UNIVERSITY OF ZAGREB
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

SEMINAR

Nmap: Scanning the Internet

Marin Koštić

Mentor: *Doc. dr. sc. Predrag Pale*

Zagreb, January 2018.

CONTENTS

1. Introduction

Nmap ("Network Mapper") is a free and open source utility for network discovery and security auditing. Many systems and network administrators also find it useful for tasks such as network inventory, managing service upgrade schedules, and monitoring host and service uptime. In this seminar I will talk about how Gordon Lyon (also known by his pseudonym Fyodor Vaskovich) has taken Nmap to a whole new level by scanning millions of Internet hosts as part of the Worldscan project. Fyodor has presented his work on DEFCON 16 conference where he talked about how he used Nmap to determine possible bugs and flaws so he could further improve upon it and also he demonstrated practical advices on how to efficiently use Nmap. In the following pages I will show examples on how to use Nmap for host discovery and port scanning as the main topic, but I will also include additional topics about detecting and subverting firewall and intrusion detection systems.

2. Scanning the Internet

2.1. Scan Challenges

Before making an scan of such scale, first question that comes to mind is what are the actual IP addresses that you want to scan. There are couple of options that one can use to determine IP addresses to scan, one could take BGP routing tables. BGP stands for Border Gateway Protocol, it is a standardized exterior gateway protocol designed to exchange routing and reachability information among autonomous systems on the Internet. Another interesting option is using DNS zone files, these are text files that describe a DNS zone. A DNS zone is a subset, often a single domain, of a hierarchical domain name structure of the DNS. The zone file contains mapping between domain names and IP addresses and other resources. But what in his talk Fyodor used Nmap's own random IP generator and demonstrated how one can use this Nmap tool to get a list of IP addresses without actually scanning the machines.

Listing 2.1: Using Nmap to generate random IP addresses

```
nmap -iR 25200000 -sL -n | grep "not␣scanned"
| awk '{print $2}' | sort -n |
uniq >! tp; head -25000000 tp >! 25M-IPs;
rm tp
```

Next question, when you obtain IP addresses to scan, is what source do you wanna use for the actual scan. It is important to understand when you get a list of 25 million IP addresses and you want to preform such a scan as Fyodor did it rises a legal concern. If you preform this scan from your home using yours ISP network you can expect to be denied access to the Internet and legal charges could be pressed upon you. So when Fyodor talked about this concern he made a joke about how he thought to use his neighbours open wireless access point. Although this is just a joke, if you look at it from your neighbours point of view having an open wireless access point is a danger and could be exploited using Nmap scans from anyone with little knowledge on how to perform scans. What he actually did when performing such big scan is that he contacted his ISP and explained what he was doing and why, this is example of best practices that any network administrator should have when using

Nmap for a costumer is to explain and ask for permission before executing network scans. The final remark that he makes about large scale scans is that when performing such scans one must be aware of the fact that scanning the Internet is long and hard work and it can be disheartening. He listed an example of an UDP scan with 65 thousand ports and 2048 hosts in a group from his original IP address list.

Listing 2.2: UDP Scan performance example

```
– Stats: 93:57:40 elapsed; 254868 hosts
 completed (2048 up), 2048 undergoing UDP Scan
UDP Scan Timing: About 11.34\%done; ETC:
 03:21 (−688:−41:−48 remaining)
```

As it can be seen from the example this was a scan of such scale that time estimation resulted in an integer overflow, one can think are such scans even worth doing.

2.2. Host Discovery

In the previous section we described how to use Nmap to random generate a list of IP addresses now we have to tackle a task of host discovery. This is one of the first steps in any network reconnaissance, what it means is to reduce a set of known IP addresses into a list of active or somehow interesting hosts. There are many methods to use within Nmap for host discovery, most common and by default active when using Nmap is Echo request and TCP ACK to port 80. Although these might be sufficient in some cases, when doing a large scale scan as one Fyodor described this would not be a good solution. Maybe this type of host discovery would be efficient in the late 90's when most of hosts did not have strict iptables rules and weren't behind a firewall. But today Internet is quite a dangerous place to have open ports and many big companies have a good practice to block scans that use Echo request and TCP ACK on port 80. But there are other different methods that are available in Nmap for host discovery and can be used in such cases.

TCP Host Discovery methods (-PS, -PA):

- SYN packet discovery (-PS) best against stateful firewall. The SYN flag suggests to the remote system that you are attempting to establish a connection. Normally the destination port will be closed, and a RST (reset) packet sent back. If the port happens to be open, the target will take the second step of a TCP three-way-handshake by responding with a SYN/ACK TCP packet.
- ACK packet discovery (-PA) best against stateless firewall. The TCP ACK ping is quite similar to the just-discussed SYN ping. The difference, as you could likely

guess, is that the TCP ACK flag is set instead of the SYN flag. Such an ACK packet purports to be acknowledging data over an established TCP connection, but no such connection exists. So remote hosts should always respond with a RST packet, disclosing their existence in the process.

As described above we see that different TCP port scanning methods have different scanning results so a best practice would be to use both methods. Now what one could be wondering is what are the actual ports that are good to scan with these methods. It would be futile to scan all ports since we know that there are 65 thousand available ports, and such a scan would just be too long to perform on a big number of hosts. What Fyodor suggests in his talk is the fallowing list of ports, what is important to note is that this information was a result of scans that he did on hosts that were heavily firewalled.
Top TCP Host Discovery Ports:

- 80/http
- 25/smtp
- 22/ssh
- 443/https
- 21/ftp
- 112/auth
- 23/telnet
- 53/domain
- 554/rtsp
- 3389/ms-term-server

Next strategy is to use the UDP discovery method (-PU) for port scanning. When using UDP port scanning the goal is to find closed ports since they are more likely to respond, because an open UDP port won't even respond to an UDP probe. But an close port will send an Port Unreachable packet which is more than enough information to find out whether a host is up or not. A best practice is to use a high number port when using this scan and to include port 53, this port is interesting and worthwhile due to firewall exceptions for DNS.

2.3. Port Scanning

3. Conclusion

Conclusion.

4. Summary

Summary.