Local Network Audit and Exploration with Nmap

Larsen Close

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

ABSTRACT
Extensive Nmap scanning is used to fingerprint a local network with clear instruction given for replication and key areas of interest are identified. Anomalies, security concerns and unexpected results are discussed and further pursued. A detailed account of supplemental tools used and actions prompted by the findings is given before presenting conclusions and a reflection.

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. As the Internet Of Things expands and our world increasingly digitizes so does the danger to our privacy, security and safety. These devices warrant attention, as believing someone else is protecting our privacy, security and safety is negligently naive. The steps clearly demonstrated in this paper are steps we should all take regularly for our own sake and for the sake of our loved ones.

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 abnormalities, 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 property owner) I connected this device to the network for the first time. The device has been running on cellular but has an ethernet option as well.

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

let shodan.io scan for you
To check on the networks public IP address and look 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 table below to organize my findings, visualize them and decide where to increase focus. Host data from the scans was combined with DHCP Reservations from the routers then supplemented with previous knowledge where applicable.

METHODS AND MEASUREMENT
To begin we need to know which IP addresses 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 a ping scan quickly lists 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 without sudo resulted in 1 host and 9 hosts. To ping scan both networks:

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

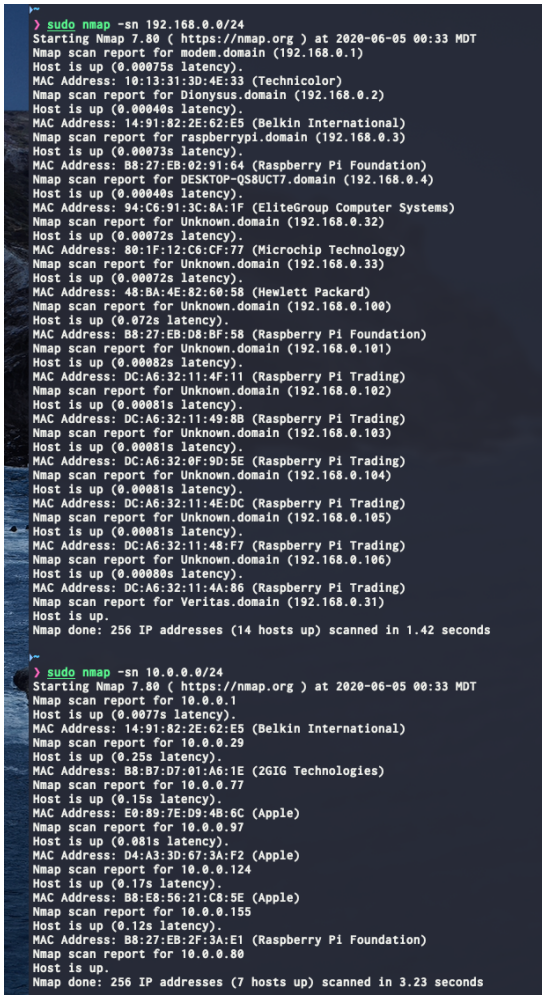

Figure 2. Ping scans returning 14 and 7 hosts.

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

flags for more thorough scans

After completing the ping scan, using nmap with the flag `-A` will initiate an aggressive scan. Including `-v` increases the verbosity of the results and `-T5` turns the 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 results and create an overview that could be visualized. The more thorough 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. For example:

```
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 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 and set up a monitor for the address.[2] An easy what to determine your public IP address is:

```
curl ifconfig.me
```

RESULTS

Shodan.io allowed us to determine that the router has an open outward facing port running http with authentication on 4567.[2] **Figure 3** shows a screen shot from their web service.

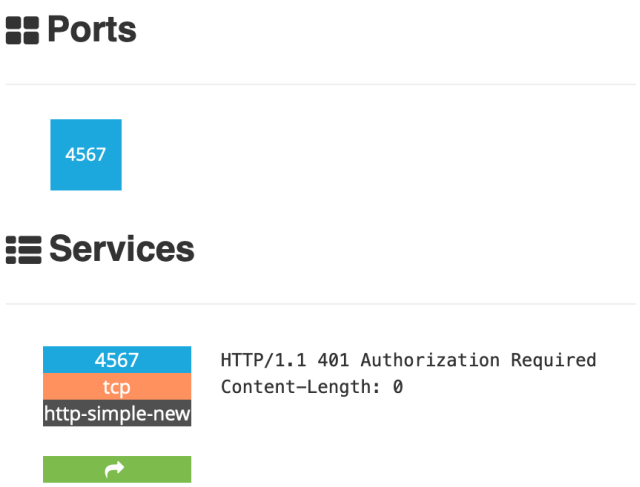

Figure 3. From website https://www.shodan.io/ often called "the most dangerous search engine in the world"

compiling results

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. My HomePod has two IP addresses on the same network. After logging into and checking the router it appears both are running on the same 5G

WiFi network. While going through the results I was familiar with everything except for host 192.168.0.32 which was running ssh on port 22. I checked the routers DCHP Reservations and found no information there either. Scans showed the mac address `80:1F:12:C6:CF:77` as well as some additional information. Seen here using a targeted aggressive scan of just this device: `sudo nmap -T5 -A 192.168.0.32`

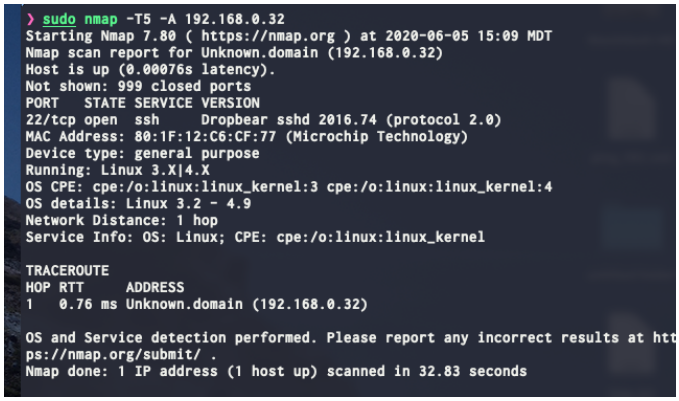

Figure 4. Resulting scan of unfamiliar device

	Hostname	Device	Purpose	OS	Open	Filtered	Services
10.0.0.1	Dionysus	Router	Gateway	Linux	53,80,443,10000	-	ssl,http
10.0.0.29	HD100	Webcam	Security	-	554,49152	-	rtsp,UPnP
10.0.0.77	Daemon	HomePod	Music	audioOS	62078	24 dif	wiretap
10.0.0.80	Veritas	MacLaptop	PC	macOS	3000	-	grafana
10.0.0.97	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
192.168.0.1	Century Link	Router	Gateway	Linux	23,80,433,8085	-	telnet
192.168.0.2	Dionysus	Router	Secondary	Linux	All	-	-
192.168.0.3	raspberrypi	rpi	dev	Buster	22	-	ssh
192.168.0.4	DESKTOP..	WindowsPC	PC	Windows	-	All	-
192.168.0.31	Veritas	MacLaptop	PC	macOS	3000	-	grafana
192.168.0.32	Unknown	?	80:1f:12:c6:cf:77	Linux	22	-	ssh
192.168.0.33	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. Which is more secure than the password based authen-

tication 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. Rather than attempting to brute force the password I followed the mac address.

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

> In the decimal notation representing the last octet of our networking addresses 3 digits are required whereas here hexadecimal 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 reserved, such as broadcast on 255.

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

yes - unplug that device

Immediately unplugging the device then rescanning confirmed this hypothesis, see **Figure 5**.

Figure 5. Rescan after unplugging

CONCLUSIONS

The foregoing results prompted further action. To look more closely into `port 4567` running `tram` open to the entire internet I enabled telnet to explore the device. I was curious to see if the port could be closed but did not find a method. Important to note here that telnet can usually be easily enabled by logging into your routers admin page at its IP address in a browser. Do not forget to disable telnet after finishing with it. While researching Century Links C2100T router I did find an ADSL attack vector using nmap listed:[3]

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

The attack was summarized - `using nmap to look for an open port 80 on a huge block of IP addresses.`[3] However, I found nothing related nor any further information on exploit-db or google. I did find a generally related article, `The weakest link on the network: exploiting ADSL routers to perform cyber-attacks`, published by the IEEE in 2013 discussing the authors discovery of two 0-day vulnerabilities and expounding upon the vulnerabilities of ADSL routers.[1] Of course my conscience of the ethical and legal concerns prevented me from attempting any thing like this myself without explicit permission.

not-so smart devices

The most concerning finding was the smart device which controls my lock and was mandated for every unit in the building. 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. 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. Here in **Figure 6** are the FCC IDs for the device.

Figure 6. The FCC ID's clearly displayed

fcc id

The FCC ID's are `2AAU7-ZB2ZWUS XMR201807EG91NA` the records for which are all public. 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, with a list of the frequencies, and a Z-Wave module for the radio connection to the lock.[5] The records give us two possible frequencies for the Z-wave. One of which is shown in **Figure 7**. It is beyond the scope of this paper but a software defined radio could be used to capture and analyze the signals on that frequency.

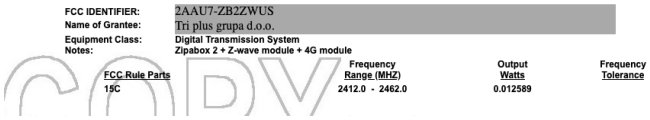

Figure 7. Radio frequency

nmap OS fingerprinting

Many of the devices on my network have unrecognized operating systems, for these cases nmap asks we submit of the fingerprint if the OS is known. I tried to submit the fingerprints but ran into a bug on the site. There is no section on the submission form for writing what the OS actually is, **Figure 8**. Alternatively the unknown services submission site has the section missing from the OS's. I signed up to the dev-mailing list, as suggested by the site for bug reporting, and will try and see if the bug is known.[6] I also ran into several issues with Zenmap on Catalina, issues installing, running, saving scans and crashing selecting topology.

next steps

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

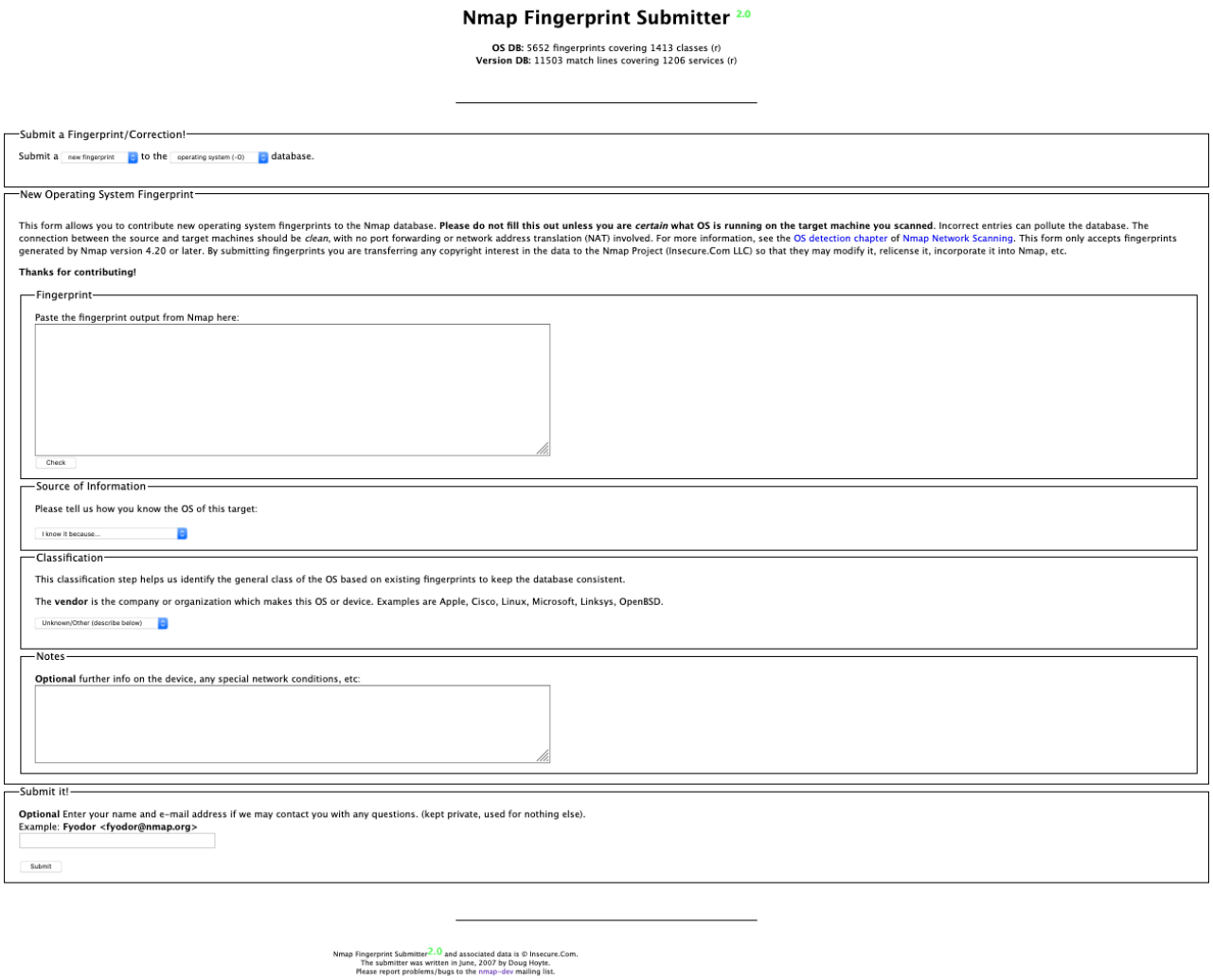

Figure 8. Nmap OS fingerprint submission page missing OS name category

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 Submitter, *https://nmap.org/cgi-bin/submit.cgi?new-os* (accessed June 2020)

REFLECTION

Throughout this exploration there was constant learning, experienced gained and improvement of my network configuration. I realized that I should be turning grafana off when I am not using it until I have it setup with production level security. I decided that my smart device was better off having less connectivity and that I would not be plugging it into my local network. With more time I would have looked more closely into exactly what was and wasn't returned from the different nmap scans including UDP scans and see if I could find more information locally using options for

firewall and IDS evasion. The questions I brought to the paper were all answered and I was left with more questions replacing them.

tails questions

I found out what is returned from tails during network scans, nothing at all and a mac address which automatically changes on every boot. I had heard that the issue of what to do with local network traffic on tails had still not been completely resolved but I take it that is mostly referring to what the machine running tails wants to do on the local network.

surprises

It was astounding to see how many services actively run on my HomePod and my appleWatch. Also strange to learn that the HomePod regularly uses two IP addresses. I just used the UncOver zero-day on iOS from a couple weeks back to jailbreak an old iPad. My hope was to find something like Little Snitch for iOS but it is pretty neat to have root access to my iPad and be able to ssh to it. Maybe an approach like this would work for the HomePod to continually see just what is happening with all those services.

telnet

I hadn't ever explored the telnet connection to my CenturyLink device. It was interesting to see how different the telnet interfaces are between devices. I found methods for closing the open 4567 port running tram on other manufacturers routers but almost none of the commands were consistent or even available between devices.

z-wave

I'd been meaning to research the smart 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 have some other Z-wave devices here so I'm curious to see if I can get them to connect. I have a hackrf-one software defined radio and would love to do security research on Z-wave but time is always a limiting factor as well as the strict and tricky legality concerns, especially when dealing with radio. I found a 2020 paper using the hackrf-one to do a denial of service attack on Z-wave versions s0-s2 but haven't seen much beyond that.

latex

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 there was some learning curve. I especially struggled towards the end figuring out formatting for the bibliography and citations while tired and needing to finish. This probably more than anything else led to the paper taking a lot of time but also added to what was gained from the assignment and gave me incredible control in formatting.