Jack Berberian

CIS 245

Networking Assignment

* Current network setup, including notes of what is installed, the last time it was updated, and any changes I’ve made to the defaults.
  + In order to share files on both virtual machine systems, I installed WinSCP on my host system and I connect to it through there with the login credentials I made in the initial setup.
* At least one page of research on what I’ve chosen to add to my script and why. Cite my sources and explain why I think these additions are valuable. (Sources do not count)

While there are lots of valuable commands that I could’ve chose to add, I tried to focus on several of the most common ones that are more useful in possible day to day tasks. In my personal experience, I use some of these on my computer frequently and some not so much.

For the first command, I thought the command to find my IP address was a valuable addition to the script. For example, on my home network, I sometimes have to find the IP addresses of my computers but for some less advanced users, having a nice and easy script to run might be better off for them. While network hostnames can be used for network shares, and other various services, sometimes knowing the exact IP address of a computer is important. For example, if someone hooks up their computer to an ethernet connection instead of Wi-Fi and they were using the IP address of the Wi-Fi connection, they’d want to find the correct corresponding IP to make sure they were using the more reliable ethernet interface.

For the second command, I thought being able to identify the DNS for the computer was also an important tool/command as well. On my own network, I have a Raspberry Pi managing it for me, but on other networks some people might not have that and therefore, need to be able to manually see what they’re. One example of where it might useful is if a user is having issues with their network connection on specific websites only, they might be suffering from a DNS resolution issue and it’d benefit them by finding and changing their IP.

Thirdly, while it may not be important for less technical users, knowing which ports are open on a computer is very valuable. One of the best practices I’ve learned for cyber security throughout my learning journey was to never have any ports open that I do not need. While it may be more so important for the ‘outside’ of a computer network, it is still important to close unused ports on devices on a network. If a network gets compromised, every open port on a computer is a threat vector or attack vector that could increase the susceptibility for it to get comprised.

Also, another important command that I thought was worth adding is the Ping command. In my personal experience, this may be the command that I’ve used the most. Oftentimes I have to connect to network shares on other computers, and manage other devices and it helps me from needing to physically check on a computer; I could be in a remote location and not need to be in close proximity to the device to check to see if it’s on. An additional flag that I added was the -c flag in order to limit the number of times the command is run. Non-technical users might not know what to do but unless they stop the command, it could run forever, affecting their system and the remote server. Hence the reason for me adding a prompt asking the user to enter their desired amount of queries.

Lastly, a favorite command of mine is traceroute. I don’t use this a lot personally myself but I like to use it on foreign servers. It’s interesting to see what routes my connections take to different servers. Although, in combination with the ping command, it can be helpful to troubleshoot network issues. Hypothetically, if my internet was not having issues, but I couldn’t connect to some random website, a traceroute would help because I can ‘follow the path’ that my computer takes and find out where the connection stopped. It would help me figure out if it was an issue with my network or if it was someone else’s ISP.

* Citations
  + <https://www.cyberciti.biz/faq/unix-linux-check-if-port-is-in-use-command/>
  + <https://www.tecmint.com/find-my-dns-server-ip-address-in-linux/>
  + <https://www.thegeekstuff.com/2009/11/how-to-execute-ping-command-only-for-n-number-of-packets/>
  + <https://kapeli.com/cheat_sheets/Bash_Test_Operators.docset/Contents/Resources/Documents/index>
  + <https://linuxize.com/post/bash-if-else-statement/>

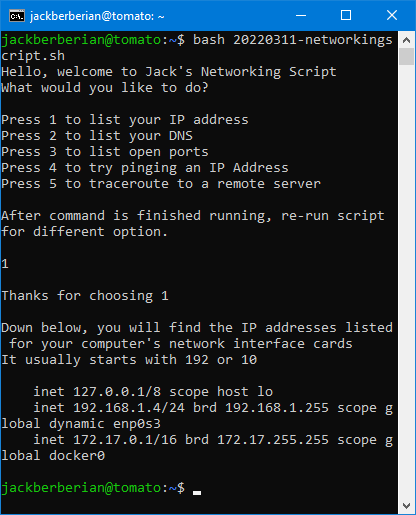
A well commented and well tested script, and a sample run of your script on each server. Include any instructions needed for running the script. Such as, does it need to be run in a special folder? Does it save the text file to a special place and where that might be? Which commands you’ve chosen to put in the script and why.

Sample Run:

CentOS:

Having network problems again. Will update later.

Ubuntu:



Commands I chose to put in the script and why:

The commands I chose to put in the script were

* ip address | grep -E '((1?[0-9][0-9]?|2[0-4][0-9]|25[0-5])\.){3}(1?[0-9][0-9]?|2[0-4][0-9]|25[0-5])'
  + The reason why I added this was because I needed to find the IP address, and I wasn’t sure if ifconfig would be installed on someone else’s system so I wouldn’t have to install something on someone else’s computer and still be able to list IPs. Afterwards, I piped the output of ip address to grep and used a regex to only find IP addresses and then output the command.
* grep "nameserver" /etc/resolv.conf
  + The reason why I used this was because nameserver is stored in resolv.conf and I could navigate to that file and then display only nameserver.
* sudo netstat -tulpn | grep LISTEN
  + The reason why I added -tulpn | grep LISTEN was in order to find the open or listening ports. -t would find TCP, -u would find UDP, -l lists only listening ports, and -p can show the program that which port belongs too. By piping the command to grep, I can clean up the output and just show the ports that are listening and not all of them.
* ping $IP -c $numberoftimes
  + The reason why I added $IP and -c $numberoftimes was because I needed to be able to ping a user specified inputted IP, and then restrict it to the number of times that they wanted instead of running it for infinity. $IP was substituted for the IP, and $numberoftimes, was the number of times it got pinged. The reason why I added the command was because I use this frequently and thought it would be useful for non-technical people, especially with the added -c flag so it wouldn’t run for infinity.
* traceroute $IP
  + Similar to ping, the reason why I added $IP was because I needed it to traceroute an IP that I user added. The reason why I added this command was because I thought it was interesting cool one and a sometimes helpful network diagnostic tool.

Script:

2022-03-11 03:26PM

#!/bin/bash

echo "Hello, welcome to Jack's Networking Script"

echo "What would you like to do?"

echo ""

echo "Press 1 to list your IP address"

echo "Press 2 to list your DNS"

echo "Press 3 to list open ports"

echo "Press 4 to try pinging an IP Address"

echo "Press 5 to traceroute to a remote server"

echo ""

echo "After command is finished running, re-run script for different option."

echo ""

read input

if [ $input == "1" ];

then

echo ""

echo "Thanks for choosing 1"

echo ""

echo "Down below, you will find the IP addresses listed for your computer's network interface cards"

echo "It usually starts with 192 or 10"

echo""

ip address | grep -E '((1?[0-9][0-9]?|2[0-4][0-9]|25[0-5])\.){3}(1?[0-9][0-9]?|2[0-4][0-9]|25[0-5])'

echo ""

elif [ $input == "2" ];

then

echo ""

echo "Thanks for choosing 2"

echo ""

echo "Your DNS server will be listed after nameserver:"

grep "nameserver" /etc/resolv.conf

echo ""

elif [ $input == "3" ];

then

echo ""

echo "Thanks for choosing 3"

echo ""

echo "Installing net-tools incase you don't already have it"

sudo apt install net-tools -y

echo "Searching for Open Ports"

sudo netstat -tulpn | grep LISTEN

echo ""

elif [ $input == "4" ];

then

echo ""

echo "Thanks for choosing 4"

echo "What IP address would you like to ping?"

echo "Enter it here, please: "

read IP

echo "The IP address that you entered was $IP"

echo "How many times would you like to ping it? Enter a number 1-10"

read numberoftimes

ping $IP -c $numberoftimes

echo ""

elif [ $input == "5" ];

then

echo ""

echo "Thanks for choosing 5"

echo "What IP address would you like to traceroute?"

echo "Enter it here, please: "

read IP

echo "The IP address that you entered was $IP"

echo ""

echo "Installing traceroute if you don't already have it"

echo ""

sudo apt install traceroute -y

echo ""

traceroute $IP

echo ""

else

echo "Sorry, unexpected input. Please try again"

echo""

fi