User-Friendly IoT Networking Scanning Tool

Ashlyn Cooper School of Computing Clemson University ancoope@clemson.edu Kaitlyn Pierce School of Computing Clemson University kjpierc@clemson.edu

ABSTRACT

In this paper, we discuss our research, design, and implementation of a command line tool that scans the network for IoT, Internet of Things, devices and monitors their activity. We explore the pre-existing network scanning tool *nmap*, which provides important information regarding connected devices but does not present this information in a user-friendly manner. Our tool simplifies the network information to view connected devices in a relevant, condensed way. This tool was built for an Ubuntu virtual machine environment on top of the existing network scanning tool, *nmap*, and filters the specific device information shown based on user specified thresholds.

INTRODUCTION

In this paper, we will detail our network scanning tool that allows users to view IoT devices connected to their network based on a user selected type of scan. The networking scanning tool *nmap* outputs a list of ports, devices, operating systems, and more that help users identify the device connection status; however, this output format can become cluttered and confusing to a new system user. Our tool aims to add a more user-friendly interface to network scanning.

Our tool prompts users to select the type of network report they would like to create through command line input and rejects any invalid input by re-prompting the user for the correct selection. There are five types of reports the user can generate. Those report types include summarized, aggressive, firewall detection, grepable, and verbose. After report selection, the tool will run a network scan and filter the results by the specified user input. The tool can also output the scan results to a file per the user's request.

This tool was designed to provide users with a more convenient way to view connected IoT devices on their network, especially those who may not be familiar with the lengthy, clunky Linux outputs. With the popularity of IoT devices and the increasing risk of unknown devices gaining access by connecting to networks, this tool will allow users to quickly and efficiently understand what kinds of and how many devices are connected to their network.

BACKGROUND

IoT, or Internet of Things, devices are computing devices that connect wirelessly to a network and have the ability to transmit data [2]. They vary widely and range from smart speakers and TVs to wearables and smart appliances. The number of IoT-connected devices has been increasing steadily over the past decade and is expected to grow to 29.4 billion by 2030 [1].

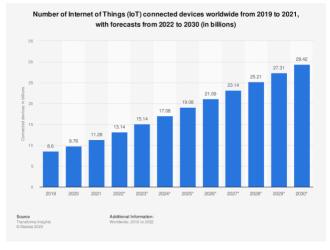


Figure 1: A bar chart of current and future IoT-connected devices

Due to the growing number of IoT devices on networks, it is important for system administrators and network owners to know who and what are connected to their network. Knowledge of the devices on a network is especially important, given that IoT devices pose certain security and privacy risks. Researchers have already shown that these types of devices can be remotely hacked and used to access the main network, posing a threat to the device and network owner. In the case of any cyber hack, identifying the devices on the work is the most crucial step towards analyzing the attack and preventing further harm [3].

Nmap, short for Network Mapper, is a free, open-source tool that performs network scans to monitor network devices, scan open ports, detect running operating systems, and more [4]. While extremely powerful and useful, it can be difficult for new users or inexperienced network administrators to fully understand, as depicted in the below image.



Figure 2: The output of the nmap command in a Linux terminal

MOTIVATIONS AND OBJECTIVES

The goal of our project was to successfully develop and implement a program that scans, analyzes, and outputs the IoT devices connected to a specified network in our Ubuntu Linux environment. We wanted to create a responsive, user-friendly tool for network users that would allow users to obtain and understand specific network scan reports based on various filters.

This tool would be extremely useful for all computer and network users due to the prominence of IoT devices and the increasing risk of cyber-attacks. As we know, cybersecurity is the responsibility of everyone; therefore, a simpler, straightforward tool such as ours would be helpful for less experienced users to utilize on their networks for threat analysis.

METHODOLOGY / DESIGN

When designing the tool, we aimed to make the tool as simple and straightforward as possible for the user. To achieve this, we made our tool rely on only one C file that the user can run through a Linux terminal. Once the program is executed, the program prompts the user for the type of scan they would like to perform and requests single-character inputs from the keyboard. Along with the prompts, the program gives detailed descriptions of what each option means for the user to better understand the option they are selecting, as seen in Figure 3.

```
What type of summary report would you like to generate?

- Summar Lead: This scan will return a list of fully IP addresses on your network along with the manufacturer information for the device if available. To use links node, enter 'S'

- Aggressive: This scan will try to aggressively get nore information out of the device. This scan will including operating system detection, version detection, script scanning, and tracerouse detection.

To use this node, enter 'A'

- ACK Scan: This scan will help differentiate between statefull and stateless firewalls. The scan report will show open and closed ports with a lable of filtered or unfiltered, with unfiltered meaning that the port was reachable by an ACK packet. If the port is labeled as filtered, a firewall is preventing the reach to the port.

- Grepo utpuit: This scan will provide you with a grepable output. The fornat lists each host on one line and can be searched with standard Unix tools such as grep, awk, cut, etc.

- Verbose: This scan will provide on output with a higher verbosity level.

It will print information on the scan process.

To use this node, enter '0'

To quit the program, enter 'Q'
```

Figure 3: Description of each type of generatable report.

Figure 4: Example output of an 'Aggressive' report.

Similarly, we automated some of the steps in our tool to replace the typical steps that a user would have to complete when using the *nmap* tool. To use *nmap*, users must first find or know the IP address of their network. Typically, a user would have to run a Linux command and analyze the output to find this information, but with our tool, this is done automatically. Our tool finds the IP address of the system and uses it when the program is executed, rather than relying on the user to go through the extra step of retrieving the IP address themselves.

Our tool additionally offers users the option to print their report output to a text file in their current directory. This allows users to reference this report outside of the executed program. This functionality and outputted text file can be seen in the below images.

```
Enter choice:
5' Printing results to terminal and file called 'results.tzt'
5'sarting Mmap 7.80 ( https://mmap.org ) at 2022-11-27 22:23 UTC
Nmap scan report for kjplerc (127.0.1.1)
10st is up.
Nmap done: I IP address (1 host up) scanned in 0.00 seconds
Nould you like to generate another report? If you wish to quit enter 'Q', other wise press any key to continue
```

Figure 5: Our tool's output when the user requests the report to be saved to a file in their current directory.

```
kjpierc@kjpierc:-/clt$ cat results.txt
Starting Nmap 7.80 ( https://nmap.org ) at 2022-11-27 22:23 UTC
Nmap scan report for kjpierc (127.0.1.1)
Host is up.
Nmap done: 1 IP address_(1 host up) scanned in 0.00 seconds
```

Figure 6: The contents of the results text file after a 'Summarized' report is run on the system.

ANALYSIS/RESULTS

We were able to build a tool for Ubuntu that utilizes *nmap* to make an easier way for users to investigate the devices connected to their network. Through use and testing our tool was able to run all the implemented modes as well as redirect results to a file if the user chooses that option. The tool also successfully allowed users to run multiple scans without restarting the program before quitting.

CONCLUSIONS AND FUTURE WORK

The tool built provides a good start for novices or inexperienced network administrators to begin learning about the IoT devices connected to their network. With this tool, the user needs to go through fewer steps and does not need to have a strong understanding of *nmap* to be able to investigate network devices. By using the tool, users can simply start the program and enter a few keys to gain information about IoT devices on their network along with provided context about what the network scan results mean.

While our tool is functioning in its current form, it does have its limitations and weaknesses. The first weakness is that *nmap* must be installed on the user's machine for the program to function. The second weakness is that we were unable to thoroughly test our program with multiple IoT devices, as we were lacking devices to test with. Because of these limitations, there is still more testing that could and should be done for the tool.

While the tool provided a great start for users getting into network administration, there are many features that could be improved and implemented to make the tool more helpful to users. The first improvement would be to implement a graphical interface rather than a command line interface. While the command line interface is simple and straightforward to use, a graphical interface could be more approachable to novice users and could display information more clearly. The second improvement would be to better format the output of the scans. Currently the program returns an output that is relatively clunky and hard to read, so doing more processing on the results before displaying them could be more helpful to the user trying to understand the devices connected to their network.

As more and more devices are being connected to the internet, users and network administrators need to be constantly aware of the devices connected to their networks. Our tool provides a helpful start for users looking to investigate their networks and with future

Extended Linux Process Management Tool

improvements could be an incredibly simple and useful tool for detecting IoT devices.

REFERENCES

- Lionel Sujay Vailshery. (2022). Number of IoT connected devices worldwide 2019-2021, with forecasts to 2030. Retrieved Nov 22, 2022 from https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/
- Brien Posey. (2022). IoT Devices (Internet of Things Devices). Retrieved Nov 22, 2022 from https://www.techtarget.com/iotagenda/definition/IoT-device Ali Imran Nagori. (2020). How can I see all Active IP Addresses on my Network?
- Retrieved Nov 22, 2022 from https://linuxhint.com/see-active-network-ip-
- addresses/
 Valentin Bajrami. (2020). Running a quick NMAP scan to inventory my network.
 Retrieved Nov 22, 2022 from https://www.redhat.com/sysadmin/quick-nmapinventory