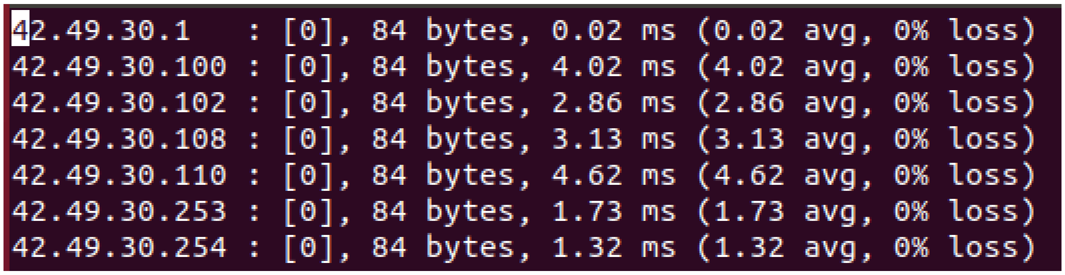
Host Discovery

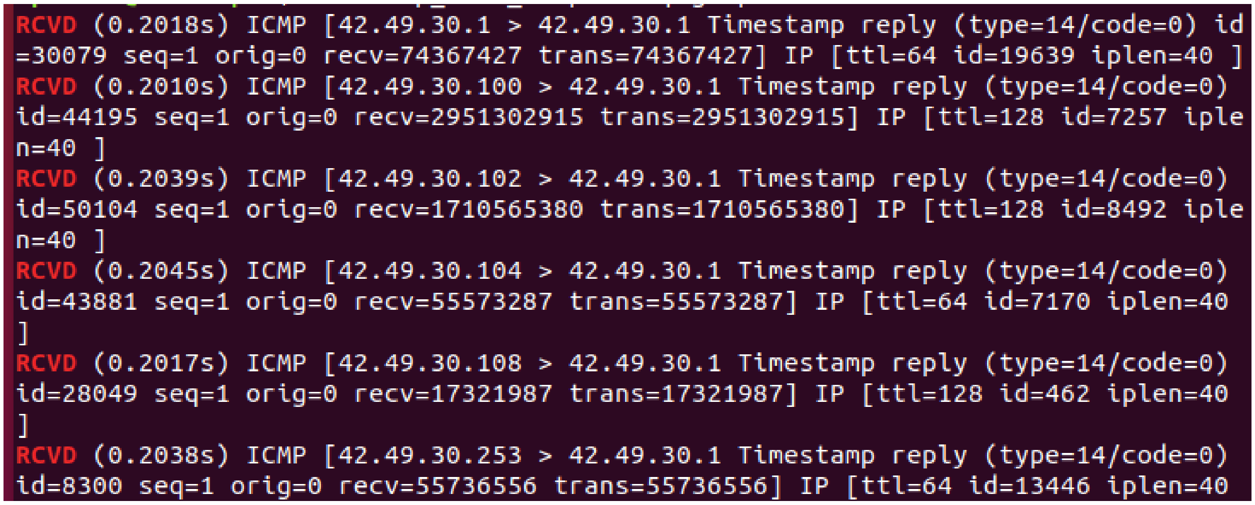
1. Screenshot of hosts responding to fping



```
42.49.30.1    : [0], 84 bytes, 0.02 ms (0.02 avg, 0% loss)
42.49.30.100 : [0], 84 bytes, 4.02 ms (4.02 avg, 0% loss)
42.49.30.102 : [0], 84 bytes, 2.86 ms (2.86 avg, 0% loss)
42.49.30.108 : [0], 84 bytes, 3.13 ms (3.13 avg, 0% loss)
42.49.30.110 : [0], 84 bytes, 4.62 ms (4.62 avg, 0% loss)
42.49.30.253 : [0], 84 bytes, 1.73 ms (1.73 avg, 0% loss)
42.49.30.254 : [0], 84 bytes, 1.32 ms (1.32 avg, 0% loss)
```

2. Did we observe more hosts than with the standard ping? List any differences. Provide a screenshot of the output to support your answer. (ICMP Time)

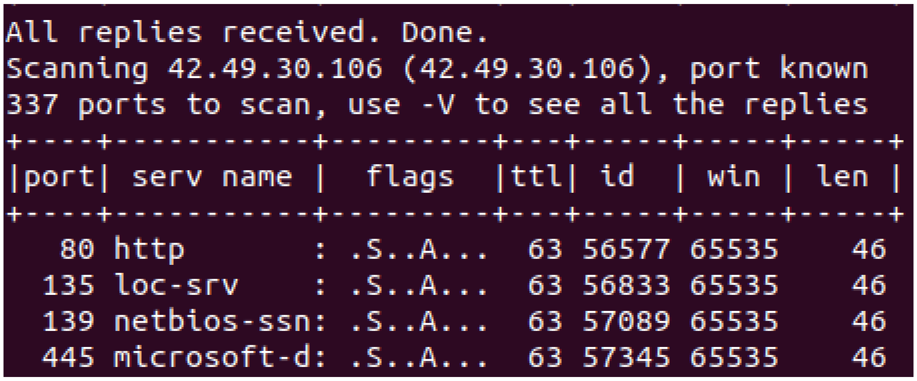
Yes, with the timestamp requests we were able to see the X.X.X.104 responds as well.



```
RCVD (0.2018s) ICMP [42.49.30.1 > 42.49.30.1 Timestamp reply (type=14/code=0) id
=30079 seq=1 orig=0 recv=74367427 trans=74367427] IP [ttl=64 id=19639 iplen=40 ]
RCVD (0.2010s) ICMP [42.49.30.100 > 42.49.30.1 Timestamp reply (type=14/code=0)
id=44195 seq=1 orig=0 recv=2951302915 trans=2951302915] IP [ttl=128 id=7257 iple
n=40 ]
RCVD (0.2039s) ICMP [42.49.30.102 > 42.49.30.1 Timestamp reply (type=14/code=0)
id=50104 seq=1 orig=0 recv=1710565380 trans=1710565380] IP [ttl=128 id=8492 iple
n=40 ]
RCVD (0.2045s) ICMP [42.49.30.104 > 42.49.30.1 Timestamp reply (type=14/code=0)
id=43881 seq=1 orig=0 recv=55573287 trans=55573287] IP [ttl=64 id=7170 iplen=40
]
RCVD (0.2017s) ICMP [42.49.30.108 > 42.49.30.1 Timestamp reply (type=14/code=0)
id=28049 seq=1 orig=0 recv=17321987 trans=17321987] IP [ttl=128 id=462 iplen=40
]
RCVD (0.2038s) ICMP [42.49.30.253 > 42.49.30.1 Timestamp reply (type=14/code=0)
id=8300 seq=1 orig=0 recv=55736556 trans=55736556] IP [ttl=64 id=13446 iplen=40
```

3. Evaluate the responses with hping3

   a. Screenshot of the missing host's response with hping3



```
All replies received. Done.
Scanning 42.49.30.106 (42.49.30.106), port known
337 ports to scan, use -V to see all the replies
+----+-----------+---------+---+-----+-----+-----+
|port| serv name |  flags  |ttl| id  | win | len |
+----+-----------+---------+---+-----+-----+-----+
   80 http       : .S..A...  63 56577 65535    46
  135 loc-srv    : .S..A...  63 56833 65535    46
  139 netbios-ssn: .S..A...  63 57089 65535    46
  445 microsoft-d: .S..A...  63 57345 65535    46
```

   b. What is the IP of the stealthy host?

      42.49.30.106

   c. What ports are open on this host?

      80, 135, 139, and 445

4. Examine the wireshark output and answer the following questions

   a. What kind of scans are taking place during nmap?

      On wireshark, I found Echo requests/replies, ARP, and SYN scans. I think in lecture it was said that default does timestamp and ACK scans. Maybe they only occur if the others get no responses, such as using timestamp to find hosts that are up if echo gets no replies.

   b. Why might performing an nmap scan not be a good idea?

      This creates a lot of traffic that routes directly back to your ip that is scanning the ip (range) that you entered. As well, it is well known that you are using nmap to scan since its defaults do specific types of tests to find up machines and open ports.

Ports and Services

5. Submit your finished “OS guess” table in the lab report.
    a. List reasoning for each OS guess made, whether in the table itself or underneath.
    b. Did any of your guesses change from OS guess (1) to OS guess (2)? List updated Reasoning

Host	Open Ports	Services	OS guess 1	Reasoning
42.49.30.100	135, 139, 445	msrpc, netbios-ssn, microsoft-ds	Windows 7	Could be any NT based Windows machine after 2000, since I think the DS service is used for Active Directory and SMB.
42.49.30.102	7, 9, 13, 17, 19, 25, 53, 80, 135, 139, 443, 445, 515, 1025, 1027, 1030, 1033, 1035, 1755, 3372, 3389, 6666	echo, discard, daytime, qotd, chargen, smtp, domain, http, msrpc, netbios-ssn, https, microsoft-ds, printer, NFS-or-IIS, IIS, iad1, netinfo, multidropper, wms, msdtc, ms-wbt-server, irc	Windows Server 2003	This has many services open that used by Windows so I am taking the guess of it being a Windows Server. Version is simply a guess.
42.49.30.104	22, 80, 443	ssh, http, https	Ubuntu 18	ssh is open so a base guess is one of the most widely used Unix-like systems. Ubuntu.
42.49.30.106	80, 135, 139, 445	http, msrpc, netbios-ssn, microsoft-ds	Windows Server 2008	By the ports, it is a newer NT with a web server up so taking a guess of a random server edition with windows.
42.49.30.108	135, 139, 445	msrpc, netbios-ssn, microsoft-ds	Windows XP	Just picking a more less random newer NT system as it could be any one of them, if my thinking is correct.
42.49.30.110	135, 139	msrpc, netbios-ssn	Windows NT X.x	This lacks port directory services but has MSRPC. So it should be an early NT (before 2000).
42.49.30.253	22	ssh	Freebsd X.x	This one is to spice it up a little. From just the open ports of 22, it could be any unix based system, so why not FreeBSD. (My favorite) Most likely Ubuntu or RHEL, though.

Host	Open Ports with different services	Changed Services	OS guess 2	Reasoning	Actual OS
42.49.30.100	Same	Same	Windows XP	Version scan return XP as the Windows version	XP or 2003
42.49.30.102	1025, 1027, 1030, 1033, 1035, 3389, 6666	msrpc, msrpc, msrpc, msrpc, msrpc, tcpwrapper, nsunicast	Windows 2000	Version scan return 2000 as the Windows version	2000, XP or Me
42.49.30.104	Same	Same	Ubuntu 12	Scan said Ubuntu for OS and has OpenSSH v6.0p1 and Apache v2.2.22. Those versions were both released in 2012.	Linux 2.6.32 - 3.10 (which is like 2010 - 2013)
42.49.30.106	139	ssh	Freebsd 7	Given Freebsd for OS that has OpenSSH v4.5 and Apache v2.2.9. OpenSSH was released in 2006 and Apache in 2008. Freebsd 7 wasn't released until 2008.	FreeBSD 6, JUNOS 10/12, m0n0wall, or Netasq
42.49.30.108	Same	Same	Windows 7	Directory Services is given as a version between Windows 7 and 10.	Windows 10
42.49.30.110	Same	Same	Windows NT X.x	This confirmed what I said before about it being an old NT as the services match the ports.	Windows NT 4.0
42.49.30.253	Same	Same	Ubuntu 16	Given Ubuntu as the OS and has OpenSSH v7.2, which was released February 29, 2016.	Linux 3.2 - 4.8 (which is like 2012 - 2017)

6. For each host, were either of your guesses accurate in guessing the operating system? Why might it be valuable to determine operating systems without performing an nmap scan?

I was actually close on my guess for the X.X.X.110 system. For the rest, I was either off for a few generations off or had the wrong OS altogether.

It might be useful to guess the OS without scanning since you can take a well educated guess of whats on their network (especially with previous recon). From there, you can find out what are the potential vulnerabilities that are on their systems. As well, you don't have to run the risk of getting detected by IDS with more scans. However, you do run the risk of being wrong about your guess and your target could use nonstandard ports as means of trickery, which messes with your ability to guess.

7. List 3 ways and/or options that you might use nmap such that no alarms might be raised.

-S spoof IP: This will make your IP look like that of another computer / network. Not sure on the usefulness of this since the scan responses go to the IP you're spoofing.
-g spoof port: This sets the port that the packets sent are coming from on your computer. For some reason, it may be set to allow in from well known ports like DNS/53.
-f fragmentation: This breaks requests into many smaller packets. The idea seems to be that some firewalls/IDS systems may not queue and reassemble fragmented packets, thus allowing you to bypass the checks.

Additional Ones I found notable:
--max-rate: This lets you limit the rate at which you send requests to the device(s) you are scanning. Not sure if this would let you slip by IDS, but it could make it look less important to a person if something is happening slowly. As well, I know that at least on the PF firewall that I use I am able to set up a system that blocks an IP if it tries to connect too many times per X seconds. So you can bypass that if you know the rate.
-D decoy IP: This makes it look as if there are many additional IPs that are scanning concurrently to your actual IP. Helps hide your actual IP as the one that is scanning. Can be routed back to your machine though.
-sI zombie scan: This makes another machine do the port scanning for you. This makes it harder for your device to be seen as the attacker.

Vulnerabilities

8. List each vulnerability that you find.
    a. Buffer Overflow in Active Directory (remote exec)
        i. What hosts are they effective against?
            1. Windows XP and 2000 so X.X.X.(100 and 102)
        ii. What resources did you use to find this vulnerability?
            1. https://nvd.nist.gov/vuln/detail/CVE-2003-0533
            2. https://www.exploit-db.com/exploits/295
        iii. What exploits may be used?
            1. There is the exploit from exploit-db (which I believe opens up port 4444)
            2. The Sasser worm: https://en.wikipedia.org/wiki/Sasser_(computer_worm) utilized this vulnerability

    b. OpenSSH uses blowfish hashing when user does not exist
        i. What hosts are they effective against?
            1. This lets us find out the users on machines running OpenSSH before v7.3
                a. Meaning X.X.X.(104, 106, and 253) are vulnerable
        ii. What resources did you use to find this vulnerability?
            1. https://nvd.nist.gov/vuln/detail/CVE-2016-6210
            2. https://www.rapid7.com/db/modules/auxiliary/scanner/ssh/ssh_enumusers/
            3. https://www.exploit-db.com/exploits/40136
        iii. What exploits may be used?
            1. #2 is a metasploit module
            2. #3 is shell program
                a. Both can find the valid users of the remote machine

    c. RPC buffer overflow (remote exec)
        i. What hosts are they effective against?
            1. Windows NT 4.0 and 2000 so X.X.X.(102 and 110)
        ii. What resources did you use to find this vulnerability?
            1. https://nvd.nist.gov/vuln/detail/CVE-2003-0003
            2. https://www.exploit-db.com/exploits/5
        iii. What exploits may be used?
            1. #2 exploit opens up port 5151 and allows you to run cmd’s remotely