ECE 404 Homework #9

Due: Thursday 03/28/2023 at 5:59PM

1 Introduction

`Iptables` is an IPv4 packet filtering and network address translation tool used to set up, maintain, and inspect the tables of IP packet filter rules in a Linux kernel. Your learning objectives for this homework are as follows:

1. Understand the overall organization of the Linux based `iptables` tool
2. Write your own `iptables` rules based on specific, real-world requirements

2 Getting Ready for This Homework

Before embarking on this homework, it is advised that you familiarize yourself with the relevant material discussed in Prof. Kak's Lecture 18 Notes. Nonetheless, you might find the following review material on `iptables` helpful when writing your own firewall rules.

In its simplest form, `iptables` is a Linux firewall program that monitors network traffic to and from a server (your local machine in this case) using a set of tables. Each of these tables consist of a set of rules typically called chains, to which incoming and outgoing packets are subject to. When a data packet matches a particular rule, it is given a target. This target can be another chain, or one of the following special values:

- `ACCEPT` - allows the packet to pass through the firewall
- `DROP` - prohibits the packet from entering, with no indication to the sender that the connection failed.
- `REJECT` - prohibits the packet from entering, and sends an error message to the source indicating that the connection failed.
- `RETURN` - stops the packet from further traversal through the chain and tells it to go back to the previous chain.

Defining new rules with `iptables` boils down to appending a new rule to a specified chain. Shown below is how one might use `iptables` to add new rules.

```
sudo iptables -t <table> -A <chain> -i <in interface> -o <out interface> -p <protocol>
   -s <source> -dport <port no.> -j <target>
```

- `-t` – specifies which table you would like to append a new rule in

- `-A` – specifies which chain in the table you would like to append a new rule to
- `-i` – specifies the network interface incoming packets are received on (eth0, lo, ppp0, etc)
- `-o` – specifies the network interface that packets that are to be sent on
- `-p` – specifies the network protocol where your filtering process takes place (tcp, udp, icmp, sctp, icmpv6, all, etc.)
- `-s` – specifies the address from which traffic comes from (can be symbolic or numerical)
- `-dport` – specifies the destination pot number (22, 80, 8000, etc.)
- `-j` – specifies the target (another chain or one of the four previously mentioned special values)

Note *** The information provided to you in this section is just the tip of the iceberg when it comes to interacting with `iptables`. The requirements for the programming portion of the assignment will draw on additional knowledge found in Lecture 18 as well as the official Linux manual found here.

3 Programming Assignment

Before making any changes to your current firewall, it is important to save the current state of it. Shown in the code listing below is how you could accomplish this. **DO NOT SKIP THIS STEP!!!**

```
# this saves the current state of your firewall into a file call MyFirewall.bk
iptables-save > MyFirewall.bk

# this restores your firewall from file MyFirewall.bk
iptables-restore < MyFirewall.bk
```

Design a firewall for your Linux machine using the `iptables` packet filtering modules. It is likely that `iptables` came pre-installed with the Linux distribution you are using. Otherwise, you may need to upgrade it to get `iptables` to work. If you don't have a Linux environment on your PC, you can try setting up a virtual machine using software such as VirtualBox or VMware.

Write a set of `iptables` rules (as a shell script titled `firewall404.sh`) to do the following:

1. Flush and delete all previously defined rules and chains
2. ~~The default policy for the `INPUT/FORWARD/OUTPUT` chains on the `filter` table are `ACCEPT`. Change the default policies for all three chains to `REJECT`.~~
3. Write a rule that only accepts packets that originate from `f1.com`.
4. For all outgoing packets, change their source IP address to your own machine's IP address (Hint: Refer to the `MASQUERADE` target in the `nat` table).
5. Write a rule to protect yourself against indiscriminate and nonstop scanning of ports on your machine.
6. Write a rule to protect yourself from a SYN-flood Attack by limiting the number of incoming 'new connection' requests to 1 per second once your machine has reached 500 requests.

7. Write a rule to allow full loopback access on your machine i.e. access using `localhost` (Hint: You will need two rules, one for the `INPUT` chain and one the `OUTPUT` chain on the `FILTER` table. The interface is 'lo'.)

8. Write a port forwarding rule that routes all traffic arriving on port 8888 to port 25565. Make sure you specify the correct table and chain. Subsequently, the target for the rule should be `DNAT`.

9. Write a rule that only allows outgoing ssh connections to `engineering.purdue.edu`. You will need two rules, one for the `INPUT` chain and one for the `OUTPUT` chain on the `FILTER` table. Make sure to specify the correct options for the `--state` suboption for both rules.

10. Drop any other packets if they are not caught by the above rules.

To run your script, you will have to include a shebang line at the beginning of the file for `sh` (this program is almost always located in `/bin/sh`). Your script should be able to run without error. You will also need superuser privileges to edit any of the packet-filtering tables.

4 Spam Filter Account Set-up

Next week, we will be doing an assignment involving spam filters. For that assignment, we are providing you temporary ECN accounts on the Shay server. To obtain your credentials for your account, first find ECE404 on Brightspace. Under the "Grades" section for this class, there are two items of interest, one is your account's username (titled "login") and the other is your account's password (titled "Password"). The "Grade" value for these items indicate the username and password you will use.

On Windows, you can use PuTTY to ssh into the account. Alternatively you can ssh into the account using the following syntax on MacOSX and Linux:

```
ssh yourUsername@shay.ecn.purdue.edu
```

Please make sure that you can log into this account. Should you face any trouble, please reach out to Joseph as soon as possible to get the issue resolved. Once you have successfully logged into the account, feel free to change your passworld using the passwd command. For future reference the email address associated with this account is `yourUsername@ecn.purdue.edu`. Do not try to log into this account with Microsoft Outlook or ECN webmail.

Once you are able to access your account, follow the instructions below to get your account up and fully operational.

1. Unzip the attached tar.gz file that contains a text file named `dot_procmailrc` and a Perl script named `GET_MESSAGE_INDEX`

2. Apply the `dos2unix` command to these files to remove the carriage return characters added after line feed. Do not skip this step as the `dot_procmailrc` file is sensitive to such characters, because it contains regular expressions.

3. Carefully read the comments in the `dot_procmailrc` file and make the necessary changes to Recipes 2 and 3.

- Recipe 2 requires that you place the name of your special account in the last line of the recipe
- Recipe 3 requires that the string user_name in the last line be repalced by your email account at Purdue

4. Rename `dot_procmailrc` to `.procmailrc` and store it your new account's home directory (e.g. `/home/shay/a/ece404q8`). You can invoke sftp or scp to move the file from your local machine to the new account.

5. Create a new directory called `Mail` at the top level of your new ECN account and place the `GET_MESSAGE_INDEX` script in that directory

6. Send a test email message to your new account. You can verify that you recived this test mail if a file titled `logfile` is created in your `Mail` directory.

7. The best command-line tool for processing the email recived by your new account is `mailx`. Do `man mailx` to see all the options that go with this command. You can even use this command to send messages to others. In my experience, mailx works best when you use SSH to access the account directly (as opposed to using ThinLinc or ECEGrid).

8. After your account has become operational, please subscribe to random newsletters, newsgroups, websites... etc. This process will cause spam to be directed to your email addresses in a fairly short time. Include a page worth of logfile contents in your pdf submission.

Submission Instructions

- For this homework you will be submitting a zip file titled `hw09_<last name>_<first name>.zip`, which consists of:
  - A pdf titled `hw09_<last name>_<first name>.pdf` containing:
    - For each requirement in section 3, your solution as well as an in depth explanation of your solution. In depth to the point where a beginner (who only has basic knowledge of `iptables` can follow along.
    - Output (e.g. screenshots) of your updated firewall after running `firewall404.sh`. The command for this is `sudo iptables -L`
    - A page worth of logfile contents from the Mail directory of your spam accounts
  - The file `firewall404.sh` containing your `iptable` commands
  - The updated `.procmail` file