Síťové aplikace a správa sítí (ISA)

2022/2023

Projekt - varianta 1:

Generování NetFlow dat ze zachycené síťové komunikace

Lucie Svobodová (xsvobo1x)

12. 11. 2022

Obsah

1 Úvod **3**

2 Teoretické podklady **4**
2.1 IP tok 4
2.2 NetFlow protokol 4
2.3 Kritéria pro export záznamů 5

3 Vlastní implementace **6**

4 Testování **8**

5 Bibliografie **10**

1 Úvod

Monitorování síťového provozu a přenosu dat je velmi důležitou činností při práci v oblasti počítačových sítích. Získané informace, například z programů nazývajících se sniffery paketů, můžeme dále využít k vytváření statistik, pomocí kterých můžeme dále odhalovat různé problémy vznikající na dané síti, nebo získávat informace, které se dají využít pro efektivní plánování dalšího rozvoje sítě. K získávání takových statistik můžeme využívat NetFlow architekturu, ve které NetFlow exportér analyzuje a sdružuje zachycené pakety podle jednotlivých IP toků, které v síti probíhají. Tyto toky odesílá na NetFlow kolektor, který je uchovává v databázi, díky čemuž je možné z těchto dat generovat tabulky a grafy, pomocí nichž můžeme provoz v síti převádět do jednoduše srozumitelných vizualizací a dále je analyzovat. Cílem projektu do předmětu Síťové aplikace a správa sítí bylo implementovat NetFlow exportér, který exportuje zachycená síťová data na kolektor.

V následující kapitole je uveden stručný přehled teoretických podkladů, které bylo třeba nastudovat před samotnou implementací NetFlow exportéru. Třetí kapitola je věnována návrhu a vlastní implementaci aplikace a poslední kapitola obsahuje pár slov k testování, které bylo při vývoji programu prováděno.

2 Teoretické podklady

NetFlow je protokol, který byl v roce 1996 vyvinutý společností Cisco Systems a který je využíván NetFlow architekturou k monitorování síťového provozu a sdružování jednotlivých paketů do IP toků. Informace, které poskytuje, slouží administrátorům počítačových sítí k tomu, aby měli přehled o provozu na jejich síti v reálném čase. Je možné díky němu plánovat budoucí rozvoj sítě, odhalovat podezřelý síťový provoz nebo účtovat ceny služeb v závislosti na množství přenesených dat. NetFlow architektura se skládá z NetFlow exportérů a NetFlow kolektoru. Exportér, většinou ve formě směrovače nebo pasivní NetFlow sondy, sdružuje jednotlivé pakety pomocí identifikace IP toků. Tyto toky sleduje a po dosažení určitých kritérií je exportuje na NetFlow kolektor. Ten je dále ukládá do databáze, kde nad nimi můžeme vytvářet statistiky potřebné ke sledování provozu na síti. [1, 2, 6, 11]

2.1 IP tok

Jak již bylo uvedeno výše, NetFlow exportér sdružuje jednotlivé pakety podle několika kritérií do IP toků, ang. flows. IP tok je identifikován sedmicí údajů: síťové rozhraní (interface), zdrojová IP adresa, cílová IP adresa, zdrojový port, cílový port, protokol a ToS (type of service). Abychom mohli zařadit více paketů do jednoho IP toku, musí v této sedmici obsahovat shodné údaje. Po vytvoření IP toku si o něm ukládáme i další informace - např. počet paketů v daném toku a součet jejich velikostí, dobu zachycení provního paketu ve flow, dobu zachycení posledního paketu, pro toky TCP protokolů uchováváme kumulativní OR TCP flagů. Vzhledem k tomu, že ICMP protokol neobsahuje informace o zdrojovém a cílovém portu, necháváme zdrojový port nulový a v cílovém portu uchováváme informaci o typu (type) a kódu (code) z ICMP paketu. [1, 7, 9, 12]

2.2 NetFlow protokol

NetFlow protokol je využíván ve více verzích, pro tento projekt byla použita verze 5 (NetFlow v5). Ta v hlavičce obsahuje především informaci o počtu toků exportovaných v daném NetFlow paketu, aktuální čas a sekvenční číslo, které je inkrementováno pro každý další exportovaný NetFlow paket. Celá hlavička má velikost 24 bytů. [11]

Tabulka 1: Formát hlavičky NetFlow v5 paketu.

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	sys_uptime	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	Sampling interval

Zdroj: Cisco Systems, Inc. (2017). Table B-3. [6]

Za hlavičkou paketu následují záznamy s jednotlivými toky. Jeden takový záznam má velikost 48 bytů a má pevně danou strukturu.

Tabulka 2: Formát NetFlow záznamu.

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

Zdroj: Cisco Systems, Inc. (2017). Table B-4. [6]

2.3 Kritéria pro export záznamů

Flow záznamy jsou uchovávány v paměti, v tzv. flow-cache. Vzhledem k tomu, že je chceme dále analyzovat a využívat k vytváření různých statistik, je potřeba je exportovat na kolektor. Velikost flow-cache není neomezená a zároveň IP toky chceme zpracovávat v co nejkratším čase po tom, co jsou pakety přenášeny, abychom si mohli vytvářet co nejaktuálnější a nejpřesnější představy o aktuálním provozu na síti. Pro rozhodování, které toky již můžeme exportovat, používáme různá kritéria.

Jedním kritériem, které pro export záznamů používáme, je velikost flow-cache. V paměti není možné uchovávat neomezené množství dat a proto po dosažení určitého počtu záznamů v cache začneme exportovat nejstarší záznamy, které by nastavenou velikost flow-cache překročily. Dalším kritériem je vypršení některého z časovačů. Pokud do IP toku již nějakou dobu nepřibývají žádné další pakety, je možné daný tok exportovat. V tomto případě se jedná o vypršení inaktivního časovače, jehož hodnota může být ve výchozím nastavení stanovena například na 10 sekund. Pokud do IP toku stále přibývají další a další pakety, musí být postaráno o to, aby byl daný tok také někdy exportován. Pokud bychom stále čekali, než vyprší inaktivní časovač a nebudou tedy do daného toku přibývat další záznamy, mohli bychom ignorovat například útok aktuálně probíhající na síti. Při exportu IP toků je tedy brán ohled i na délku doby, po kterou do daného toku další pakety přibývají. Pokud je aktivní časovač nastaven na 60 sekund, aktivní IP tok je vždy po 60 sekundách exportován a je založen tok nový. Posledním kritériem pro export IP toků je stav TCP spojení. Pokud je na síti zachycen TCP paket, který nese příznak FIN či RST, můžeme danou TCP komunikaci uzavřít a exportovat. Export všech záznamů probíhá na zadaný NetFlow kolektor, který data sbírá a ukládá do databáze. [3]

3 Vlastní implementace

NetFlow exportér `./flow` byl implementován v jazyce `C++` s použitím knihovny `libpcap` (`https://www.tcpdump.org/`). Tato konzolová aplikace zachytává vstupní síťovou komunikaci a sdružuje ji do IP toků, přičemž tyto toky dále exportuje na NetFlow kolektor. Aplikace poskytuje nápovědu, která uživatele informuje o možném použití přepínačů. Nápovědu je možné vypsat pomocí přepínače `-h`. Pro nastavení kritérií pro export NetFlow paketů je možné zadat následující parametry:

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

kde
`-f <file>` je jméno analyzovaného souboru nebo v případě nezadání parametru informace, že vstupní `.pcap` soubor bude zadán na standardní vstup
`-c <netflow_collector>:[<port>]` je NetFlow kolektor, na který jsou zasílány exportované datové toky - může být zadán jako IP adresa nebo hostname a případně doplněn o informaci o UDP portu, výchozí hodnota je `127.0.0.1:2055`
`-a <active_timer>` je aktivní časovač, výchozí hodnota je 60 sekund
`-i <inactive_timer>` je inaktivní časovač, výchozá hodnota je 10 sekund
`-m <count>` je velikost flow-cache, výchozí hodnota je 1024.

Význam kritérií pro export NetFlow paketů byl popsán v kapitole 2.3. Informace o zadaných přepínačích jsou v aplikaci ukládány do struktury `opts`. Vzhledem k tomu, že implementovaný exportér čte síťovou komunikaci z `.pcap` souboru, je použita funkce knihovny `libpcap` (`https://www.tcpdump.org/`) - `pcap_open_offline()`. Po úspěšném otevření zadaného souboru je inicializována proměnná `pcap_t *pcap`, která představuje pcap handler. Po aktivaci tohoto handleru je zjištěn typ hlavičky linkové vrstvy (pro jednotlivé typy viz [10]), přičemž jediný podporovaný typ touto aplikací je `LINKTYPE_ETHERNET`.

Program analyzuje a zpracovává pouze pakety protokolu IPv4, konkrétně TCP, UDP a ICMP protokolů. Z tohoto důvodu je pro filtrování paketů vytvořen filtr ve funkci `make_filter()`, přičemž tento filtr je zadán jako řetězec ve tvaru `ip and (udp or tcp or icmp)`. Tento řetězec je zpracován a zapnut funkcemi `pcap_compile()` a `pcap_setfilter()`. [13]

Pakety jsou následně zachytávány ve smyčce do té doby, než jsou zpracovány všechny pakety z daného vstupního `.pcap` souboru. Hlavní smyčku představuje knihovní funkce `pcap_loop()`. Zachycené pakety jsou postupně zpracovávány ve funkci `process_frame()`.

Funkce `process_frame()` získá z paketu timestamp a vytvoří záznam typu `Flowformat`. V tomto záznamu inicializuje všechny jeho položky na výchozí hodnoty (v případě, že jsou hodnoty zatím neznámé, jsou nastaveny na hodnotu 0) a doplní informace o času prvního zachyceného paketu (tedy aktuálně zpracovávaného). Do NetFlow záznamů se čas ukládá jako SysUptime - počet mikrosekund uplynulých od prvního zachyceného paketu. Tento čas je vypočítán vždy jako čas aktuálně zpracovávaného paketu minus čas prvního zachyceného paketu. K jeho výpočtu je využívána funkce `get_sysuptime()`, která pracuje s informacemi ze struktury `pcap_pkthdr *header` z položek `ts.tv_sec` a `ts.tv_usec`. Dále se po vytvoření záznamu podle seznamu etherTypes ([8]) zkontroluje, zda se jedná o protokol IPv4 a zpracování se přesouvá do funkce `process_ipv4()`. V programu je k získávání informací z hlaviček paketů využíváno přetypování typu `u_char *` na struktury knihovny `netinet`. Vždy je potřeba pomocí velikostí jednotlivých hlaviček určit, na jaké pozici v řetězci obsahujícím celý rámec se další datagram nachází, poté je možné řetězec přetypovat na strukturu hlavičky daného protokolu (např. `tcphdr`,

`ether_header`). Díky tomuto přetypování je následná práce s paketem jednodušší a přehlednější. [8]

Do funkce `process_ipv4()` je předán odkaz na nově vytvořený flow záznam. Z hlavičky IPv4 paketu je do flow záznamu přidána hodnota `dOctets`, která představuje počet bytů v daném paketu podle hodnoty `ip_len`. Dále jsou uloženy informace o zdrojové a cílové IP adrese, o protokolu a type of service. Podle protokolu síťové vrstvy je následně zavolána funkce pro zpracování TCP, UDP nebo ICMP paketu. [4]

Ve funkcích `process_tcp()` pro TCP a `process_udp()` pro UDP protokol jsou do flow záznamu přidány hodnoty zdrojového a cílového portu a pro TCP protokol i hodnoty příznaků (TCP flags). Ve funkci `process_icmp()` je pro zdrojový port ponechána hodnota 0 a cílový port je spočten jako $icmp->type * 256 + icmp->code$. Po uložení všech informací z aktuálně zpracovávaného paketu do daného záznamu o IP toku je volána funkce `record_flow()`. [1, 5, 7, 9, 15]

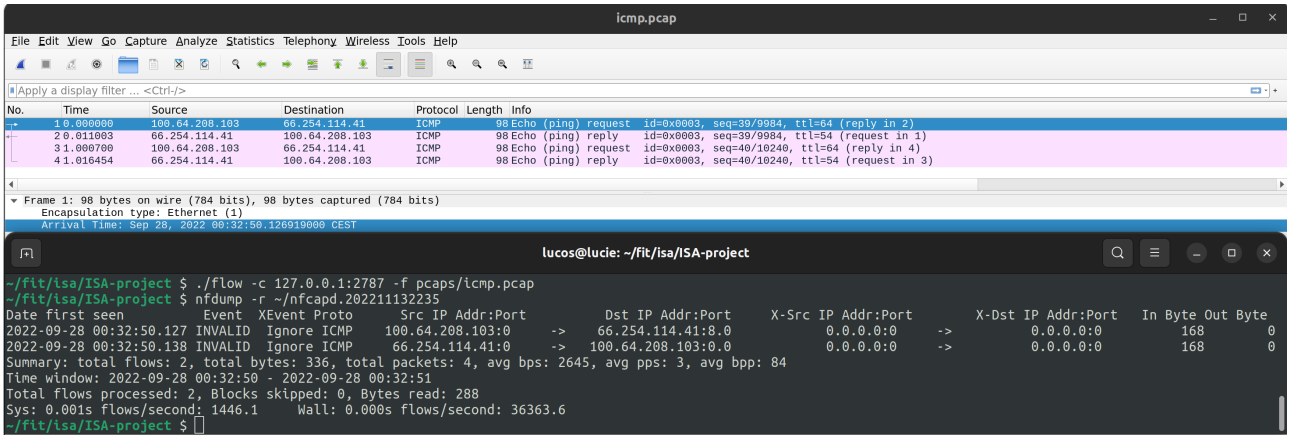
Funkce `record_flow()` zajišťuje kontrolu kritérií pro export uložených IP toků a následně pro přidání aktuálního záznamu do cache. V případě, že velikost flow-cache je nastavena na hodnotu 0, je daný záznam rovnou exportován na kolektor a funkce je ukončena. V jiném případě je provedena kontrola FIN a RST příznaků. Pokud je některý z těchto TCP příznaků v daném záznamu nastaven na hodnotu 1, jedná se ukončení TCP spojení. Pokud již ve flow-cache existuje IP tok, do kterého aktuálně zpracovávaný záznam patří, je tento tok aktualizován a následně ihned exportován a odstraněn z flow-cache. Aktuální záznam je rovnou exportován i v případě, že daný IP tok ve flow-cache neexistuje. Následně je provedena kontrola vypršení časovačů ve funkci `check_timers()`. V této funkci je procházena celá flow-cache a v případě, že pro některé toky vypršel aktivní nebo inaktivní časovač, jsou tyto toky exportovány (v pořadí od nejstaršího). Každý tok, který je exportován, je ihned z flow-cache odstraněn. Po kontrole všech časovačů je potřeba aktuálně zpracovávaný záznam uložit. Pokud ve flow-cache již je uložený IP tok odpovídající aktuálnímu záznamu, je tento IP tok aktualizován - je k němu připočtena délka nového paketu, počet paketů v daném toku je inkrementován, je proveden kumulativní OR TCP příznaků a čas posledního paketu z daného toku je nastaven na čas aktuálně zpracovávaného paketu. Pokud však daný IP tok ve flow-cache ještě není, je potřeba před jeho uložením zkontrolovat, zda není naplněna flow-cache a v případě, že je, je nejstarší záznam nejdříve exportován a až poté je nový tok uložen. Tímto končí celý proces zpracování jednoho paketu. Po zpracování celého `.pcap` souboru musí být exportovány všechny záznamy, které ve flow-cache zůstaly uložené, a to opět v pořadí od nejstaršího (nejdříve jsou exportovány záznamy, které mají nejmenší hodnotu v položce `flowrecord->first`). [3, 15]

K DNS rezoluci v případě zadání exportéru jako hostname je použita funkce `gethostbyname()`. Dále je vytvořen klientský socket a funkcí `connect()` je spojen se serverem. Samotné odesílání NetFlow paketů je zajištěno funkcí `send()`.

V projektu bylo ošetřena detekce signálu `SIGINT`, která je realizována pomocí funkcí knihovny `csignal` (`https://en.cppreference.com/w/cpp/header/csignal`). Při zachycení signálu `SIGINT` je zavolána funkce `handle_signal()`. Ta nastaví globální proměnnou `sigint_received` na hodnotu 1, díky čemu je možné ukončit případně právě prováděný cyklus. Zároveň je zavolána funkce `pcap_breakloop()`, díky které je ihned ukončeno případně probíhající zachytávání paketů. Po zachycení signálu jsou uvolněny všechny alokované zdroje a program je ukončen s návratovým kódem číslo 1. V případě, že v jakékoli části programu nastane chyba, je na chybový výstup vypsána chybová hláška, jsou uvolněny alokované zdroje a program je ukončen.

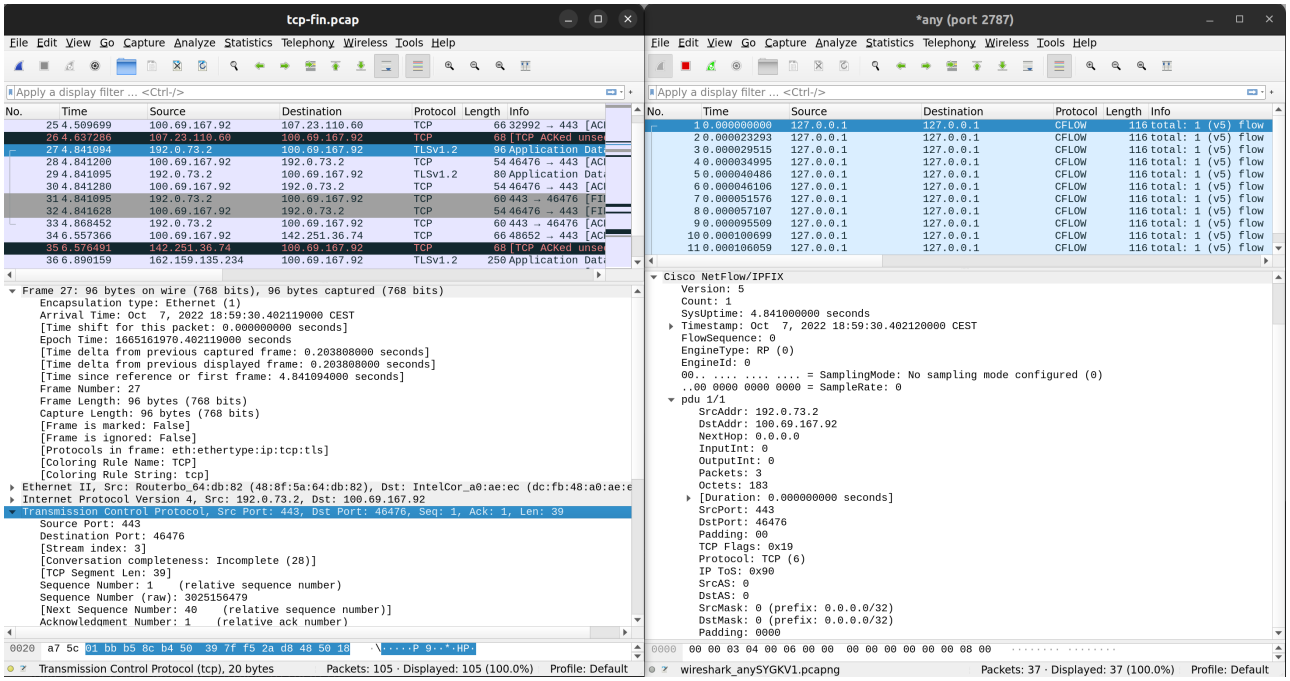
4 Testování

Testování implementovaného Netflow exportéru bylo prováděno na zařízení s operačním systémem Linux (Ubuntu 22.04 LTS). Exportované toky byly zachytávány pomocí nástroje nfcapd (netflow capture deamon, `https://nfdump.sourceforge.net/`)) a následně vypsány pomocí nástroje nfdump (`https://nfdump.sourceforge.net/`)), případně zobrazeny programem Wireshark (`https://www.wireshark.org/`) pro kontrolu více informací. Soubory formátu `.pcap` byly vytvářeny jak pomocí nástroje tcpdump (`https://www.tcpdump.org/`), tak pomocí monitorování běžného provozu na síti a jeho ukládání programem Wireshark. Generovány byly jak `.pcap` soubpry s malým počtem paketů, které byly následně ručně procházeny a bylo kontrolováno, zda počty paketů, jejich velikosti, časové značky a další informace souhlasí s exportovanými IP toky, tak i soubory obsahující velké množství paketů. Tyto soubory byly kontrolovány spíše namátkově, případně byly jednotlivé pakety v programu Wireshark filtrovány podle IP adres a portů. Zachycené IP toky byly následně podle těchto filtrů kontrolovány. Na následujících snímcích obrazovky jsou zobrazeny výstupy implementovaného NetFlow exportéru zobrazené programy nfdump a Wireshark při porovnávání s původní zachycenou komunikací.



Obrázek 1: Ukázka zachycených ICMP paketů (nahoře) a exportovaných IP toků vypsaných programem nfdump (dole).

Program byl dále testován pomocí debugovacích výpisů přidaných především pro kontrolu, zda jsou ve správný čas kontrolována kritéria pro export, například vypršení časovačů nebo velikost flow cache. Tyto výpisy byly následně ručně procházeny a porovnávány s výstupy programu Wireshark, přičemž případné nesrovnalosti byly postupně řešeny. Kromě správnosti výstupů NetFlow exportéru byl kladen důraz i na kontrolu práce s pamětí. Ta byla sledována nástrojem `valgrind` (https://valgrind.org/), aby bylo zajištěno správné použití a uvolňování všech alokovaných zdrojů. Všechny detekované chyby byly v průběhu implementace úspěšně odstraněny.



Obrázek 2: Ukázka zachycených paketů programem Wireshark (vlevo) a z nich vytvořených a exportovaných IP toků (vlevo). Vlevo je označený první paket z IP toku, který je zobrazený na pravé části obrazovky.

5 Bibliografie

[1] Wikipedia contributors. *NetFlow*. Wikipedia, The Free Encyclopedia [online]. Wikipedia, The Free Encyclopedia, Revidováno 13. 10. 2022. [cit. 2022-11-05]. Dostupné z: https://en.wikipedia.org/wiki/NetFlow.

[2] Cisco Systems contributors. *Configuring NetFlow and NetFlow Data Export*. Cisco Systems, Inc. [online]. Revidováno 10. 05. 2012. [cit. 2022-11-04]. Dostupné z: https://www.cisco.com/en/US/docs/ios-xml/ios/netflow/configuration/15-2s/cfg-nflow-data-expt.html#GUID-F17603FA-9205-460A-AE77-CB7F62EDF5C7.

[3] *Cisco NetFlow Configuration*. Cisco Systems, Inc. [online]. [cit. 2022-11-05]. Dostupné z: https://www.cisco.com/c/dam/en/us/td/docs/security/stealthwatch/netflow/Cisco_NetFlow_Configuration.pdf.

[4] GeeksForGeeks. *Introduction and IPv4 Datagram Header*. GeeksForGeeks [online]. Revidováno 28. 6. 2021 [cit. 2022-11-04]. Dostupné z: https://www.geeksforgeeks.org/introduction-and-ipv4-datagram-header/.

[5] Wikipedia contributors. *ICMP*. Wikipedia, The Free Encyclopedia [online]. Wikipedia, The Free Encyclopedia, Revidováno 13. 12. 2020. [cit. 2022-11-15]. Dostupné z: https://cs.wikipedia.org/wiki/ICMP.

[6] Cisco Systems contributors. *NetFlow Export Datagram Format*. Cisco Systems, Inc. [online]. Revidováno 14. 09. 2007. [cit. 2022-11-05]. Dostupné z: https://www.cisco.com/c/en/us/td/docs/net_mgmt/netflow_collection_engine/3-6/user/guide/format.html#wp1003394.

[7] GeeksForGeeks. *User Datagram Protocol (UDP)*. GeeksForGeeks [online]. Revidováno 26. 10. 2021 [cit. 2022-11-04]. Dostupné z: https://www.geeksforgeeks.org/user-datagram-protocol-udp/

[8] *IEEE 802 Numbers*. Internet Assigned Numbers Authority [online]. Revidováno 22. 2. 2022 [cit. 2022-11-05]. Dostupné z: https://www.iana.org/assignments/ieee-802-numbers/ieee-802-numbers.xhtml#ieee-802-numbers-1.

[9] *Internet Control Message Protocol (ICMP) Parameters*. Internet Assigned Numbers Authority [online]. Revidováno 25. 9. 2020 [cit. 2022-11-04]. Dostupné z: https://www.iana.org/assignments/icmp-parameters/icmp-parameters.xhtml.

[10] The Tcpdump Group. *LINK-LAYER HEADER TYPES*. The Tcpdump Group [online]. [cit. 2022-11-04]. Dostupné z: https://www.tcpdump.org/linktypes.html.

[11] *NetFlow Export Datagram Format - Cisco*. Cisco Systems, Inc. [online]. Revidováno 14. 09. 2007 [cit. 2022-10-29]. Dostupné z: https://www.cisco.com/c/en/us/td/docs/net_mgmt/netflow_collection_engine/3-6/user/guide/format.html.

[12] *Protocol Numbers*. Internet Assigned Numbers Authority [online]. Revidováno 7. 4. 2021 [cit. 2022-11-05]. Dostupné z: https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml.

[13] *PCAP(3PCAP) MAN PAGE*. The Tcpdump Group [online]. Revidováno 9. 9. 2020 [cit. 2022-11-05]. Dostupné z: https://www.tcpdump.org/manpages/pcap.3pcap.html.