CS3235 - Laboratory #1 for weeks 4, 5 and 6
(Aug 29-Sept 16, 2005)

This laboratory may be done in a group. The size of your group is up to you, and could be 1,2,3 or 4 group members. The laboratory is not structured, and is graded based on accomplishment. When you feel comfortable using `nmap` and `ethereal` on the machines in the laboratory, then ask Pradeep to assess you. He will ask you a few questions, get you to use `nmap` and `ethereal`, and mark the assessment sheet. At the same time, you must hand him your lab writeup for step 1, and the review questions. For this first laboratory, the assessment should be very easy and fast.

1 The lab

Unfortunately, we have not completed setting up the laboratory, but when you first enter the labs, the first 10 machines are all dual-boot Linux/windows, divided into two linked networks (the left and right sides). The machines on the front desks of the room are connected to a promiscuous port, and can thus see all the traffic between the machines on each network.

At the time of writing this, the IP addresses and so on are not set, so you will have to discover this information for yourself.

2 Using nmap

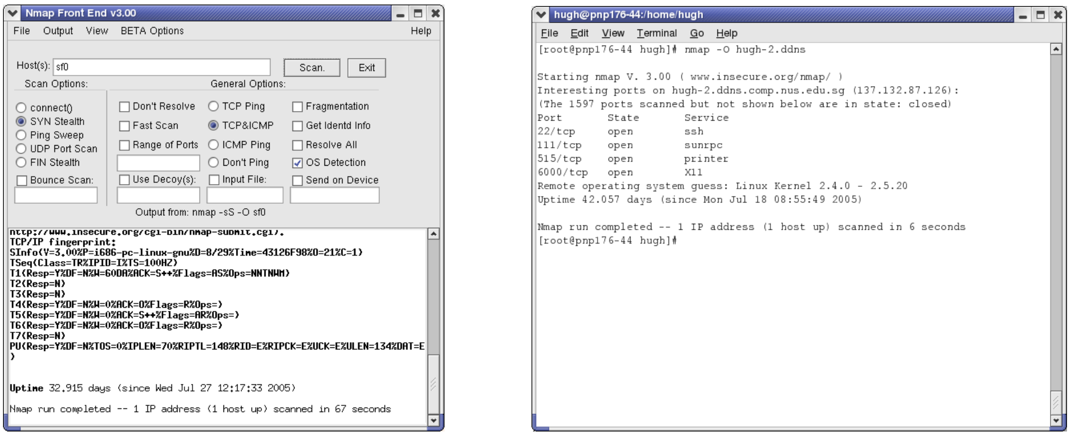
This lab will familiarize you with `nmap`, a nugget in a network hacker's toolchest. You can find `nmap` at http://www.insecure.org/nmap/.

Those of you who are network inclined will come to appreciate the power, efficiency, and versatility of this tool. A prettied up version of the `nmap` paper is available at
http://www.comp.nus.edu.sg/~cs3235/2005-semesterI/nmap-fingerprinting-article.mr.html.

The program `nmap`, short for "network mapper" can be used to probe a single computer or a whole network (i.e., all the computers with addresses in a specified consecutive range) for services that are running on them. While performing its probe, `nmap` can take care to avoid being detected. It can also make a very good guess about the architecture and operating system of the computer that is probed (for e.g., how would you determine the operating system running on `129.128.1.2`?).

Start by reading the nmap paper and try to understand as much as you can. It does require some bit of understanding of TCP but since we are not expecting you to be TCP/IP wizards, a rough understanding of how nmap achieves its goals is adequate. Find your networking textbooks and re-familiarize yourself with an IP datgram header and the TCP header. The code bits from the TCP header are important to understand how nmap can be stealthy. Install nmap and read its man page as well to understand its different command line options. Then try the following steps.

2.1 Step 1



Host(s): sf0

○ connect()
● SYN Stealth
○ Ping Sweep
○ UDP Port Scan
○ FIN Stealth
□ Bounce Scan:

□ Don't Resolve
□ Fast Scan
□ Range of Ports
□ Use Decoy(s):

○ TCP Ping
● TCP&ICMP
○ ICMP Ping
○ Don't Ping
□ Input File:

□ Fragmentation
□ Get Identd Info
□ Resolve All
☑ OS Detection
□ Send on Device

Output from: nmap -sS -O sf0

```
http://www.insecure.org/cgi-bin/nmap-submit.cgi/.
TCP/IP fingerprint:
SInfo(V=3.00%P=i686-pc-linux-gnu%D=8/29%Time=43126F98%O=21%C=1)
TSeq(Class=TR%IPID=I%TS=100HZ)
T1(Resp=Y%DF=N%W=60DA%ACK=S++%Flags=AS%Ops=NNTNWM)
T2(Resp=N)
T3(Resp=N)
T4(Resp=Y%DF=N%W=0%ACK=O%Flags=R%Ops=)
T5(Resp=Y%DF=N%W=0%ACK=S++%Flags=AR%Ops=)
T6(Resp=Y%DF=N%W=0%ACK=O%Flags=R%Ops=)
T7(Resp=N)
PU(Resp=Y%DF=N%TOS=0%IPLEN=70%RIPTL=148%RID=E%RIPCK=E%UCK=E%ULEN=134%DAT=E
)


Uptime 32.915 days (since Wed Jul 27 12:17:33 2005)

Nmap run completed -- 1 IP address (1 host up) scanned in 67 seconds
```

```
[root@pnp176-44 hugh]# nmap -O hugh-2.ddns

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on hugh-2.ddns.comp.nus.edu.sg (137.132.87.126):
(The 1597 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
111/tcp    open        sunrpc
515/tcp    open        printer
6000/tcp   open        X11
Remote operating system guess: Linux Kernel 2.4.0 - 2.5.20
Uptime 42.057 days (since Mon Jul 18 08:55:49 2005)

Nmap run completed -- 1 IP address (1 host up) scanned in 6 seconds
[root@pnp176-44 hugh]#
```

Figure 1: The program nmap on Linux, GUI and command line.

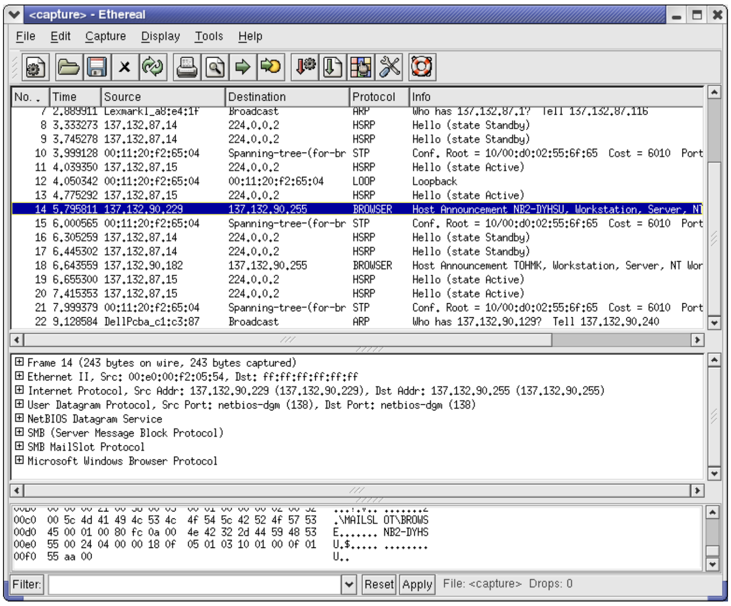
Use both the GUI and command line versions of nmap.

Use nmap to scan all the machines in the lab to determine what ports are open (try both connect scanning and stealth scanning). Record the time it took to find and scan all the machines in each case. Run the scan against a specific machine. What is the average time to scan one machine? Use nmap to OS fingerprint all the machines in the lab. What fraction of the machines did nmap classify correctly? Using the information from the previous two scans, identify vulnerable network services that can be broken into. Hint: once you know what services are running on which operating systems, you can look for publicly available exploits against this combination. See how resourceful you are in finding these exploits.

Write up and tabulate your results - you will have to give them to Pradeep.

2.2 Step 2

While OS fingerprinting a machine, use ethereal to capture the packet flow between the two machines. This can be done by running ethereal on the target machine itself, or alternatively by running it on the machines in the front of the room (These machines can *see* all the network traffic to and from the machines behind them). Can you identify from the ethereal trace whether the machine is being scanned by nmap? Hint: Are there funny packets that nmap uses to do its fingerprinting that can be manually identified in a packet trace? Are there suddenly too many connections to the target machine from a single machine?



```
No. .  Time     Source              Destination          Protocol  Info
   7 2.889911 LexmarkI_a8:e4:1f   Broadcast            ARP       Who has 137.132.87.1?  Tell 137.132.87.116
   8 3.333273 137.132.87.14       224.0.0.2            HSRP      Hello (state Standby)
   9 3.745278 137.132.87.14       224.0.0.2            HSRP      Hello (state Standby)
  10 3.999128 00:11:20:f2:65:04   Spanning-tree-(for-br STP      Conf. Root = 10/00:d0:02:55:6f:65  Cost = 6010  Port
  11 4.039350 137.132.87.15       224.0.0.2            HSRP      Hello (state Active)
  12 4.050342 00:11:20:f2:65:04   00:11:20:f2:65:04    LOOP      Loopback
  13 4.775292 137.132.87.15       224.0.0.2            HSRP      Hello (state Active)
  14 5.795811 137.132.90.229      137.132.90.255       BROWSER   Host Announcement NB2-DYHSU, Workstation, Server, NT
  15 6.000565 00:11:20:f2:65:04   Spanning-tree-(for-br STP      Conf. Root = 10/00:d0:02:55:6f:65  Cost = 6010  Port
  16 6.305259 137.132.87.14       224.0.0.2            HSRP      Hello (state Standby)
  17 6.445302 137.132.87.14       224.0.0.2            HSRP      Hello (state Standby)
  18 6.643559 137.132.90.182      137.132.90.255       BROWSER   Host Announcement TOHMK, Workstation, Server, NT Wor
  19 6.655300 137.132.87.15       224.0.0.2            HSRP      Hello (state Active)
  20 7.415353 137.132.87.15       224.0.0.2            HSRP      Hello (state Active)
  21 7.999379 00:11:20:f2:65:04   Spanning-tree-(for-br STP      Conf. Root = 10/00:d0:02:55:6f:65  Cost = 6010  Port
  22 9.128584 DellPcba_c1:c3:87   Broadcast            ARP       Who has 137.132.90.129?  Tell 137.132.90.240
```

```
⊞ Frame 14 (243 bytes on wire, 243 bytes captured)
⊞ Ethernet II, Src: 00:e0:00:f2:05:54, Dst: ff:ff:ff:ff:ff:ff
⊞ Internet Protocol, Src Addr: 137.132.90.229 (137.132.90.229), Dst Addr: 137.132.90.255 (137.132.90.255)
⊞ User Datagram Protocol, Src Port: netbios-dgm (138), Dst Port: netbios-dgm (138)
⊞ NetBIOS Datagram Service
⊞ SMB (Server Message Block Protocol)
⊞ SMB MailSlot Protocol
⊞ Microsoft Windows Browser Protocol
```

```
00c0  00 5c 4d 41 49 4c 53 4c  4f 54 5c 42 52 4f 57 53   .\MAILSL OT\BROWS
00d0  45 00 01 00 80 fc 0a 00  4e 42 32 2d 44 59 48 53   E....... NB2-DYHS
00e0  55 00 24 04 00 00 18 0f  05 01 03 10 01 00 0f 01   U.$..... ........
00f0  55 aa 00                                            U..
```

Figure 2: Ethereal on Linux

The advanced student may try to refine `nmap's` rule set for OS fingerprinting for an existing machine, or write a rule set for a new machine type.

2.3 Review Questions

1. What can `nmap` do to avoid being detected at the firewall which can easily detect an anomalous number of SYN packets going into the protected network but no handshake completion ACK?

2. Why do we need half-open (stealth) scans? What's wrong with making complete TCP connections to the destination port in order to test that it's live?

3. Why are half-open scans not so good for the end host?

4. The program `nmap` usually does a great job of doing OS fingerprinting. What is OS fingerprinting and why is it useful?

5. What in your opinion is a way for `nmap` to guess the service running at a port?

3 Assessment

When you are ready to be assessed, fill in this sheet with your details, attach your writeup for Step 1, and your answers to the review questions, give it to Pradeep and ask him to assess you.

Date:

Group members:

Name	Matriculation ID

Leave this section for Pradeep to fill in:

Using nmap:		
Using ethereal:		
General knowledge:		
Lab writeup:		/4
Review questions:		/4
