Fakulta informačních technologií
Vysoké učení technické v Brně

Síťové aplikace a správa sítí
Offline NetFlow sonda

Petr Stehlík (xstehl14) 20. listopadu 2015

1 Zadání projektu

Cílem projektu je vytvoření offline NetFlow sondy, která analyzuje zachycenou síťovou komunikaci ve formátu PCAP. Při analýze jednotlivých paketů exportér vytváří jednotlivé flow podle zadaných kritérií a ty následně exportuje do NetFlow kolektoru pomocí UDP spojení.

NetFlow sonda je síťový prvek, většinou umístěný v routeru, který agreguje zachycenou síťovou komunikaci (pakety) do tzv. *flow*. Datové pakety jsou zařazeny do jednotlivých flow podle následujících kritérií

- `Protokol` – transportní protokol daného paketu
- `Src IP` – zdrojová IP adresa
- `Dst IP` – cílová IP adresa
- `Src port` – zdrojový port
- `Dst port` – cílový port

Pokud se tyto položky shodují, je paket přidán do odpovídajícího flow. V zadání jsou tyto kritéria rozšířeny o následující položky.

- `UDP paket` – všechny UDP pakety jsou okamžitě expirovány
- `TCP timeout` – pokud jsou dva stejné TCP pakety časově vzdáleny o určitou hodnotu, vytváří se nové flow
- `TCP RST a FIN flag` – TCP paket obsahující RST nebo FIN flag expiruje flow

Expirací flow rozumíme, že už žádné další pakety do sebe flow nemůže agregovat a expirované flow je připraveno na odeslání do kolektoru.

2 Návrh aplikace

2.1 Analýza

Projekt je vhodné rozdělit na několik jednotlivých celků, které mezi sebou spolupracují. Rozdělení na parsování paketů a tvorba flow je zřejmě nejdůležitější, následně export flow na kolektor a jako poslední část je vhodné implementovat zpracování základních požadavků.

2.2 Logické rozvržení

Projekt je rozdělen celkem na 4 části.

1. `Analýza paketu` – vyčtení paketu z PCAP a jeho parsování na jednotlivé struktury.
2. `Tvorba flow` – zpracování jednotlivých paketů do flow, vyčtení požadovaných informací z paketu do flow.
3. `Odesílání na kolektor` – serializace jednotlivých flow do jednoho paketu (po 30 flow) a odeslání na kolektor.
4. `Zpracování základních požadavků` – parsování parametrů programu, kontrolní výpisy, statistiky.

2.3 Implementační prostředky

Projekt je napsán v jazyku C++ s použitím několika systémových knihoven. Nejdůležitější knihovnou je `pcap.h`, která z PCAP souboru tvoří struktury jednotlivých paketů, se kterými je možno pracovat dále.

Celý projekt je navržen a implementována objektově. Byla možnost tvořit projekt i v jazyce Python, ale po analýze knihovny Scapy jsem usoudil, že není vhodná pro tvorbu NetFlow exportéru (zejména špatná manipulace s PCAP soubory).

Dalším důvodem pro vypracování v jazyce C++ je rychlost zpracování PCAP souboru, kde Python je přibližně 5x pomalejší než C++.

3 Popis implementace

Běh programu můžeme rozdělit na celistvé části jednotlivých úkonů. Tyto části se částečně překrývají a reagují na současnou situaci.

3.1 Základní požadavky

Program po spuštění prvně zpracuje parametry a jejich argumenty a uloží si aktuální hodnotu `clock()` pro pozdější výpočet času běhu programu. Následuje inicializace potřebných proměnných a struktur potřebných pro zpracování PCAP souboru.

3.2 Zpracování PCAP souboru

Pro práci s PCAP souborem je využita systémová knihovna `pcap.h`, která daný soubor zpracuje po jednotlivých paketech. Jelikož pracujeme offline, otevíráme PCAP soubor pomocí funkce `pcap_open_offline()`.

Následně procházíme PCAP soubor po jednotlivých paketech a ty analyzujeme. První rozdělení probíhá podle verze IP, protože NetFlow v5 neumí pracovat s IPv6 pakety. IPv6 pakety tedy při analýze přeskočíme.

Poté jednotlivé IPv4 pakety rozdělujeme dle protokolu. Program rozlišuje mezi TCP, UDP a ICMP pakety, protože každý protokol vyžaduje mírně odlišné zpracování a obsahuje různé položky. Obecný postup je uložení paketu do flow a uložení reference na dané flow do seznamu pro pozdější použití.

Jednotlivé flow jsou implementovány jako instance třídy `Flow` s vlastními metodami.

3.3 Zpracování paketů

3.3.1 UDP a ICMP pakety

Oba protokoly mají stejné chování a analýzu paketu. Paket je okamžitě expirován a ukládán jako nové flow. Program nevyhledává v existujících paketech zda existuje flow s potřebnými parametry a okamžitě vytváří nové flow.

3.3.2 TCP pakety

Zpracování TCP paketu je rozdílné z hlediska tvorby flow. Nad TCP paketem probíhá agregace podle zadaných parametrů, které jsou následující:

1. `základní pětice` – src IP, dst IP, src port, dst port, protokol
2. `timeout` – pokud jsou pakety od sebe časově vzdáleny více než parametr `--tcp-timeout`, vytvoří paket nové flow
3. `TCP flags` – pokud již flow obsahuje RST nebo FIN příznak, paket se ukládá do nového flow

Pokud najdeme flow, které splňuje všechny požadavky, aktualizujeme potřebné hodnoty flow (počet paketů, časové značky a TCP příznaky) a pokračujeme dalším paketem.

3.4 Flow cache

Jednotlivé pakety neporovnáváme se všemi flow, ale pouze s těmi, které jsou obsaženy ve flow cache. Její velikost je limitována parametrem `-m` (default 50). Pokud nalezneme nové flow, které se nevmístí do flow cache, nejstarší flow expirujeme (přesuneme do seznamu flow, které jsou připravené pro export na kolektor).

V případě analýzy velkého souboru se všechny expirovaného flow odesílají na kolektor po uplynutí zadané doby (parametr `-I`).

3.5 Odesílání expirovaných flow na kolektor

Pokud dokončíme analýzu všech paketů v souboru, případně narazíme na časový limit, exportujeme všechny expirované flow na kolektor. V případě dokončení analýzy všech paketů navíc expirujeme všechny flow ve flow cache.

Flow jsou posílány po 30 flow na kolektor (maximální počet dle NetFlow v5).

3.5.1 Tvorba paketu pro odeslání do NetFlow kolektoru

Serializací struktur jednotlivých flow do jednoho paketu je program připraven odeslat dané flow na kolektor. Těsně před odesláním navíc vytvoří potřebný header paketu, který je následně poslán pomocí UDP na kolektor.

V posledním kroku odesleme zbyvajici flow po odeslani vsech tricetiprvkových množin flow na kolektor.

3.6 Závěrečné úkony

V posledním kroce uvolníme alokované zdroje, uzavřeme UDP socket, PCAP soubor a program se korektně ukončí. V případě chyby během celého běhu programu vypíšeme na `STDERR` chybu a ukončíme program po uvolnění zdrojů.

4 Návod na použití

```
./isa_exporter [-i <file>] [-c <netflow_collector>[:<port>]] [-I <interval>]
[-m <count>] [-t <secs>]
```

`-i [--input]`	PCAP soubor pro analýzu (STDIN)
`-c [--collector]`	Adresa NetFlow kolektoru (127.0.0.1:2055)
`-I [--interval]`	Časový interval pro dobu na export do kolektoru (300)
`-m [--max-flows]`	Velikost flow cache (50)
`-t [--tcp-timeout]`	TCP timeout hodnota (300)
`-v [--verbose]`	vytvoření záznamu o činnosti do log.log
`-b [--body]`	počet bodů za projekt
`-h [--help]`	vypsání nápovědy

Všechny parametry jsou volitelné a mohou být jakkoliv kombinovány (krom přepínačů `-h` a `-b`)

4.1 Záznam činnosti

Jako volitelný parametr je v programu přidán `-v`, který vytvoří, nebo přepíše existující soubor log.log, ve kterém jsou obsaženy veškeré informace o celkové činnosti programu. Na konci souboru je vytvořena statistika o celkové době běhu programu, počtu zpracovaných paketů a vytvořených flow.

5 Použitá literatura

- 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
- man nfcapd
- man nfdump
- https://bto.bluecoat.com/packetguide/8.7/info/netflow5-records.htm