

## Generování NetFlow dat ze zachycené síťové komunikace

# Obsah

<b>1</b>	<b>NetFlow</b>	<b>2</b>
<b>2</b>	<b>Implementácia v C++</b>	<b>2</b>
2.1	Popis implementácie . . . . .	2
2.2	Zaujímavé časti . . . . .	3
2.2.1	Kontrola TCP flagu . . . . .	3
2.2.2	Hľadanie najstaršieho flowu . . . . .	3
2.2.3	Určovanie času exportu . . . . .	3
2.2.4	NetFlow cache . . . . .	3
<b>3</b>	<b>Testovanie</b>	<b>4</b>
3.1	Popis spôsobu testovania . . . . .	4
3.2	Obrázky z testovania . . . . .	4

# 1 NetFlow

NetFlow je sieťový protokolový systém, ktorý zbiera IP pakety. Využitie tohto systému sa uplatňuje najmä pri správe siete, kontroly komunikácie a na výpomoc pri zapchatí siete.

NetFlow systém sa skladá z troch hlavných častí:

**Exportér:** Exportér vykonáva sprácu agregátora paketov do tzv. flowov/tokov. Tieto agregácie sa uskutočňujú na základe podobnosti parametrov komunikantov. Tými sú IP adresy, porty a typ servisu. Tieto toky sa exportujú na kolektor podľa daného hostnameu.

**Kolektor:** Kolektor je zodpovedný za príjem a uskladenie prijatých tokov, ktoré mu boli zaslané exportérom.

**Analyzáčná aplikácia:** Analyzuje prijaté dáta na kolektore a vyhodnocuje na základe toho stav sieťového toku. Detekuje zápchu v sieti.

Exportér pracuje ako agregátor paketov, ktoré boli zachytené v sieťovej komunikácii. Pakety sa agregujú podľa piatich parametrov a nimi sú: zdrojová a cieľová IP adresa, zdrojové a cieľové porty a typ servisu. Podľa týchto parametrov prichádzajúce pakety sú vkladané do NetFlow cache. Následne pomocou UDP protokolu sú toky postupne exportované na kolektor. Tieto toky pozostávajú z hlavičky a dát. Hlavička je špecifikovaná podľa štandardu NetFlow verzie 5. Dáta sú tvorené štruktúrou, ktorá obsahuje všetky informácie o paketovej komunikácii, ktorá mohla byť spolu agregovaná.

## 2 Implementácia v C++

### 2.1 Popis implementácie

Program začína parsovaním príkazovej riadky aby špecifikoval vlastnosti exportéru. Vlastnosťami exportéru je myslený čas aktívneho a neaktívneho exportu alebo veľkosť cache. Po upravení vlastností, sa otvára pcap handler, ktorý načíta buď zo súboru alebo zo stdinu pakety. Následne sa volá `pcap_loop()` funkcia, ktorá volá svoju callback funkciu `callbackFunc`. Táto funkcia aktualizuje momentálny čas prijatého paketu. Rovnako pri úplne prvom pakete si do štruktúry uloží počiatočný čas exportu. Po aktualizovaní časov, sa volá funkcia `flow`, ktorá má na starosti určovanie kedy je flow pripravený na export, vytváranie nových flowov a aktualizovanie už vytvorených.

Funkcia `flow` ako prvé vytvorí kľúč, ktorým je vektor o piatich hodnotách, podľa ktorých sa má agregovať. Tento kľúč sa bude používať v prehľadávaní dátovej štruktúry `map`, ktorá bude predstavovať flow cache. Po vytvorení kľúča momentálne spracovávaného paketu, sa skontroluje, či sa v cache nenachádzajú flowy, ktoré sú pripravené na export. Po preiterovaní celej cache exportuje flowy podľa aktívneho času alebo neaktívneho času. Následne po exporte týchto flowov, flowy odstráni z cache. Po exporte danom časom sa kontroluje či momentálny paket s jeho vlastnosťami už má miesto v cache, v prípade, že má miesto tak sa iba údaje aktualizujú. Údajmi sa myslí celková veľkosť flowu, čo predstavuje veľkosť všetkých paketov mínus požadované hlavičky. Vo všetkých prípadoch sa odpočítava ip hlavička a v prípade protokolu UDP a ICMP sa odpočítava konštanta 8B, ktorá predstavuje veľkosť týchto hlavičiek. Rovnako sa aktualizuje čas posledného prijatého paketu, čo bude predstavovať momentálny paket. Ak paket s určitými vlastnosťami nemá miesto v cache, pokúsi sa funkcia pridať nový do cache. Ako prvé sa skontroluje či nie je plná cache a ak je exportuje najstarší flow. Po exporte sa vytvorí nový flow záznam v dátovej štruktúre a podľa špecifikácií daného paketu sa definujú hodnoty. Všetky hodnoty treba previesť na správny bit order, ktorý sa zabezpečí pomocou

funkcie `htonl`(32 bitov) alebo `htons`(16 bitov). Osem bitové hodnoty netreba prevádzat. Po pridání nového paketu sa ide kontrolovať či náhodou paket nebol protokolu TCP a nemal práve aktívny FIN alebo RST flag. Ak toto všetko platí, flow sa ide exportovať.

Funkcia `exportFlow` vytvára špecifickú hlavičku pre exportovaný flow a agreguje ho dokopy s NetFlow dátami. Následne sa vytvorí klientsky soket a program sa pokúsi odoslať UDP paket na daný hostname.

Po načítaní všetkých paketov sa nakoniec kontroluje či je cache prázdna a ak nie je tak všetky zvyšné pakety exportuje.

## **2.2 Zaujímavé časti**

### **2.2.1 Kontrola TCP flagu**

Kontrola TCP flagu prebieha vo funkcii `TCP`, ktorá okrem toho, že parsuje z paketu zdrojový a cieľový port, kontroluje či náhodou `th_header` nezaznamenal FIN alebo RST výskyt. Ide o jednoduché bitové porovnanie s makrami `TH_FIN` a `TH_RST` z knižnice `tcp.h`. V prípade výskytu sa detekcia flagu posúva ďalej v kóde, kde potom následne dochádza k exportu flowu. Tento export prebieha na úplnom konci spracovania jedného paketu, kvôli tomu, že paket s danými flagmi je rovnako súčasťou flowu a jeho veľkosť a čas príchodu musí byť aktualizovaný v prípade, že sa už flow s rovnakými parametrami v cache nachádza. V inom prípade sa pridá nový flow a automaticky sa exportuje.

### **2.2.2 Hľadanie najstaršieho flowu**

Hľadanie najstaršieho flowu prebieha pomocou iterovania celej NetFlow cache a hľadania flowu, ktorý má najmenší čas posledného prijatého paketu. Za najstarší flow považujeme ten, s ktorým sa najdlhšie nič nedialo, a teda je to ten, ktorého čas posledného paketu je "najvzdialenejší".

### **2.2.3 Určovanie času exportu**

Pri príchode úplne prvého paketu sa do štruktúry `ts` (skrátene time settings) pridá initial time exportu. Tento čas slúži ako pomyselná nula/počiatok exportu. Aktuálny čas sa získava z hlavičky paketu. Tento aktuálny čas sa ukladá v `sec`, `ms` a `us`. Čas exportu je momentálny čas od ktorého sa odpočítava initial time. Do hlavičky exportovaného paketu sa rovno pridávajú hodnoty z `ts` štruktúry. Všetky tieto časy sú dátového typu `unsigned long` kvôli mikrosekundám.

### **2.2.4 NetFlow cache**

Netflow cache je dátového typu `map`. Pohodlne sa dá cez neho iterovať, vkladať a vymazávať flowy. Kľúč je zložený z piatich parametrov paketu. Tými sú IP adresy, porty a ToS. Kľúč je vektor, ktorý obsahuje hodnoty dátového typu `u_int32_t`. Táto hodnota bola zvolená na základe toho, aby sa do kľúča bitovo zmestili IP adresy, ostatné položky sa bitovo dorovnali na 32 bitov.

## 3 Testovanie

### 3.1 Popis spôsobu testovania

Testovanie prebiehalo pomocou kontroly výsledkov s funkciou `nfdump` a kontrolou vo Wiresharku, či sa flow pakety zaslali správne na požadovaný kolektor. Testovacie súbory boli vygenerované pomocou zachytávania komunikácie vo Wiresharku a následným exportom do súboru.

### 3.2 Obrázky z testovania

Vstup: `./flow -f udp.pcap`

```
Date first seen      Event  XEvent Proto  Src IP Addr:Port  Dst IP Addr:Port  X-Src IP Addr:Port  X-Dst IP Addr:Port  In Byte Out Byte
2022-09-28 00:33:58.777 INVALID Ignore UDP    100.64.195.73:54915 -> 100.64.223.255:54915  0.0.0.0:0 -> 0.0.0.0:0 526 0
2022-09-28 00:33:58.858 INVALID Ignore UDP    100.64.192.180:54915 -> 100.64.223.255:54915  0.0.0.0:0 -> 0.0.0.0:0 526 0
2022-09-28 00:34:00.211 INVALID Ignore UDP    10.190.100.195:59970 -> 10.190.103.255:5353  0.0.0.0:0 -> 0.0.0.0:0 1550 0
2022-09-28 00:33:59.441 INVALID Ignore UDP    10.190.100.195:34387 -> 10.190.103.255:5353  0.0.0.0:0 -> 0.0.0.0:0 1448 0
2022-09-28 00:33:58.588 INVALID Ignore UDP    100.64.216.215:54915 -> 100.64.223.255:54915  0.0.0.0:0 -> 0.0.0.0:0 526 0
2022-09-28 00:34:00.265 INVALID Ignore UDP    100.64.192.223:57621 -> 100.64.223.255:57621  0.0.0.0:0 -> 0.0.0.0:0 44 0
2022-09-28 00:33:59.609 INVALID Ignore UDP    100.64.204.255:51437 -> 100.64.223.255:59870  0.0.0.0:0 -> 0.0.0.0:0 216 0
2022-09-28 00:33:59.615 INVALID Ignore UDP    100.64.204.255:51439 -> 100.64.223.255:59870  0.0.0.0:0 -> 0.0.0.0:0 130 0
Summary: total flows: 8, total bytes: 4966, total packets: 13, avg bps: 23689, avg pps: 7, avg bpp: 382
Time window: 2022-09-28 00:33:58 - 2022-09-28 00:34:00
Total flows processed: 8, Blocks skipped: 0, Bytes read: 600
Sys: 0.002s flows/second: 3155.8 Wall: 0.000s flows/second: 22988.5
```

Obr. 1: Výsledok `nfdump -r udp.pcap`

udp.port==2055						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	127.0.0.1	127.0.0.1	CFLOW	114	total: 1 (v5) flow
2	0.000030579	127.0.0.1	127.0.0.1	CFLOW	114	total: 1 (v5) flow
3	0.000043350	127.0.0.1	127.0.0.1	CFLOW	114	total: 1 (v5) flow
4	0.000054656	127.0.0.1	127.0.0.1	CFLOW	114	total: 1 (v5) flow
5	0.000065578	127.0.0.1	127.0.0.1	CFLOW	114	total: 1 (v5) flow
6	0.000076669	127.0.0.1	127.0.0.1	CFLOW	114	total: 1 (v5) flow
7	0.000090206	127.0.0.1	127.0.0.1	CFLOW	114	total: 1 (v5) flow
8	0.000101100	127.0.0.1	127.0.0.1	CFLOW	114	total: 1 (v5) flow

Obr. 2: Wireshark po spustení príkazu

```

▼ Cisco NetFlow/IPFIX
  Version: 5
  Count: 1
  SysUptime: 1.677000000 seconds
  ▶ Timestamp: Sep 28, 2022 00:34:00.265706000 CEST
  FlowSequence: 2
  EngineType: RP (0)
  EngineId: 0
  00.. .... = SamplingMode: No sampling mode configured (0)
  ..00 0000 0000 0000 = SampleRate: 0

```

Obr. 3: Hlavička paketu

```

▼ pdu 1/1
  SrcAddr: 10.190.100.195
  DstAddr: 10.190.103.255
  NextHop: 0.0.0.0
  InputInt: 0
  OutputInt: 0
  Packets: 2
  Octets: 1550
  ▶ [Duration: 0.017000000 seconds]
  SrcPort: 59970
  DstPort: 5353
  Padding: 00
  TCP Flags: 0x00
  Protocol: UDP (17)
  IP ToS: 0x00
  SrcAS: 0
  DstAS: 0
  SrcMask: 0 (prefix: 0.0.0.0/32)
  DstMask: 0 (prefix: 0.0.0.0/32)
  Padding: 0000

```

Obr. 4: Rekord paketu