VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ
DEPARTMENT OF INFORMATION SYSTEMS

PCAP NETFLOW V5 EXPORTÉR

SEMESTRÁLNÍ PROJEKT
TERM PROJECT

AUTOR PRÁCE **ANASTASIIA MIRONOVA**
AUTHOR

VEDOUCÍ PRÁCE **Ing. KAMIL JEŘÁBEK, Ph.D.**
SUPERVISOR

BRNO 2024

Abstrakt

Tento projekt popisuje implementaci nástroje, který převádí soubory zachycené síťové komunikace ve formátu PCAP na protokol NetFlow v5. Program analyzuje síťové pakety a generuje jejich odpovídající NetFlow záznamy, které jsou následně odesílány do NetFlow kolektoru pro sběr a analýzu dat o síťovém provozu.

Abstract

This project describes the implementation of a tool that converts captured network traffic files in PCAP format into the NetFlow v5 protocol. The program analyzes network packets and generates corresponding NetFlow records, which are then sent to a NetFlow collector for network traffic monitoring and analysis.

Klíčová slova

Síť, Správa sítí, Sledování sítí, PCAP, NetFlow v5, Exportér, Programovací jazyk C++, Síťové programování, Zpracování paketů, C++ knihovny

Keywords

Network, Network Management, Network Monitoring, PCAP, NetFlow v5, Exporter, C++ Programming Language, Network Programming, Packet Processing, C++ Libraries.

Citace

MIRONOVA, Anastasiia. *PCAP NetFlow v5 exportér*. Brno, 2024. Semestrální projekt. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Kamil Jeřábek, Ph.D.

Obsah

Kapitola 1

Úvod

Projekt „PCAP NetFlow v5 exportér“ měl za cíl vytvořit program s názvem p2nprobe, který extrahuje informace o síťových tocích ze souboru PCAP a odesílá je kolektoru ve formátu NetFlow v5. Nástroj čte pakety ze souboru PCAP, který je zadán jako argument příkazového řádku, a agreguje je do jednotlivých toků. Tyto toky následně odesílá protokolem UDP do kolektoru NetFlow v5, kde jsou přijaty a dále zpracovány. Moje úloha spočívala v implementaci exportéru, přičemž jsem se zaměřil výhradně na export TCP toků, zatímco ostatní typy provozu byly ignorovány.

Kapitola 2

Postup řešení projektu

2.1 Úvod do programu

Tento program byl napsán v programovacím jazyce C++ s využitím knihovny libpcap pro zachytávání a analýzu síťových paketů.

2.2 Struktura aplikace

Aplikace byla rozdělena do několika hlavních fází:

1. Inicializace a zpracování parametrů zadaných uživatelem přes příkazový řádek.
2. Načítání a analýza paketových dat ve formátu PCAP a jejich organizování do jednotlivých toků.
3. Export zpracovaných toků do formátu NetFlow v5 pro následné využití v síťových analýzách.

Pro správu informací o toku byly použity specifické datové struktury, včetně dynamicky alokovaného pole, které uchovává jednotlivé toky.

2.3 Podrobný popis implementace

V rámci implementace byly vytvořeny veřejné třídy a metody, které řeší jednotlivé kroky zpracování dat. Program začíná získáním informací z příkazového řádku, pokračuje analýzou paketů a končí exportem toků v požadovaném formátu. K zajištění funkčnosti je použita knihovna libpcap pro práci s paketovými daty.

AppInfoFactory

Třída `AppInfoFactory` je zodpovědná za zpracování argumentů zadaných uživatelem při spuštění programu. Tato třída se zaměřuje na extrakci informací z příkazového řádku a jejich validaci. Po úspěšném zpracování argumentů vytváří instanci třídy `AppInfo`, která obsahuje všechny potřebné údaje pro další zpracování (například název souboru PCAP nebo IP adresu kolektoru). Pokud dojde k neplatnému argumentu, `AppInfoFactory` vyvolá výjimku.

AppInfo

Třída `AppInfo` uchovává všechny důležité informace o aplikaci, které byly extrahovány z příkazového řádku pomocí `AppInfoFactory`. Tato třída obsahuje parametry jako název PCAP souboru, informace o kolektoru, IP adresu, port a další nastavení potřebná pro provedení exportu. `AppInfo` slouží jako centralizovaný zdroj nastavení pro celou aplikaci.

FlowAggregatorProcessor

Třída `FlowAggregatorProcessor` je zodpovědná za agregaci paketových dat do jednotlivých toků. Využívá metody pro analýzu paketů a jejich organizování do struktur toku. Třída uchovává seznam toků a poskytuje metody pro přidávání nových toků, jejich aktualizaci a získání všech agregovaných toků. `FlowAggregatorProcessor` komunikuje s dalšími komponentami programu, jako je `PcapReader`, aby zpracovala přicházející pakety a přidala je do správného toku.

PcapReader

Třída `PcapReader` je zodpovědná za čtení souboru ve formátu PCAP a předávání paketů do strategie zpracování, která je definována v třídě `PacketProcessor`. Třída používá knihovnu `libpcap` pro otevření a analýzu souboru. Aplikuje filtr pro zachycení pouze TCP paketů a následně předává každý paket metodě `processPacket` v aktuálně nastavené strategii. `PcapReader` také zajišťuje správu životního cyklu souboru PCAP, včetně otevření a uzavření souboru.

NetflowSender

Třída `NetflowSender` je zodpovědná za odesílání zpracovaných toků ve formátu NetFlow v5 na kolektor. Používá informace obsažené v třídě `AppInfo` k nastavení připojení na správný kolektor (IP adresa a port). `NetflowSender` formátuje toky podle specifikace NetFlow v5 a odesílá je prostřednictvím UDP. Tento proces je navržen tak, aby byl efektivní a spolehlivý, s ohledem na časové intervaly pro odesílání dat.

PacketProcessor

Třída `PacketProcessor` je abstraktní základní třída pro zpracování jednotlivých paketů. Poskytuje rozhraní pro zpracování paketů a agregaci informací o toku. Různé implementace této třídy mohou specifikovat různé způsoby zpracování dat, například analýzu paketů podle protokolu nebo metod pro uložení informací do datových struktur.

Třída `PacketProcessor` je využívána třídou `FlowAggregatorProcessor`, která je konkrétní implementací zpracování paketů a související agregace dat.

Kapitola 3

Dokončená aplikace

3.1 Spuštění aplikace

./p2nprobe <host>:<port> <pcap_file_path> [-a <active_timeout> -i <inactive_timeout>]

Pořadí parametrů je libovolné. Popis parametrů:

- <pcap_file_path> - zde bude uvedena cesta k souboru PCAP, který se má zpracovat;
- <host> - IP adresa nebo doménové jméno kolektoru;
- <port> - port kolektoru, kam budou zprávy odesílány;
- -a <active_timout> - počet sekund pro nastavení aktivního timeoutu exportu flow (defaultní hodnota při nepoužití parametru 60);
- -i <inactive_timeout> - počet sekund pro nastavení inaktivního timoutu exportu flow (defaultní hodnota při nepoužití parametru 60).

3.2 Testování a výsledky testů

Testování funkčnosti programu bylo realizováno pomocí nástroje `nfcapd`, který slouží jako kolektor pro záznamy v protokolu NetFlow. Pro ověření správnosti exportovaných dat byl použit nástroj `softflowd`, jehož výstupy byly považovány za referenční, protože plní podobnou roli jako náš nástroj p2nprobe. Porovnání výstupů těchto dvou nástrojů bylo klíčové pro potvrzení, že p2nprobe správně implementuje formát NetFlow.

Během testování byly zaměřeny na porovnání několika základních parametrů, jako je celkový počet aktivních toků, objem přenesených dat a počet přenesených paketů. Testy prokázaly, že p2nprobe správně vykazuje hodnoty těchto parametrů, které jsou ve shodě s výsledky nástroje `softflowd`. Nicméně během analýzy byly zaznamenány malé časové odchylky mezi časovými značkami jednotlivých toků, které jsou způsobeny zaokrouhlovacími chybami při výpočtu časových údajů. Tyto odchylky byly velmi malé a jejich vliv na přesnost měření je zanedbatelný.

Pro usnadnění srovnání výstupů jsou výsledky testů uloženy v samostatných souborech. Výstupy generované nástrojem `softflowd` jsou k dispozici v souboru `results/softflowd.txt`, zatímco výstupy našeho programu p2nprobe najdete v souboru `results/p2nprobe.txt`. Oba soubory obsahují podrobné údaje o všech relevantních parametrech toků a slouží jako základ pro detailní porovnání výkonu obou nástrojů.

Kapitola 4

Bibliografie

- TCPdump and libpcap Homepage – Oficiální stránka nástroje TCPdump, který je široce používán pro zachytávání a analýzu síťových paketů.
- softflowd, a flow-based network monitor – GitHub repozitář nástroje `softflowd`, který umožňuje sběr a analýzu síťového toku.