SUPINFO Academic Dept.

SUPINFO International University

INSTITUTE OF INFORMATION TECHNOLOGY

Group Projects
A.Sc.2 - Architecture

Contents

2017-2018

Version 1.1
Last update: 05/12/2017
Use: Students/Staff
Author: SAD

TABLE OF CONTENTS

1. Project Overview

As the R&D department of a famous network device manufacturer, you have been mandated to work on the future best seller: an IP Firewall. The operating system has not been elected yet so you don't have any limit but you know that you need to build an IP Firewall solution able to process and filter packets according to a set of rules and running in the userland.

You decide to write a generic demonstration of your know-how by presenting two cases. The first one to show how it works even without network access by reading a classical capture file and the second by actually filtering packets on a live system.

2. Functional Expression

2.1. Operation

Your solution must be able to process complete network datagrams (data including the whole stack, from Ethernet frame to IP packet). You decide to complete your work in two steps :

- Write a program that reads from a capture file (tcpdump, wireshark etc) that print out results on stdout.
- Write a program that plugs into an existing OS and make the decision of what to do with an incoming packet

The first step will let you create and test your filtering logic (rules, etc), while the second step will allow you to see real results with real network traffic. For the second part you'll need to leverage system API that allow to defer decisions on packets to the userland. You're free to choose and use whichever OS you want (Linux, Minix, etc.).

2.1.1. Rules

Your program must be able to parse rules written in a close-to-natural language that can express the following needs:

- Catch packets using conditions based on all significant protocols headers (mac addresses, IP ports, IP addresses, ICMP types, etc.).
- Do something with the packet : Accept, Drop.

You are free to use existing grammars and parsers.

2.2. Technical requirements

You are free to use whichever programming language you want and any helper library you find suitable. That is also valid to read the ".cap" files.

However, using an existing firewall solution such as netfilter is strictly forbidden.

3. Deliverables

Students should include the following elements in their final delivery:

- A zip archive with the project source code. The source code must also come with the build system used (Project file, autotools...), if any.
- Project documentation
  - Technical documentation explaining your choices and/or implementation choices/details on the following items (at least):
    - Parsing rules
    - Matching packets
    - Technical choices for filtering
  - User manual

The first document is an academic document. Address the reader as a teacher, not a client. This document can be in French or in English, at your option.

4. Graded Items

The project will be graded as follows, on a 250/275 scale:

- Documentation: 50 points
  - User documentation (25 points)
  - Technical documentation (25 points)
- Standalone operation: 100 points
  - The program can parse rules in a natural language. (25 points)
  - It's possible to create rules based on protocol headers. (25 points)
  - The program is able to read a capture file, apply rules, and tell the outcome. (50 points)
- Live operation: 100 points
  - The program can be plugged into an existing OS and work on real network traffic (100 points)
- Bonus: 25 points
  - Bonus features done by the students (25 points)