Advanced Unix System Administration
Fall 2008
Homework 4

This assignment is due via email to <sluo+decal@ocf.berkeley.edu> by 11:59 PM on **Monday, November 17**.

1. *Networking on paper.* Here's an exercise to test your understanding of TCP/IP over Ethernet.

   a. Your company needs five different networks, four small networks of about 20 servers each, and a larger network of clients with addresses assigned by DHCP. You have the IP address range `172.17.42.0/24` to work with (I know this is RFC 1918 space – it's an example). Suggest a way to divide up this netblock into the networks you need.

   b. A computer on an Ethernet network with MAC address `FF:FF:FE:09:42:A3` and IP address `172.17.42.37` sends the message `Hello, world!\r\n` via UDP from port 51500 to a computer with MAC address `FF:FF:FB:3D:28:9C` and IP address `172.17.42.58` on port 9. Describe each of the Ethernet frames resulting from this conversation. Assume the sender's ARP cache is empty at the beginning of the conversation. *Note:* you do not need to write out each frame byte-by-byte – a description of the header and contents of each frame will do.

   c. A computer with IP address `172.17.42.37` initiates a TCP connection from port 51501 to a computer with IP address `172.17.42.58` on port 7. The computer on `.37` sends the string `Hello, world!\r\n` to the peer, which echos back the same message; the two computers then close the connection. Describe each of the IPv4 packets resulting from this conversation. *Note:* you do not need to write out each packet byte-by-byte – a description of the header and contents of each packet will do.

2. *Idle scan.* TCP initial sequence numbers aren't the only numbers that are problematic if they are predictable. There's an interesting technique called "idle scan", implemented in recent versions of `nmap`, that relies on a "zombie" host whose IPID numbers are predictable.

   a. How does this scan work? Why does the zombie host have to be idle? Where do the predictable IPID numbers come in?

b. From where does the scan appear to be coming from, the scanning host or the zombie? Why? Why might this be a problem if a zombie on your network is being used to scan one of your machines? *Optional:* If you have access to a suitable zombie host and a machine which you can do network configuration on (not your scanning host!), verify this.

c. What can you do to prevent idle scans from being launched from inside your network?

3. *Stalking hosts on networks.* There are a surprising number of instances when you'll want to discover information about the identity of a host based upon limited information, whether that host is on your local network or not. While there are no silver bullets, here's a chance to try a grab bag of useful techniques.

   a. *MAC addresses.* You'd think you'd know the hosts corresponding to each IP on your LAN, but whether it's other people not documenting what's plugged in, or someone stealing IPs on your network, that doesn't always turn out to be the case. If you have a managed switch, you can interrogate it to find out what switch port the host in question is plugged into, and follow the network cable to find the machine, but if not, you can use the knowledge that the upper half of the MAC address is a manufacturer code to make an educated guess. (Even if you have a managed switch, these techniques turn out to be generally useful.)

   Log in to a general-access OCF machine using your username and password. (If you don't have a regular OCF account, you may use `10.20.3.2` with your login server username and password, but it's strongly recommended you do this on the actual OCF if possible.) Determine the manufacturer of the host behind each of the following IP addresses, and make educated guesses about the function of each host where possible. Note any relationships between hosts that stand out to you. (OCF staffers, no cheating by looking at the staff documentation wiki!)

   - `192.58.221.229`
   - `192.58.221.247`
   - `192.58.221.246`
   - `192.58.221.231`

   Hint: How could you get your computer to display a table of IP to MAC address mappings? If the entry you're looking for isn't there, how might you make it appear?

Optional. Supposing you have a MAC address, how might you find a corresponding IP address?

b. *Reverse DNS.* Recall that DNS can provide PTR records that give the hostname corresponding to an IP address; the resulting hostnames can provide a surprising amount of information about the box you're studying.

   Perform reverse DNS lookups (I suggest `dig` or `host`, though there are other ways) on the following IP addresses, and where possible, draw conclusions about who controls the host, where it's located, and what it does.

   - `192.54.112.30`
   - `198.32.251.123`
   - `128.32.30.70`
   - `69.220.8.31`
   - `205.188.153.121`
   - `128.122.108.71`
   - `69.104.2.12`

   Suppose you're investigating suspicious traffic from a host. Why might a reverse DNS lookup notify the host's owners that you're looking into their behavior?

c. *WHOIS information.* The WHOIS system is a classic way of providing information about the owners of IP addresses and domains.

   Use a WHOIS client (like `whois`) to query WHOIS information on all of the above IP addresses. Combining this information with the information from part (b), tell me as much as you can about each host; in all cases, you should be able to obtain at least an owner or an upstream ISP, and contact information. (If all you get is an entry with something starting in `NET-`, try querying WHOIS for the `NET-` handle.)

   Also, try querying WHOIS for all of the domains that you find above. What information do you get?

   Optional. You'll undoubtedly have noticed that there are various different types of "handles" that come up in the outputs. Try querying some of these. What information do you get? What types of information can you get from WHOIS?

   Optional. Query `POEM-RIPE55-SONG`; does the content seem familiar to you?

d. What other techniques for discovering information about these unknown hosts

can you think of? *Optional:* Try them out and see what information you discover.

4. *Packet sniffing. Optional.* Go capture some packets on your favorite network. Analyze the traffic streams, and identify the conversations going on in the capture (who's involved, what protocols they're using, what they're doing). *Note:* This is a lot more fun if you do it on a non-switched network – public wireless networks are probably best.