Local Network Audit with Nmap

Larsen Close

Computer Science, Metropolitan State University, Denver, CO
Student: lclose@msudenver.edu

ABSTRACT
Extensive Nmap scanning to fingerprint a local network is demonstrated and areas of interest are identified. Anomolies, security concerns and unexpected results prompt further investigation. Actions taken upon the basis of the findings are then discussed.

MOTIVATION
We often only pay attention to our internet connected devices when we lose connectivity. We then try and restore the connection as quickly as possible. The truth is that as the Internet Of Things expands and our world increasingly digitizes the danger of these devices to our safety, privacy and security increases. The steps clearly demonstrated in this paper are steps everyone should take regulary on their home network for their sake and for their families sake.

INTRODUCTION
I started fingerprinting my network with a few areas of heightened interest. In order to ensure a thorough investigation I first used widespread scanning and looked for abnormalties, anything which surprised me or that raised security concerns. I made sure to turn on and log into every device on the network before conducting the scans. Also in order to asses the behavior of a new automated system for the door lock and thermostat, mandated by the poperty owner, I connected to the network for the first time. The device has been running on cellular connection but has both options.

how does Tails handle local traffic?
To research the local network scan response of Tails, The Amnesic Incognito Live System (a security-focused Debian based distribution of Linux) I booted a machine with the live system during the scans. Starting with cursory scans and increasing the intensity each time while saving the results allowed for a detailed fingerprint of the local network.

let shodan.io scan for you
To gain information about my public IP address and check for any outward facing access I used shodan.io.[2] After using their search function I created a monitor with triggers to notify me of any any unexpected activity. After scanning extensively I created the graphs below to organize the findings, visualize them and decide upon which areas to increasingly focus. Host data from the scans is combined with DHCP Reservations from the routers to verify the identity of devices and fill gaps left by scans.

METHODS AND MEASUREMENT
To begin scanning we needed to know which IP addresses to tell nmap to scan. Using the command:

```
ifconfig | grep broadcast
```

We determined the local networks and the subnet ranges. As can be seen in **Figure 1**, the networks are `10.0.0.0/24` `192.168.0.0/24`. The `netmask 0xffffff00` tells us the range for hosts is limited the last octet, digits 1-254.

Figure 1. First the command 'ifconfig | grep broadcast' identifies the local networks.

super user privileges for scanning

Next I ran a ping scan of both networks to quickly get a list of the hosts. It's likely that you will not get a full list if you do not run this command with super user privileges. Testing this without sudo resulted in 1 host and 9 hosts respectively. To run ping scans for both networks we used:

```
sudo nmap -sn 192.168.0.0/24
sudo nmap -sn 10.0.0.0/24
```

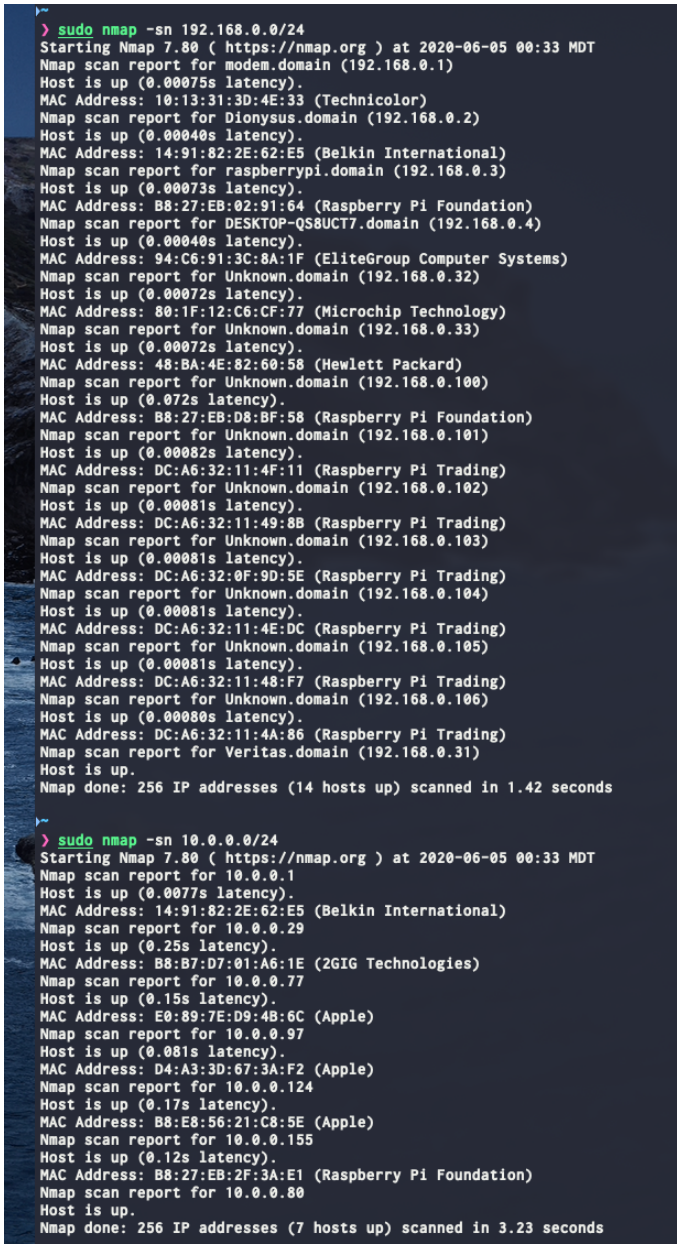

Figure 2. Ping scan returning 14 and 7 hosts.

As we can see in **Figure 2**, the ping scan with sudo returned a list of 14 hosts and 7 hosts respectively.

Nmap flags for more thorough scans

After completing the quick and simple ping scan we scanned both networks using nmap with the flag `-A` to initiate an agressive scan. We also included `-v` to increase the verbosity of our results and `-T5` to turn the scans speed all the way up. The commands used:

```
sudo nmap -T5 -A -v 192.168.0.0/24
sudo nmap -T5 -A -v 10.0.0.0/24
```

At this point it was necessary to create **Table 1** to organize the results and create an overview that could actually be visualized. These types of scans return so much information that it demands systematic organization. Zenmap can do a good job of helping with this but it has not been running well for me on macOS Catalina. Another option which I employed is to use an additional flag when running nmap `-oX 'scan-%T-%D.xml'` which outputs the scan in xml format named by the date and time. To give some full examples:

```
sudo nmap -oX 'scan-%T-%D.xml' -T5 -A -v 192.168.0.0/24
sudo nmap -oX 'scan-%T-%D.xml' -T5 -A -v 10.0.0.0/24
```

monitoring public IP addresses with shodan.io

Due to the ethical and legal restrictions on scanning the public IP owned by Century Link I used shodan.io to search for my IP and set up a monitor for the address.[2] An easy what to determine your public IP address is with command:

```
curl ifconfig.me
```

RESULTS

To begin with, shodan.io allowed us to determine that our router has an open outward facing port running http with authentication.[2] **Figure 3** shows a screen shot from their webservice.

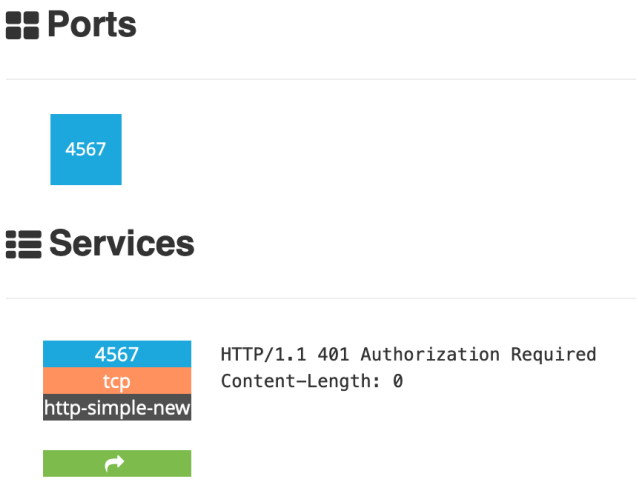

Figure 3. From website https://www.shodan.io/ often called "the most dangerous search engine in the world"

Compiling all of the findings

The table below, **Table 1**, is a compilation of all the scans. The aggressive scans discovered more hosts missed by the privileged ping scans. Immediately some interesting things are visible from the table. First, my HomePod has two IP addresses on the same network. After logging into and checking the router it appeared that both were running on the 5G wifi. While going through the devices I was familiar with all of the results except for 192.168.0.32 which was running ssh on port 22. I check the routers DCHP Reservations and found no information there either. The scan did show the mac address `80:1F:12:C6:CF:77` as well as some additional information. Seen here using a targeting scan of just this device: `sudo nmap -T5 -A 192.168.0.32`

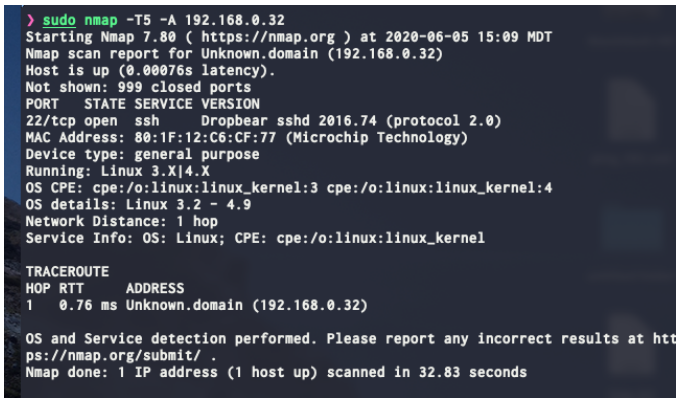

Figure 4. Resulting scan of unfamiliar device

	Hostname	Device	Purpose	OS	Open	Filtered	Services
10.0.0.001	Dionysus	Router	Gateway	Linux	53,80,443,10000	-	ssl,http
10.0.0.029	HD100	Webcam	Security	-	554,49152	-	rtsp,UPnP
10.0.0.077	Daemon	HomePod	Music	audioOS	62078	24 dif	wiretap
10.0.0.080	Veritas	MacLaptop	PC	macOS	3000	-	grafana
10.0.0.097	Daemon	HomePod	Music	audioOS	6 dif	100 dif	wiretap
10.0.0.124	Mercury	Watch	Fit	watchOS	-	17 dif	wiretap/track
10.0.0.155	raspberrypi	rpi	dev	Buster	22	-	ssh
192.168.0.001	Century Link	Router	Gateway	Linux	23,80,433,8085	-	telnet
192.168.0.002	Dionysus	Router	Secondary	Linux	All	-	-
192.168.0.003	raspberrypi	rpi	dev	Buster	22	-	ssh
192.168.0.004	DESKTOP..	WindowsPC	PC	Windows	-	All	-
192.168.0.031	Veritas	MacLaptop	PC	macOS	3000	-	grafana
192.168.0.032	Unknown	?	80:1f:12:c6:cf:77	Linux	22	-	ssh
192.168.0.033	Unknown	HPLaptop	tails	tails live	-	All	-
192.168.0.100	Ares	rpi	motion	Buster	22,3389,8081	-	xrdp,motion
192.168.0.101	Master	rpi	Kubernetes	Buster	22,80,443	-	-
192.168.0.102	Node1	rpi	Kubernetes	Buster	22,80,443	-	-
192.168.0.103	Node2	rpi	Kubernetes	Buster	22,80,443	-	-
192.168.0.104	Node3	rpi	Kubernetes	Buster	22,80,443	-	-
192.168.0.105	Node4	rpi	Kubernetes	Buster	22,80,443	-	-
192.168.0.106	Node5	rpi	Kubernetes	Buster	22,80,443	-	-

Table 1. Compilation of the results from all scans. Purpose column filled from previous knowledge of the network.

identifying the unknown device

Attempting to connect via ssh to the device confirmed that it was not something I was familiar with. I change all of my ssh connections to passwordless key based authentication. This has much better security than the password based authentication running on this device which allowed me to connect and try passwords as well as connect as root to try passwords which I regularly disallow.

researching the OUI

Nmap includes a lookup of the mac addresses OUI, organizationally unique identifier, the first three octets.

```
MAC Address: 80:1F:12:C6:CF:77 (Microchip Technology)
```

binary note

Note that in the decimal notation representing the last octet of our networking addresses 3 digits are required whereas here hexidecimal notation is used and only two digits are required. Both of these represent 8 binary digits and have 256 unique possibilities. Usually indexed at 0 so 0-255 and in the case of networking some addresses are always reserved.

The products section of their website at `https://www.microchip.com/` included embedded architecture, security, smart monitoring and networking devices among others.[4] This prompted the question, could my security device for the lock on my door have started running in unsecure ssh connection on my network as soon as I plugged in the ethernet cable?

unplug that device

Immediately unplugging the device then rescanning confirmed this hypothesis as seen in **Figure 5**.

Figure 5. Rescan after unplugging

CONCLUSIONS

The scans and research prompted some further action. To look more closely into the `4567 tram` port open on the router to the internet I enabled telnet and explored the device. I was curious to see if the port could be closed but didn't find a method. Throughout the research I did find a possible ADSL attack vector for the router:[3]

```
nmap -sS -sV -vv -n -Pn -T5 A.B.C.D -p80 -oG -
```

This attack was summarized, `using nmap to look for an open port 80 on a huge block of IP addresses.`[3] Article, `The weakest link on the network: exploiting ADSL routers to perform cyber-attacks`, published by the IEEE in 2013 discusses their discovery of two 0-day vulnerabilities and expounds the vulnerabilities of ADSL routers.[1] Of course my conscience of the ethical and legal concerns prevented me from attempting any attacks without explicit permission.

the intelligence of smart security devices

The most concerning finding on my local network was the opreation of the security device which controls my locks and that was mandated for all apartment units. Along with a possible network attack vector, I was able to plug and unplug the device because it is not stored in a secure cabinet. I have total access to it. All of the devices in my building

are the same so if one is compromised every unit is compromised. Likely nationwide a huge number of homes would be compromised. Access also gave me the FCC IDs seen in **Figure 6**.

Figure 6. The FCC ID's clearly displayed

fcc id

The FCC ID's listed are `2AAU7-ZB2ZWUS XMR201807EG91NA` and are all matters of public record. The FCC website is pretty terrible to use but using `www.fcc.io` we can search FCC site with a much better UI. The records tell us that the device has a Quectel Wireless chip for the cellular connection as well as a Z-Wave module to connect to the devices.[5] Leaving the cellular aside the records give us two possible frequencies for the radio communication to the locks. In **Figure 7** we can see one of the radio frequencies.[5] It is beyond the scope of this paper but a software defined radio could be used to capture and analize the signals on that frequency.

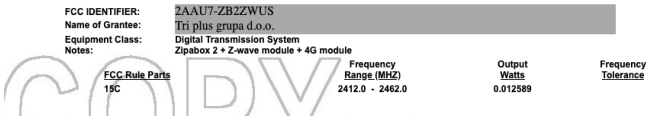

Figure 7. Radio frequency

nmap OS fingerprinting

Many of the devices on my network had operating systems which were unrecognized and nmap asked to submit the fingerprint if the OS is known. I followed the link to try and submit the fingerprints but ran into a bug on the Nmap site. There was no section on the submission form for writing what the OS actually was, **Figure 8**. On the site to submit service types there was the section missing from the OS submission page. I signed up to the dev-mailing list, as suggested by the site for bug reporting, to try and see if the bug was known.[6] I also ran into several issues with Zenmap on Catalina from install, to running, to save permissions and with it crashing when selecting topology.

next steps

In addition to the further avenues for research within the network, next steps would include more testing on these issues and bug reporting to nmap dev-list and issue trackers.

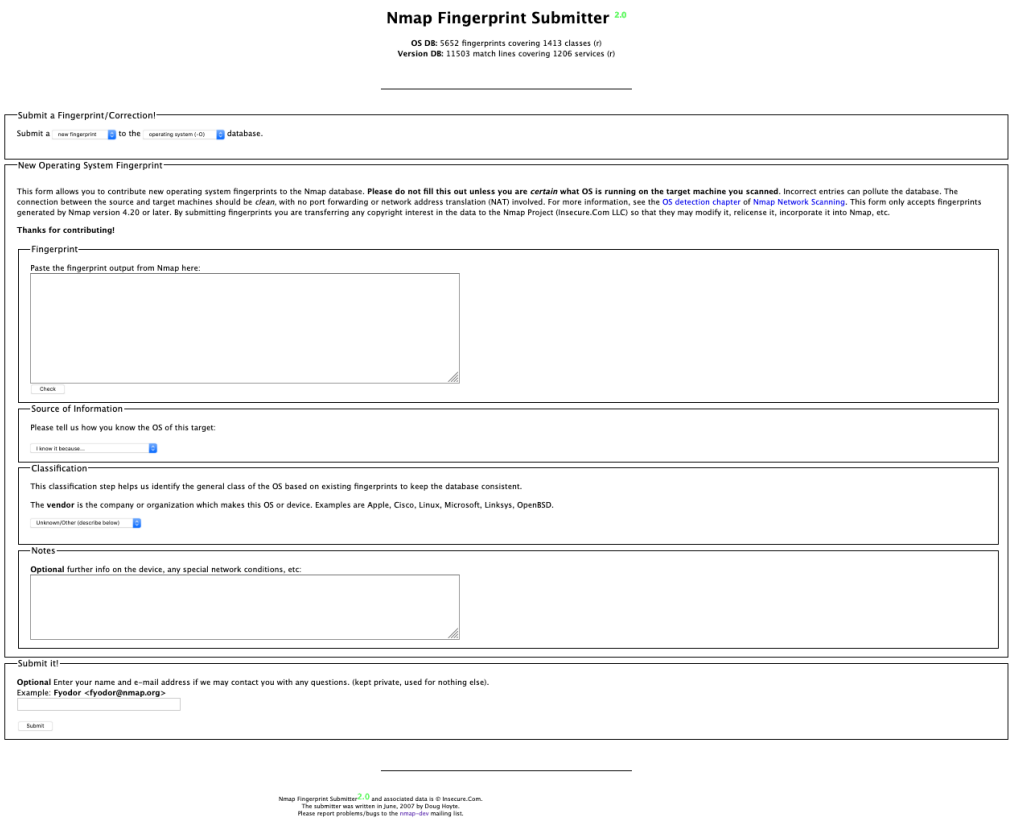

Figure 8. Nmap OS fingerprint submission page missing OS name category

REFERENCES

1. Stasinopoulos, Ntantogian and Xenakis(2013) The weakest link on the network: Exploiting ADSL routers to perform cyber-attacks, *IEEE International Symposium on Signal Processing and Information Technology* 000135-000139. *https://doi.org/10.1109/ISSPIT.2013.6781868*
2. The search engine for The Internet of Things, *https://www.shodan.io/* (accessed June 2020)
3. Technicolor C2100T, *https://charlesreid1.com/wiki/Technicolor_C2100T* (accessed June 2020)
4. Microchip Technology Inc, *https://www.microchip.com/* (accessed June 2020)
5. Federal Communications Commission, *https://www.fcc.gov/* (accessed June 2020)
6. Nmap Fingerprint Subbmitter, *https://nmap.org/cgi-bin/submit.cgi?new-os* (accessed June 2020)

REFLECTION

There were some changes I made to the configuration of my network on the basis of this paper. I realized that I should be turning grafana off when I am not using it until I have it setup with production level safety. I decided that my smart device was better off having less connectivity and that I would not be plugging it into my local network. It was interesting to check what returned from the tails during network scans, which was nothing at all and a mac address which changed on every boot. It was also suprising to see how many services actively run on my HomePod and my appleWatch as well as the HomePod using two IP addresses. I hadn't explored the telnet connection to my CenturyLink device before either and it was a good learning experience. It was interesting to see how different the telnet interfaces are between devices. I'd been meaning to check the protocols on the smart security device and so was glad to get a chance. I think the current version of Z-wave is relatively secure but found some talk about downgrade attacks forcing devices to use older versions. I also used this as an opportunity to try using LaTeX for the first time, which I borderline came to regret as I didn't realize before hand that there is come complexity to it.