

Sieťové aplikácie a správa sietí
Generovanie NetFlow dát zo zachytenej sieťovej
komunikácie

Krištof Šiška (xsiska16)

November 2022

Contents

Úvod	3
Netflow v5	3
Návrh aplikácie	4
Implementácia	5
Implementačné detaily	5
Testovanie	6
Spustenie aplikácie	7
Zdroje	8

Úvod

Netflow je sieťový protokol vyvinutý spoločnosťou Cisco, ktorý zo zachytenej sieťovej komunikácie vytvára agregované flows. Tieto flows slúžia hlavne na analýzu sieťovej prevádzky, obsahujú dôležité informácie ako napríklad objem dát, počiatok a cieľovú destináciu. Je využívaný na monitorovanie siete, pričom pomáha k jej bezpečnosti a údržbe.

Netflow v5

V tomto projekte bol implementovaný netflow exportér, ktorý využíva netflow verziu 5. Každá flow odoslaná na kolektor sa skladá z netflow headeru a netflow recordu. Podoba týchto 2 častí je popísaná nasledujúcimi obrázkami.

Table B-3 Version 5 Header Format

Bytes	Contents	Description
0-1	version	NetFlow export format version number
2-3	count	Number of flows exported in this packet (1-30)
4-7	SysUptime	Current time in milliseconds since the export device booted
8-11	unix_secs	Current count of seconds since 0000 UTC 1970
12-15	unix_nsecs	Residual nanoseconds since 0000 UTC 1970
16-19	flow_sequence	Sequence counter of total flows seen
20	engine_type	Type of flow-switching engine
21	engine_id	Slot number of the flow-switching engine
22-23	sampling_interval	First two bits hold the sampling mode; remaining 14 bits hold value of sampling interval

Table B-4 Version 5 Flow Record Format

Bytes	Contents	Description
0-3	srcaddr	Source IP address
4-7	dstaddr	Destination IP address
8-11	nexthop	IP address of next hop router
12-13	input	SNMP index of input interface
14-15	output	SNMP index of output interface
16-19	dPkts	Packets in the flow
20-23	dOctets	Total number of Layer 3 bytes in the packets of the flow
24-27	First	SysUptime at start of flow
28-31	Last	SysUptime at the time the last packet of the flow was received
32-33	srcport	TCP/UDP source port number or equivalent
34-35	dstport	TCP/UDP destination port number or equivalent
36	pad1	Unused (zero) bytes
37	tcp_flags	Cumulative OR of TCP flags
38	prot	IP protocol type (for example, TCP = 6; UDP = 17)
39	tos	IP type of service (ToS)
40-41	src_as	Autonomous system number of the source, either origin or peer
42-43	dst_as	Autonomous system number of the destination, either origin or peer
44	src_mask	Source address prefix mask bits
45	dst_mask	Destination address prefix mask bits
46-47	pad2	Unused (zero) bytes

Návrh aplikácie

Aplikácia bude rozdelená na 3 časti. Prvá časť bude slúžiť na spracovanie argumentov. Táto časť bude najľahšia na implementáciu a zaberie najmenej času

na testovaní. Druhá časť bude riešiť vytváranie a správu flowov zo zachytenej komunikácie. Tretia časť sa bude zaoberať odosielaním flowov na kolektor. Táto tretia časť zaberie najdlhší čas na testovanie.

Implementácia

Projekt bol implementovaný v jazyku c++ kvôli veľkej výhode oproti jazyku C v tomto konkrétnom projekte a to je výskyt dátovej štruktúry, ktorá je založená na key : value pároch. V c++ sa nazýva Map a je súčasťou štandardných knižníc. Každý vytvorený flow je uložený do tejto mapy, pričom je identifikovaný kľúčom zloženým z piatich parametrov, ktoré jasne identifikujú každú flow. Tento kľúč je dátového typu tuple a má nasledujúcu podobu :

```
std::tuple<std::string, std::string, u_int16_t, u_int16_t, int>
```

Postupne zprava tieto argumenty vyjadrujú : IP adresu zdroja, IP adresu destinácie, port zdroja, port destinácie a číslo protokolu. Aplikácia podporuje iba protokoly ICMP (č. prot. 1), TCP (č. prot. 6) a UDP (č. prot. 17). Každým spracovaným paketom sa dejú nasledujúce operácie :

1. Prebehne kontrola timerov a veľkosti cache. Pokiaľ došlo k porušeniu jedného z timerov alebo k prekročeniu veľkostnému limitu, pristúpime k odoslaniu daného flowu (v prípade prekročenia veľkosti odosielame najstarší flow). Pred odoslaním je však nutné zmeniť odosielané dáta do network byte orderu. Na to nám slúžia funkcie htons a htonl.
2. Prebehne aktualizácia mapy. Pokiaľ sa už v mape nachádza flow, ktorý má identický kľúč s našim novo vytvoreným kľúčom z nového paketu, daná flow sa aktualizuje (zvýši sa počet paketov a bytov a zmení sa čas posledného paketu).
3. Prebehne kontrola TCP flagov. Pokiaľ spracovávaný paket má nastavený TCP FIN alebo TCP RST flag, automaticky bude exportovaný flow.

Samotná implementácia exportovania na kolektor je z časti inšpirovaná programom echo-udp-client2 od Petra Matouška.

Aplikácia po spustení skontroluje argumenty a pokúsi sa otvoriť súbor v pcap formáte (ak bol zadáný) na čítanie.

Implementačné detaily

Pri spracovaní ICMP paketov sú čísla zdrojového a cieľového portu 0, keďže ICMP je na 3. vrstve OSI modelu a porty sú využívané až vo štvrtej (transportnej) vrstve. Počet flowov v každom netflow package odoslaného na kolektor je 1, čo inak znamená že každá flow má svoj osobitný header. Je to spravené hlavne z dôvodu jednoduchosti implementácie a testovania. Parametre flowov, ktoré nám nie sú známe sú nastavené na 0.

Testovanie

Testovanie bolo prevedené pomocou `nfcapd` a následný výpis pomocou `nfdump` bol porovnávaný s paketmi zobrazenými cez program `Wireshark`.

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
2022-10-07 18:59:26.874	INVALID	Ignore	TCP	104.70.109.120:443 ->	100.69.167.92:41064	0.0.0.0 ->	0.0.0.0	2566	0
2022-10-07 18:59:26.861	INVALID	Ignore	TCP	100.69.167.92:41064 ->	104.70.109.120:443	0.0.0.0 ->	0.0.0.0	4796	0
2022-10-07 18:59:30.190	INVALID	Ignore	TCP	107.23.110.60:443 ->	100.69.167.92:32992	0.0.0.0 ->	0.0.0.0	52	0
2022-10-07 18:59:30.402	INVALID	Ignore	TCP	192.0.73.2:443 ->	100.69.167.92:46476	0.0.0.0 ->	0.0.0.0	223	0
2022-10-07 18:59:30.070	INVALID	Ignore	TCP	100.69.167.92:32992 ->	107.23.110.60:443	0.0.0.0 ->	0.0.0.0	52	0
2022-10-07 18:59:30.402	INVALID	Ignore	TCP	100.69.167.92:46476 ->	192.0.73.2:443	0.0.0.0 ->	0.0.0.0	120	0
2022-10-07 18:59:32.138	INVALID	Ignore	TCP	142.251.36.74:443 ->	100.69.167.92:48652	0.0.0.0 ->	0.0.0.0	52	0
2022-10-07 18:59:32.119	INVALID	Ignore	TCP	100.69.167.92:48652 ->	142.251.36.74:443	0.0.0.0 ->	0.0.0.0	52	0
2022-10-07 18:59:34.014	INVALID	Ignore	TCP	142.251.36.147:443 ->	100.69.167.92:38530	0.0.0.0 ->	0.0.0.0	329	0
2022-10-07 18:59:33.994	INVALID	Ignore	TCP	100.69.167.92:38530 ->	142.251.36.147:443	0.0.0.0 ->	0.0.0.0	812	0
2022-10-07 18:59:43.409	INVALID	Ignore	TCP	107.23.110.60:443 ->	100.69.167.92:32992	0.0.0.0 ->	0.0.0.0	233	0
2022-10-07 18:59:36.250	INVALID	Ignore	TCP	142.250.102.188:5228 ->	100.69.167.92:46116	0.0.0.0 ->	0.0.0.0	52	0
2022-10-07 18:59:44.755	INVALID	Ignore	TCP	142.251.36.147:443 ->	100.69.167.92:38530	0.0.0.0 ->	0.0.0.0	381	0
2022-10-07 18:59:38.288	INVALID	Ignore	TCP	142.251.36.74:443 ->	100.69.167.92:48654	0.0.0.0 ->	0.0.0.0	52	0
2022-10-07 18:59:36.234	INVALID	Ignore	TCP	162.159.129.232:443 ->	100.69.167.92:45596	0.0.0.0 ->	0.0.0.0	40	0
2022-10-07 18:59:25.561	INVALID	Ignore	TCP	162.159.135.234:443 ->	100.69.167.92:59936	0.0.0.0 ->	0.0.0.0	2107	0
2022-10-07 18:59:32.929	INVALID	Ignore	TCP	3.65.102.105:443 ->	100.69.167.92:53686	0.0.0.0 ->	0.0.0.0	372	0
2022-10-07 18:59:32.889	INVALID	Ignore	TCP	3.65.102.105:443 ->	100.69.167.92:53694	0.0.0.0 ->	0.0.0.0	320	0
2022-10-07 18:59:43.409	INVALID	Ignore	TCP	100.69.167.92:32992 ->	107.23.110.60:443	0.0.0.0 ->	0.0.0.0	168	0
2022-10-07 18:59:36.215	INVALID	Ignore	TCP	100.69.167.92:46116 ->	142.250.102.188:5228	0.0.0.0 ->	0.0.0.0	52	0
2022-10-07 18:59:44.742	INVALID	Ignore	TCP	100.69.167.92:38530 ->	142.251.36.147:443	0.0.0.0 ->	0.0.0.0	3699	0
2022-10-07 18:59:38.263	INVALID	Ignore	TCP	100.69.167.92:48654 ->	142.251.36.74:443	0.0.0.0 ->	0.0.0.0	52	0
2022-10-07 18:59:36.215	INVALID	Ignore	TCP	100.69.167.92:45596 ->	162.159.129.232:443	0.0.0.0 ->	0.0.0.0	40	0
2022-10-07 18:59:25.561	INVALID	Ignore	TCP	100.69.167.92:59936 ->	162.159.135.234:443	0.0.0.0 ->	0.0.0.0	400	0
2022-10-07 18:59:32.850	INVALID	Ignore	TCP	100.69.167.92:53686 ->	3.65.102.105:443	0.0.0.0 ->	0.0.0.0	424	0
2022-10-07 18:59:32.853	INVALID	Ignore	TCP	100.69.167.92:53694 ->	3.65.102.105:443	0.0.0.0 ->	0.0.0.0	424	0
Summary: total flows: 26, total bytes: 17870, total packets: 105, avg bps: 7394, avg pps: 5, avg bpp: 170									
Time window: 2022-10-07 18:59:25 - 2022-10-07 18:59:44									
Total flows processed: 26, Blocks skipped: 0, Bytes read: 2208									
Sys: 0.004s flows/second: 5658.3 Wall: 0.006s flows/second: 4061.2									

4 1.314073 104.70.109.120 100.69.167.92 TCP 74 443 -> 41064 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1

Wireshark · Packet 4 · tcp-fin.pcap

▼ Frame 4: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
Encapsulation type: Ethernet (1)
Arrival Time: Oct 7, 2022 18:59:26.875098000 Central Europe Daylight Time
[Time shift for this packet: 0.000000000 seconds]
Epoch Time: 1665161966.875098000 seconds
[Time delta from previous captured frame: 0.012385000 seconds]
[Time delta from previous displayed frame: 0.012385000 seconds]
[Time since reference or first frame: 1.314073000 seconds]
Frame Number: 4
Frame Length: 74 bytes (592 bits)
Capture Length: 74 bytes (592 bits)
[Frame is marked: False]
[Frame is ignored: False]
[Protocols in frame: eth:ethertype:ip:tcp]
[Coloring Rule Name: TCP SYN/FIN]

0000 dc fb 48 a0 ae ec 48 8f 5a 64 db 82 08 00 45 a4 --H--H- Zd---E-
0010 00 3c 00 00 40 00 36 06 62 b8 68 46 6d 78 64 45 -<--@-6- b-hFmxdE
0020 a7 5c 01 bb a0 68 21 9b c0 a8 e2 ce d1 7d a0 12 -\---h!- ----}-
0030 fe 88 30 f0 00 00 02 04 05 b4 04 02 08 0a ed 2e --0-----
0040 2e a6 8e fa 53 93 01 03 03 07S--- --

Spustenie aplikácie

Aplikácia je dodaná aj s makefilom. Prekladá sa príkazom `make` a spúšťa sa v nasledujúcej podobe :

```
./flow [-f <file>] [-c <netflow_collector>[:<port>]] [-a  
<active_timer>] [-i <inactive_timer>] [-m <count>]
```

Popis parametrov :

`-f <file>`

meno analyzovaného súboru alebo STDIN.

`-c <netflow_collector:port>`

IP adresa, alebo hostname NetFlow kolektoru. voliteľne aj UDP port (127.0.0.1:2055, ak není špecifikované),

`-a <active_timer>`

Interval v sekundách, po ktorom sa exportujú aktívne záznamy na kolektor (60, ak není špecifikované),

`-i <inactive_timer >`

Interval v sekundách, po jeho vypršaní sa exportujú neaktívne záznamy na kolektor (10, ak není špecifikované),

`-m <count>`

Veľkosť flow-cache. Pri dosiahnutí max. veľkosti dôjde k exportu najstaršieho záznamu v cachi na kolektor (1024, ak není špecifikované).

K projektu je dodaná aj manuálová stránka, ktorú je možné otvoriť nasledovným príkazom :

```
man -l flow.1
```

Zdroje

<https://en.wikipedia.org/wiki/NetFlow>
http://www.cisco.com/c/en/us/td/docs/net_mgmt/netflow_collection_engine/3-6/user/guide/format.html#wp1003394
<https://www.flowmon.com/en/solutions/network-and-cloud-operations/netflow-ipfix>
<https://www.kentik.com/kentipedia/what-is-netflow-overview/>
https://moodle.vut.cz/pluginfile.php/502893/mod_folder/content/0/udp/echo-udp-client2.c