Tu Lam

CS 373 (Defense Against the Dark Arts)

Dr. Bram Lewis

November 11th, 2021

**Homework #3**

In the case for this homework, we were task to explore the VM using a script and continue writing the script to look at the data in the ***R*** and ***O*** file. This script will be a way for us to explore the concept behind the network security in cybersecurity. Through this, we will get a sense of the network going through the cybersecurity background via these two files that were given.

First, the homework gave us a tar file and in there, there are a ***bunch of script*** and ***CSV*** files that we can look at the network security for this week homework. Then we have a lab file where it gave us the background of network security and learn about the ***TCP***, ***IP protocol***, ***UDP***, and more. In there, there is also couple instruction we will use and follow to answer and explore what the lab offer for this homework as well. Below, we will look at individual questions and answer them in order. In total, there are ***11 questions*** to answer with an extra challenge if we want to do it or not.

Next, after extracting all the files that were given in the homework, I was able to run the python file and look at the ***R.csv*** file using the ***scancsv.py*** file. Through looking through this, I found that the ***IP protocol*** have number align with ***1, 2, 6, 17***. A figure below will show those number and what do they print out to be. In the lab manual, I found out that each number represent a protocol. For example, 1 represent the ***ICMP***, 2 represent the ***IGMP***, and so on. Then we move onto locating the ***IP address*** using the grep function and then we will move onto the questions that are ask in the lab.

![Graphical user interface

Description automatically generated with low confidence]()

**Figure #1**: *Showing what R.csv offer when run the program*

**1**. *Extend your script’s statistics gathering to count the use of all well-known destination port numbers for TCP and UDP (ports 1-1024). For example, you should be able to look up in your output how many TCP packets have destination port 80 and how many UDP packets have destination port 53. Run your new script on R and O data. Enable this function using a ‘-stats’ flag (i.e., the script should have no output unless there is a -stats flag in the command line).*

**Answer**: To achieve the script to display the ***TCP*** and ***UDP*** if looking from port ranging from 1-1024 (In this case just using ***80*** and ***53***), we can go inside the scancsv.py to look and add command to help print out the data if a port is being called. To get the correct data printed out, I would need to look at the ***R*** and ***O*** file to see the layout of the CSV file look like. With the trial and guess and looking through the file, below are couple figures to help represent how to see ***TCP*** and ***UDP*** content.

![Text

Description automatically generated]()

**Figure #2**: *Display the code to implement the printout of TCP and UDP*

![Text

Description automatically generated]()![Text

Description automatically generated]()

**Figure #3**: *Display printout of TCP & UDP of O.csv*

![A picture containing text

Description automatically generated]()

**Figure #4**: *Display the TCP & UDP of R.csv*

**2**. *Based on this information, characterize the main functions on each network. What kind of a network is it? (e.g., work, home, data center, ISP)*

**Answer**: Through looking through both ***R*** and ***O*** file. I saw that TCP in the ***R*** file have 9455 users from ***port 139*** while ***port 53*** have around 400 users. From that, we can gather through online help that ***port 53 deals with DNS*** while ***port 139 deal with NetBIOS service***. Then move onto ***O*** file, I see that TCP ***port 25*** and UDP at ***port 53*** have the most traffic. At ***port 25 is SMTP client*** and ***port 53*** have discuss in the previous statement for R file. And these are just some examples I will be using to look at the network that people are connected to.

**3**. *Add to your script an option called “-countip” which creates list of distinct IP addresses with their usage counts. Sort the list by the usage count, not by the IP address.*

**Answer**: Now, we then add in a command called ***“-countip”*** into the argument and it will list distinct IP addresses with usage counts. With this, we can determine if the last question we answer are correct. Below will be some more screenshot of code and display of this new command.

![Text

Description automatically generated]()![Text

Description automatically generated]()

**Figure #5**: *Code of implementing the “-countip”*

![Text

Description automatically generated]()

**Figure #6**: *The output of “-countip” using R file*

**4**. *Run your “-countip” script on R and O data. Does this inform your answer in [2]?*

**Answer**: With looking at the ***-countip*** command that was implement in question 3, it looks like it ***does inform*** what question 2 when it was answer. Most of the user uses the port are listed on the IP address in high demand, but at the same time, we can’t really tell much if the port is exactly the port that people visit.

**5**. *Attempt to determine the network number (network prefix) that seems to dominate the traffic.*

**Answer**: Looking in both files, the most ***R*** file get dominated with the traffic that start with ***10.5.63.xx*** prefix. While the ***O*** file have the traffic of ***192.245.12.xx*** prefix.

**6**. *Generate sorted output from ‘-countip’ for the IP protocols to identify all the IP addresses that use:*

**a**. *GRE (Generic Routing Encapsulation)*  
 **b**. *IPSEC*  
 **c**. *OSPF*

**Answer**: Below will be some figures showing the general output of the ***-countip*** with couple usage of the options that are given. Through the file, R file did not have any output while O file does and to use the command, add in another command ***“-other”*** to print out.

![Text

Description automatically generated]()![Text

Description automatically generated]()

**Figure #7**: *Display some codes of implementing Question #6*

**7**. *Find another network prefix that also seems to be associated with this traffic*.

**Answer**: When determine the network prefix, we will look at the output for the ***O.csv***. Since the ***R.csv*** was blank, I believe the clear network prefix that can be seen is the 207.182.xx.xx are in the ***R.csv*** file.

**8**. *Does the OSPF information inform your answer to question 2?*

**Answer**: When talking about ***OSPF***, this is the ‘standard’ routing protocol for Internet routers, allowing them to discover the topology and choose the best routing paths, we can find that R.csv file does not have any prefix network route from the last answer question, believe to be determined that it is more onto sharing a route rather than mapping one. From this, we can see that question 2 display the center network that most of the IP addresses share when routing to Internet router.

**9**. *Add an option to your script ‘-connto’, which counts the number of packets sent to each service (ports 1-1024) on the network. For example, a dictionary maps each ipdst to the tuple <proto, dport>, where proto is tcp or udp, based on the IP protocol (6 or 17) and dport is the value of tcpdport or udpdport.*

**Answer**: Below are some images of the implementation of ***-connto*** command line.

![Text

Description automatically generated]()![Text

Description automatically generated]()

**Figure #8**: *Display code of the Question 9 asking to implement the -connto*

![Text

Description automatically generated]()

**Figure #9**: *The output of the -connto output*

**10**. *Run your -connto option on R and O data (ignore anything that ends in .255 – this is a broadcast address). Does this suggest a set of servers to you?*

**Answer**: Below are some results printing out in R file that limit in 20 lines. The O files was too big to run, but the code should work for both file and ***-connto*** above in Figure #9 show an example result.

![Graphical user interface, text

Description automatically generated with medium confidence]()

**Figure #10**: *The code of Question 10*

**11**. *Update your answer from [5] based on this information.*

**Answer**: From the cmd implement from above, I ***really cannot say much*** has changed looking at the outputs from the R.csv file and the O.csv. The information displays the common prefix, but the information still remains the same with the command line added.