možné přeložit a spustit. Následně jsem porovnal údaje z výpisu / logů činnosti NetFlow exportéru na stroji běžícím na OS Ubuntu 20.04. Zde jsem ale musel posílat datagramy na jednu adres, které byly již aktivní. K vyhledání takové adresy jsem použil příkaz „netstat -n --udp --listen“.



Obrázek č. 3 – Výpis zpracování NetFlows na školním serveru Merlin



Obrázek č. 4 – Výpis zpracování NetFlows na OS Ubuntu 20.04

Citace použitých zdrojů

[1] Cisco NetFlow export datagram formats [online]. Kalifornie (USA): Cisco Systems, 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

[2] NetFlow identification key [online]. Cisco community [cit. 2022-11-13]. Dostupné z: https://community.cisco.com/t5/security-knowledge-base/netflow-on-asa/ta-p/3119176

[3] TCP flags decoding [online]. Manito Networks [cit. 2022-11-13]. Dostupné z: https://www.manitonetworks.com/flow-management/2016/10/16/decoding-tcp-flags

[4] Cisco NetFlow default ports [online]. NSRC [cit. 2022-11-13]. Dostupné z: https://nsrc.org/workshops/2017/sanog29-cndo/networking/cndo/en/presentations/9.2_Netflow.pdf