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 datagram 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

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?
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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Matriculation ID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Leave this section for Pradeep to fill in:

Using nmap:

Using ethereal:

General knowledge:

Lab writeup:  

Review questions:  /4