CS 491/591 Security and Privacy Spring 2010 Lab 2a

50 points (lab 2b will be the other 50)

Assigned: 13 March 2010

Due: Monday, 29 March 2010, at 11:59pm

Read these instructions carefully, all of this is important.

The purpose of this lab is to become familiar with port scanning. You should gain a sense of the tools that attackers use to plan an attack on a network, and the capabilities and limitations of those tools.

This lab is an individual effort, you may discuss it with your classmates at a high level only. Send a *.pdf to my gmail account for your final submission. It should be 12 pt font, one page in length, half a page explaining what you you found out about the network and how you found it out, and half a page explaining what you weren't able to find out and giving an attack strategy for lab 2b and beyond. Put some thought into it. Was the nmap tool ever wrong or misleading? Why? Why do you think that there's still some information about the network that needs to be found out in part b? How should we try to find this out in part b, *i.e.*, what are the limitations of our current tools?

In lab 2b, you'll develop a more powerful idle scan technique to answer some unanswered questions from 2a. On one of the machines on the `192.168.111.0/24` network, a copy of the final is in the `/root` directory. What do you still need to know before you develop a strategy for getting access on that machine? I strongly encourage you to submit early drafts so that I can give you feedback on things you can do to get more points.

Do not try your lab 1 exploit on any machines other than starbuck until lab 2 is over. It won't work anyway, and the purpose of lab 2 is to do some reconnaissance from the perspective of the `192.168.33.0/24` network only.

Start early and bounce your ideas off of me a few times and also check with me to see if you've found everything there is to find. Completeness of your results will be a large part of the grade. This isn't like lab 1, where you either got it done or didn't, for lab 2a there's a lot of things about the virtual networks on shasta that you need to figure out to get the points.

All scans should be done in polite (`-T2`) mode. Sneaky (`-T1`) and paranoid (`-T0`) are okay, too, especially if a lot of folks are logged in (*e.g.*, on the night that this is due). All scans should be performed from helo, and should target only the `192.168.26.0/24` and `192.168.111.0/24` IP address ranges.

Some resources are completely shared, such as zombies for idle scan, so that some coordination among those who are logged in and doing port scans is necessary. You should always work on this lab in two open terminals, one where you just log in to shasta and do "mesg y" so that you can send and receive messages, and another where you log in and do "mesg n" before sshing to helo, so that messages are suppressed in that terminal. Then, you can communicate with everyone logged into shasta at the time through the wall command. Do "man mesg" and "man wall". You are required to have at least one terminal accepting broadcast messages at all times while working on lab 2 (both parts a and b).

For example, if you're getting weird results from an idle scan on several zombies, you can type this in the shasta terminal:

```
wall Are lots of people doing idle scans right now?
```

Or, if nmap is giving inconsistent results you can type:

```
wall Is someone doing an aggressive scan right now?
```

Your message will be broadcast to everyone who is logged into shasta.

Your job is to tell me everything you can about the two virtual networks `192.168.26.0/24` and `192.168.111.0/24`. Your grade will be based on the completeness of what you tell me. There's a few things to be discovered beyond what machines there are and what ports they listen on, so start early and bounce your thoughts off of me.

Here's some tips:

* TCP RSTs and ICMP packets from `192.168.33.0/24` to `192.168.26.0/24` and `192.168.111.0/24` are dropped by shasta. You'll see the reason for this when we do lab 2b. What it means for part 2a is that ping scans are pointless, so every nmap command you type should have the "`-PN`" option.

* DNS is not configured on shasta. Nmap always tries to do reverse DNS lookups by default, so every command should have the "`-n`" option to disable this.

* All of the machines that I put on the virtual network run open source operating systems. They're headless virtual machines, meaning I can't access their terminals.

* To cut the time it takes you to discover machines in half (since this isn't really the interesting part), I didn't configure in machines above 127, i.e., every virtual machine is at `192.168.26.x` or

192.168.111.x where 2 <= x <= 127. Note that 192.168.26.1 and 192.168.111.1 are both shasta, which you're allowed to scan using these addresses only.

* To scan port 80 (HTTP) in the lower half of the 192.168.26.0/24 network, you would do :

```
nmap -n -PN -T2 -p80 192.168.26.1-127
```

* Note that "closed" and "filtered" can mean a lot of things, and not everything that nmap tells you is true.

* Once you've found a machine, you can throw "the works" at it. Nmap can detect the OS, the versions of services, and some other things for you. Where x is the last number in an IP address you want to scan, try:

```
nmap -n -PN -T2 -A 192.168.26.x
```

* You'll find a machine that appears to offer two types of services that are filtered (note that both of these services listen on multiple ports). Who's doing the filtering, shasta or the host itself? Are both services actually running on this server? Which subnetworks are allowed to access the services?

* Some machines in the virtual environment have been reconfigured to limit the rate at which they send TCP RSTs to 10 per second.

* Keep in mind that one of the purposes of port scanning is for the attacker to find machines that offer versions of services for which they have a remote exploit, such as an integer overflow. You can expect that the serverv3 capitalization service listens on 50 consecutive ports somewhere between 8000 and 9000.

* Perl source code for an idle scan based on FreeBSD RST rate limitations can be found on helo in /root. This is already tuned for 10 RSTs per second in the vritual environment (as opposed to 200 RSTs per second by default in FreeBSD), so you shouldn't need to modify it (but feel free to, after copying it to your own home directory). Note that this form of idle scan is somewhat limited and won't be able to answer everything you might try to find out with it (we'll develop a more powerful idle scan in part b of this lab).