VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta informačních technologií

ISA 22/23

Generování NetFlow dat ze zachycené síťové komunikace

PROJECT DOCUMENTATION

Vladyslav Tverdokhlib

(xtverd01)

Brno, 14.11.2022

TABLE OF CONTENTS

1 INTRODUCTION TO THE PROBLEMATIC

1.1 THE PROBLEMATIC

Traffic analysis is one of the activities to achieve the optimal network configuration. Thanks to analysis, network errors, vulnerabilities, loads and inappropriate or malicious communications can be detected. There are many services and methods for capturing and monitoring traffic. This project considers the *Cisco Netflow* method with a protocol.

1.2 SOLUTION

The *Netflow protocol* is designed to record network traffic in a specific format. Thanks to this, we get the opportunity to analyze traffic according to certain criteria. The protocol contains packets aggregated into so-called flows. The *flow* contains the number of packets in itself, source and destination addresses and ports, and so on.

An *exporter* is a packet interceptor, usually a L3 router. The main processing and aggregation is done here. Next, the exporter sends the records to the collector, where the analyzed information is stored.

2 APPLICATION DESIGN

2.1 PREREQUISITE

This application processes the captured traffic in *pcap* format, creates netflow packets and sends them to the collector. To simulate real-time packet capture, the packet *timestamps* are used. Thus, the application works discretely and only updates the time with each new packet.

2.2 CONCEPT

Reading packets from a file is carried out by a *sniffer*. Packets are read and processed one by one.

A new package is added to an existing flow or forms a new one.

The *flow* is characterized by the unique six-value key: *srcaddr, dstaddr, srcport, dstport, prot and tos*. The flows are stored in the *flow cache* and exported after some time.

The flow cache has an *active* and *inactive* timer, and a *cache size* counter. When updating the time, it is necessary to check the expiration of the timers for each flow. When the timer expires or the size is exceeded, the flows are exported to the *collector*.

3 DESCRIPTION OF IMPLEMENTATION

3.1 USED CODES

The program works with the ipv4 address family and sends UDP packets to the collector. The example *echo-udp-client2*[1] was taken as the basis for the UDP client.

A sniffer is implemented to read packets from a file, based on the example *sniff-filter.c*[2].

3.2 DEFINITION

The flow is represented as a *record_flow* structure that contains all the fields of Table B-4[3].

The flow cache is implemented using an ordered FIFO vector of the structures – *vector<record_flow>*.

Many variables are declared in the global scope, including flow cache, constants, counters, and backup variables.

3.3 MECHANISM

The main activity of the program takes place in the *mypcap_handler* function, which is called by the sniffer to process a captured packet.

The time is updated in the sysuptime variable using the formula:

$$(\text{packet}_{\text{sec}} - \text{boot}_{\text{sec}}) * 1000 + round(packet_{usec} - boot_{usec}) \,/\, 1000$$

After that, the execution of a large function begins which sequentially goes through the flow cache vector and updates it. The function for each record:

1. Checks if timers overrun. If yes, the outdated flow must be exported.
2. Then compares its six-value key with the packet's key. If the key matches, the packet is written to the flow by updating the *Last*, *dPkts*, *dOctets*, and *tcp_flags* parameters.

[1] (c) Petr Matousek, 2016, from Elearning
[2] (c) Petr Matousek, 2020, from Elearning
[3] www.cisco.com/c/en/us/td/docs/net_mgmt/netflow_collection_engine/3-6/user/guide/format.html#wp1003394

In case the packet doesn't have a suitable flow, a new record is added to the end of the vector.

Also, this function monitors the size of the cache and reacts to the FIN and RST tcp flags to do an instant export.

The function returns a *vector<record_flow> flow_export* with the records to be exported.

Export to the collector is carried out in the *send_netflow_packets* function. Since a netflow packet can contain a maximum of 30 entries, large sets are divided into multiple packets.

4 INSTRUCTIONS FOR USE

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

- -f <file> file name or STDIN for analysis.
- -c <neflow_collector:port> IP address or hostname of the NetFlow collector. Optionally also UDP port (127.0.0.1:2055, if not specified)
- -a <active_timer> - the interval in seconds after which active records are exported to the collector (60 if not specified)
- -i <seconds> - the interval in seconds after which inactive records are exported to the collector (10 if not specified)
- -m <count> - flow-cache size. When the maximum size is reached, the oldest record in the cache is exported to the collector (1024, if not specified).

All parameters are considered optional. If any of the parameters is not specified, the default value is used instead.

The application creates netflow records for TCP, UDP and ICMP protocols only.

Data becomes useful if the analyzer is involved in the process. It analyzes netflow data and generates human-readable reports (often in the form of graphs). Thus, the application allows, for example, an administrator to analyze the saved traffic.

Literature

[1] https://en.wikipedia.org/wiki/NetFlow

[2] http://www.cisco.com/c/en/us/td/docs/net_mgmt/netflow_collection_engine/3-6/user/guide/format.html#wp1003394

[3] https://www.tcpdump.org/manpages/

[4] https://www.geeksforgeeks.org/vector-in-cpp-stl/

[5] https://linux.die.net/man/

[6] https://man7.org/linux/man-pages/man3/getopt.3.html

[7] https://www.ibm.com/docs/en/zos/2.4.0?topic=hosts-network-byte-order