Assignment 6

Network Scanning and Iptables

In this assignment, you will get familiar with network scanning and iptables rules. Iptables is used to set up, maintain, and inspect the tables of IP packet filter rules in the Linux kernel. Several different tables may be defined. Each table contains a number of built-in chains and may also contain user-defined chains. Each chain is a list of rules which can match a set of packets. Each rule specifies what to do with a packet that matches. Additional information on iptables can be found at https://en.wikipedia.org/wiki/Iptables.

The assignment assumes background knowledge in networks and familiarity with bash scripting. Specifically, you will develop a bash script (named "adblock.sh"), that uses the linux command iptables in order to create a simple adblocking mechanism that is described below. Information about the iptables command can be found in the appropriate man pages (i.e., man iptables(8)). A Bash shell script, at its simplest form, is just a list of shell commands separated by newlines and can be used for running multiple commands together, customizing administrative tasks, performing task automation, etc. You can find a Bash scripting cheatsheet at https://devhints.io/bash#functions.

The adblock.sh script is responsible for generating a set of firewall rules that block access for specific network domain names. Unfortunately, the iptables command doesn't work with domain names and you have to resolve the domain names provided in the "domainNames.txt" and "domainNames2.txt" files to the appropriate IP addresses. You may use different network administration command-line tools (e.g., dig, host, nslookup) for querying DNS nameservers for information about host addresses. It is possible for one domain name to have multiple corresponding IP addresses, as such you can use your favorite command-line tool (e.g., grep, awk, sed, etc.) to parse the results of the DNS query. The iptables(8), dig(1), host(1), nslookup(1), grep(1), awk(1) and sed(1) are the man pages for the aforementioned command-line tools.

Your adblock.sh script is expected to do the following:

1. Open domainNames.txt and domainNames2.txt and find the same and different domain names in both files.
2. Write the IP addresses of the same domains in "IPAddressesSame.txt" and the different domains in "IPAddressesDifferent.txt"
3. Drop all packets from the IP addresses of "IPAddressesSame.txt".
4. Reject all packets from the IP addresses of "IPAddressesDifferent.txt"
5. Save rules to "adblockRules" file (this file does not have a filename extension).
6. Load rules from "adblockRules" file (this file does not have a filename extension).

7. Reset rules to default settings (i.e. accept all).
8. List current rules.

You are given the following files. You are **NOT** allowed to change the naming scheme of the given files.

Files:

- adblock.sh
    - The corpus of the adblock.sh script.
- domainNames.txt, domainNames2.txt
    - Files Containing network domain names.
- IPAddressesSame.txt, IPAddressesDifferent.txt
    - Empty files used to write the IP addresses of the network domain names.
- adblockRules
    - Empty file used to load/save the iptables rules.

Tool Specification

The provided corpus already implements the options/arguments of your tool. Again you are **NOT** allowed to change the naming scheme of the options. Your script will receive the following arguments from the command line upon execution.
Options:

-domains	Find different and same domains in ‘domainNames.txt’ and ‘domainNames2.txt’ files and write them in “IPAddressesDifferent.txt" and IPAddressesSame.txt" respectively
-ipssame	Configure the DROP adblock rules based on the IP addresses of ‘IPAddressesSame.txt’ file.
-ipsdiff	Configure the REJECT adblock rules based on the IP addresses of ‘IPAddressesDifferent.txt’ file
-save	Save rules to ‘adblockRules’ file.
-load	Load rules from ‘adblockRules’ file.
-list	List current rules.
-reset	Reset rules to default settings (i.e. accept all).
-help	Display help and exit.

Hints

1. The iptables command requires root privileges. Run your script with sudo.
    a. With great power comes great responsibility. Accidentally deleting crucial OS partitions is a common issue.
2. You can make your script executable with chmod +x adblock.sh
3. Use the -j DROP option to reject the connection.
4. Use the -j REJECT option to reject the connection.
5. Use iptables-save to save rules to file.
6. Iptables-restore to load rules from file.
7. DNS queries for many domains can take some time. You can force a command (or more) to run in the background (e.g., host google.com &) and then use wait to suspend the execution until the subprocesses have finished.
8. Both domainNames.txt and domainsNames2.txt files contains some popular advertising domains. Feel free to add more or remove some. Your script should be able to handle hundreds of domain names easily.
9. Obviously there are specific website categories that contain many advertisements for you to evaluate your tool.

Questions

1. After configuring the adblock rules test your script by visiting your favourite websites without any other adblocking mechanism (e.g., adblock browser extensions). Can you see ads? Do they load? Some ads persist, why?

Notes

1. The naming scheme of the given files must remain as-is.
2. The options defined in the “Tool specification” section must remain as-is.
3. You need to create a README with your name, your AM and a short description (1-2 lines) of your implementation.
4. You must submit the following files: adblock.sh, README.
5. You should place all these files in a folder named <AM>_assign6 and then compress it as a .zip file. For example, if your login is 2020123456 the folder should be named 2020123456_assign6 you should commit 2020123456_assign6.zip