Manuál k implementaci Netflow exportéru

Říjen 2022

Augustin Machyňák
xmachy02

Obsah

1 Úvod

Tento dokument slouží jako úvod do problematiky implementace netflow exportéru a popisu implementace pomocí knihovny *pcap*.

Netflow exportér slouží k exportování záznamů o příchozích a odchozích paketech na rozhraní v síti pomocí netflow protokolu na netflow kolektor, na kterém jsou tyto data agregována. Z těchto dat je následně možné detekovat například podezřelý provoz v síti - skenování sítě aj.

2 Návod na použití

Pro přeložení je možné použít přiložený Makefile. Tento Makefile slouží pouze pro spuštění cmake. Po zavolání příkazu *make* se provede vytvoření složky *build/*, spuštění cmake v této složce, přeložení aplikace a následně je spustitelný soubor přemístěn do kořenového adresáře. Aplikaci je následně možné spustit příkazem *./flow*. Je také možné vypsat použití pomocí flagu *-h* (*./flow -h*) nebo manuálu *flow.1* (*man -l flow.1*).

3 Implementace

Pro implementaci byla zvolena verze V5 Netflow protokolu a jazyk C++ (verze C++17). Všechny informace byly získané ze zdrojů uvedených v doporučené literatuře v zadání projektu (studijní materiály k předmětu ISA), případně jsou uvedené ve zdrojových souborech.

3.1 Podrobnosti

Jako první je vhodné si ujasnit některé zásadní podrobnosti, týkající se této implementace netflow exportéru, aby je nebylo nutné zmiňovat dále. Implementace:

- funguje "online" (vstup zpracovává sériově) - po přečtení každého paketu ze souboru/stdin je vyhodnoceno, jestli budou flow záznamy exportovány
- využívá **pouze** časové značky získané z pcap souboru - aktuální čas je roven času příchodu paketu; tedy čas příchodu prvního paketu je roven času spuštění systému
- nekontroluje IPv4 checksum

3.2 Struktury

Pro potřeby implementace bylo definováno několik struktur:

- Netflow V5 Header a Record
- Ethernet Header
- IPv4 a IPv6 Header
- TCP, UDP a ICMP Header

Netflow Header a Record je zřejmě použit při exportování. Všechny ostatní headery slouží ke zpracování dat přečtených z pcap souborů.

Už při psaní těchto struktur může vzniknout problém - totiž v C/C++ není zaručeno zarovnání. Některé překladače/architektury můžou z 1B udělat 4B po zarovnání, což může být problematické, pokud je cílem napsat přenositelný kód. Je možné použít direktivy jako *#pragma pack(1)*, avšak to není standardem. Nakonec bylo zvoleno použití *std::uint8_t* u všech hodnot s tím, že pokud je potřeba více, než 8b, tak je použito pole.

U Netflow V5 Header a Record není na zarovnání brán ohled. Pro exportování je dostupná metoda *Serialize()*, která serializuje příslušnou strukturu do pole *std::uint8_t* vhodného pro zaslání po síti.

3.3 Parsování dat z pcap souboru

Při přečtení každého z paketů je prvně paket přetypován na Ethernet header. Následuje kontrola "EtherType" pole v tomto headeru a pokud EtherType odpovídá IPv4 nebo IPv6 EtherType, tak je dále přetypován s tím, že ukazatel je posunut o velikost Ethernet headeru.

Pokud "protocol" pole v IPv4 nebo "next" pole v IPv6 headeru odpovídá TCP/UDP/ICMP číslu protokolu, tak je dále přetypován a uložen ukazatel stejným způsobem jako u IPv4/IPv6. Netflow V5 nepodporuje IPv6, takže zpracovávání takových paketů nemá žádný význam, což si autor uvědomil až později. Teoreticky je možné

tuto implementaci upravit takovým způsobem, aby podporovala Netflow V8, proto tato část byla ponechána pro případné rozšíření.

Jestliže číslo protokolu neodpovídá žádnému z podporovaných protokolů (které jsou specifikované v zadání), tak není zahozen. Vzhledem k tomu, že jsou pakety zpracovávány tímto způsobem, nebyl nalezen důvod k nezahrnutí těchto paketů do flow záznamů. Knihovna pcap totiž nabízí možnost filtrování paketů při čtení z pcap souborů, čehož nebylo využito.

3.4 Přidávání flow záznamů

Po naparsování (IPv4) paketu je provedeno jeho přiřazení do flow záznamu nebo vytvoření nového flow záznamu. Pokud již existuje záznam, který má stejnou:

- IPv4 zdroje
- IPv4 cíle
- číslo protokolu
- "Type of Service" ("DSCP")
- port zdroje
- port cíle

jako příchozí paket, tak jsou v tomto záznamu aktualizovány data (počet paketů, počet bajtů, čas příchodu posledního paketu a případně TCP flagy).

Pokud neexistuje takový záznam, tak je vytvořen nový.

3.5 Exportování záznamů

Při příchodu paketu je aktualizován čas a je tedy nutné zkontrolovat, jestli nevypršel jeden z časovačů ("inactive/active timer") v některém ze záznamů. Pokud některý z těchto časovačů vypršel, tak je nutné záznam exportovat. Mj. je také exportován nejdéle uložený záznam v případě, že je naplněna *"flow-cache"*.

Jako první se projde každý z uložených záznamů a ty, kterým vypršely časovače, jsou uloženy do pole pro exportování. Jakmile jsou všechny záznamy zkontrolovány, tak dojde k serializaci těchto záznamů a odeslání na netflow kolektor.

4 Testování

Vzhledem k povaze úlohy bylo testování prováděno manuálně. Ideální testování by využívalo implementace "dummy" netflow kolektor serveru. Avšak vzhledem k složitosti takového testování bylo využito pouze porovnání výsledků pomocí již existující implementace - **softflowd**.

Bylo vygenerováno několik pcap souborů z reálné komunikace s malým (cca 10), středním (cca 100) a velkým (cca 10000) množstvím paketů. Následně byly porovnány výsledky z agregovaných netflow záznamů zaslané na *nfcapd* pomocí této implementace netflow kolektoru a *softflowd*.

Výsledky byly podobné, avšak ne úplně stejné. Počet zaslaných paketů i bajtů jsou stejné; počet flow záznamů je však jiný - v referenční implementaci je jich méně. Důvodem je pravděpodobně fakt, že referenční implementace nebere v potaz časové údaje poskytované pcap souborem a zasílá všechny zachycené pakety najednou. To má za následek, že některé flow záznamy se vyskytnou ve finálním shrnutí pouze jednou místo toho, aby se vyskytly několikrát v různých časech (- po N sekundách jsou exportovány neaktivní záznamy).