CS 6500 – Network Security
Assignment 5: Firewalls and Rule Matching
Dr. Manikantan Srinivasan
Even Sem. 2021
Due Date: **April 23, 2021, 11:59PM**. On Moodle
Extension: 15% penalty for each 24-hr period; Max. of 48-hrs past the original deadline

1 Assignment Description

The objective of this task is to implement a simple firewall rule matching algorithm. The match will be done a set of 6 fields, including source IP address, destination IP address, source port, destination port, protocol, and payload data.

The program will be invoked as follows:

```
./lab5-fw rulefile.txt pktfile.txt
```

Here, rulefile.txt specifies the set of rules to be stored in the firewall database, and pktfile.txt contains relevant packet field information for a set of packets. Each packet has to be compared against all the rules in the database and the matching rules should be output by the program.

1.1 Rule File

The rule file will consist of a set of records, where each record spans multiple lines, in the format given below:

```
BEGIN
NUM: <<integer>>
SRC IP ADDR: <<a.b.c.d/w>>
DEST IP ADDR: <<j.k.l.m/w>>
SRC PORT: <<integer1>>-<<integer2>>
DEST PORT: <<integer3>>-<<integer4>>
PROTOCOL: tcp | udp | icmp
DATA: <<string>>
END
```

- Rule number is an integer in the range $\{1, 2, ..., \infty\}$. It is assumed that each rule number will be in strictly increasing order and that all rule numbers are unique.
- IP Address range is given in standard notation. For example, 121.11.240.0/24 denotes that the IP address prefix length is 24 bits; 121.11.192.0/20 denotes that the IP address prefix length is 20 bits.

  The special value 0.0.0.0/0 indicates that any input IP address in the corresponding field (source or destination) will match this field of the rule.

  You can assume that each of the 4 components of the IP address (a, b, c, d) are within the appropriate limits (0 – 255) and that the prefix length (w) is between 8 and 32.

For information regarding prefix lengths and IP address prefixes, please refer to any standard undergraduate networks textbook (e.g. Kurose and Ross, Tanenbaum).

- The port number field specifies a range of port values, e.g. 1-127, 21-21, etc.

  Each port should be in the range of 1 to 65535. If a rule has a value outside this range, the rule will be discarded. Also, the first integer in the range should be less than or equal to the second integer. if this condition is not met, the rule is discarded.

  The special value *0-0* implies that any port number in the specified packet field (source or destination) can match this field of the rule.

- Data: A string of length 10 bytes (characters), where the characters are from the sets, 'A'–'Z', 'a'–'z', '0'–'9', and the " " (space) character. There is no need to check for validity of this data string.

  If the above string appears anywhere in a packet's payload data, then a match is reported for the data field.

  The special string with only one character ("*") will match all payload data.

A sample rule is shown below:

```
BEGIN
NUM: 123
SRC IP ADDR: 121.11.240.0/24
DEST IP ADDR: 0.0.0.0/0
SRC PORT: 0-0
DEST PORT: 21-22
PROTOCOL: tcp
DATA: FTPServer
END
```

1.2 Packet File

The packet file will consist of a set of records, where each record spans multiple lines, in the format given below:

```
BEGIN
NUM: <<integer>>
SRC IP ADDR: r.s.t.u
DEST IP ADDR: j.k.l.m
SRC PORT: <<integer>>
DEST PORT: <<integer>>
PROTOCOL: tcp | udp | icmp
DATA: <<string>>
END
```

- Packet number is an integer in the range $\{1, 2, ..., \infty\}$. It is assumed that each packet number will be in strictly increasing order and that all packet numbers are unique.

- IP Address range is given in standard notation, e.g. 128.205.31.1.

  You can assume that each of the 4 components of the IP address (a, b, c, d) are within the appropriate limits (1 - 255).

- Port number should be in the range of 0 to 65535. If a packet has a value outside this range, the packet will be discarded and not matched against any firewall rule.

- Data: A string of length 100 bytes (characters), where the characters are from the sets, 'A'–'Z', 'a'–'z', '0'–'9', and the " " (space) character . There is no need to check for validity of this data string.

A sample packet information is shown below:

```
BEGIN
NUM: 346
SRC IP ADDR: 121.11.240.15
DEST IP ADDR: 195.20.34.4
SRC PORT: 45678
DEST PORT: 21
PROTOCOL: tcp
DATA: Hello to FTPServer
END
```

Please figure out if the above packet matches the sample rule 123, listed earlier.

1.3 File Processing and Output

1. The program will first read all the input files and store it in memory in a suitable manner. This can be in a simple in-memory array. If you prefer to use a database (such as mySQL) to store the rules and call the appropriate APIs, you are free to do so.

2. After reading all the rules, the program will print the rule numbers of all valid rules stored in the database, as follows:

   `A total of <num1> rules were read; <num2> valid rules are stored.`

3. The program will then read the packet information file, one packet at a time. The program will match each packet's information against all the rules in the database and print the numbers of the rules that match this packet, as follows:

   `Packet number <id> matches <num> rule(s): <r1>, <r2>, <r3>.`

   It is possible that a packet may match no rule in the database.

   If the packet has invalid data, the program should print a message as follows;

   `Packet number <id> is invalid.`

   For each valid packet, calculate the time taken to find the matching rule(s). You can use suitable timers to measure the computation time. At the end, the average per-packet computation time will be reported.

4. After reading the entire packet file, the program will terminate with the following message (this includes all packets read including invalid ones):

   `A total of <num> packet(s) were read from the file and processed. Bye.`

2 Sample Session

Assume that you have created the file lab5-fw.c and the corresponding executables in your LAB5 directory.

```
DCF15> cd LAB5
DCF15> ./lab5-fw rule.txt pkt.txt
A total of 300 rules were read; 295 valid rules are stored.
Packet number 1 matches 2 rule(s): 113, 245.
Packet number 2 matches 3 rule(s): 143, 285, 290.
..
Packet number 13 is invalid.
..
Packet number 500 matches 1 rule(s): 16.
A total of <num> packet(s) were read from the file and processed. Bye.
Average time taken per packet: 12.34 microseconds.
```

3 What to Submit

Name your project directory as LAB5 (Note: ALL UPPERCASE). Once you are ready to submit, change directory to the directory above LAB5, and tar all files in the directory with the command:

`tar czf Lab5-RollNo.tgz LAB5`

The directory should contain the following files:

- Source Files
- Makefile
  Typing command 'make' at the UNIX command prompt, should generate all the required executables.
- a README file containing instructions to compile, run and test your program. The README should document known error cases and weaknesses with the program.
- Script command, demonstrating the output for given Input file and an input file set with a rule file having at least 300 rules and a packet info file having at least 500 entries.
- a COMMENTS file which describes your experience with the project, suggestions for change, and anything else you may wish to say regarding this project. This is your opportunity for feedback, and will be very helpful.

4 Help

1. WARNING ABOUT ACADEMIC DISHONESTY: Do not share or discuss your work with anyone else. The work YOU submit SHOULD be the result of YOUR efforts. Any violation of this policy will result in an automatic ZERO on the project, a potential F in the course, and other academic action.
2. Ask questions EARLY. Do not wait until the week before. This project is quite time-consuming.
3. Implement the solutions, step by step. Trying to write the entire program in one shot, and compiling the program will lead to frustration, more than anything else.
4. Questions raised within 24 hours of the deadline will not be answered; you have to make your own assumptions and justify them.
5. Sample Input Files and programs for generating input files will be provided.

5 Grading

- Parsing Rules File: 45 points
- Parsing Packet Information and Matching against Rules: 45 points
- Viva voce: 10 points

No README/COMMENTS: -5 points; Incomplete Compilation: -10 points