CS 458 — Module 4: Networks

1 Intro to Networks

- To create a network, you need 3 things:
  1. Devices able to receive and send signals
  2. A way to connect devices to each other
  3. Rules for communicating, or a protocol
- Some examples of protocols:
  - Token ring — a person can only talk if they have the token
  - CSMA/CD — all listen to the wire, if they hear no signal they try to transmit. If there is a collision, then they all stop and resend.
- Well what are some problems?
  - The Internet's design connects many computer networks together. It also assumes that participants are honest and will cooperate — they will not look at messages that don't belong to them, they will not delete your messages, etc. Everyone should mutually work together. . . right?
  - There's also no routing logic in the addressing scheme — given some IP address, who knows where it comes from? For example, a phone number has an area/country code. An IPV4 address like `136.192.63.0` could come from anywhere!
  - Nor can you control the path your message follows!
  - Your message can be broken up with each part following a different route.
  - There is no real hard stop limit to the number of nodes (at least everywhere).
  - It's really hard to conceptualize.
  - Nobody is in charge (both good and bad).

2 Daemons, Servers, Ports

- A server is a computer on a network to do tasks for other computers (clients).
- A daemon is like a servant that can only do one task within a server.
- We can think of a server like a huge apartment building, and each apartment can have one servant (daemon).
- For example, the mail sending daemon (SMTP) is 25.
- Some apartments (ports) can be empty. Many ports are actually empty!
- One could hide a service in a port it's not supposed to be in.
- For example, an HTTP daemon is in port 80. This is implied by default (ie: `https:`*`//www.uwaterloo.ca`* implies `https:`*`//www.uwaterloo.ca:80`*).
- But one could put a web service at, for example, port 8080.
- A "loose-lipped" system may reply to an attacker and advertise what services they are running *and* what at what port.

3 Port Scanning, Information Gathering, Wiretapping, Impersonation

3.1 Port Scanning

- A port scan checks every port in sequence.
- We would ideally want, at least for security, to not reply when ports are checked.
- Unfortunately, this isn't really possible as we need this replying for actual use.
- Tools like `nmap` would give many details about a machine.
- A command like `finger` allows you to look up a user in a machine. If this is not closed, one could do it from outside a machine...
- But this is just the beginning... maybe you could get some info, but port scanning is not really malicious on its own yet.

3.2 Intelligence Gathering

- Social Engineering is attacking people via exploiting other humans, which can give valuable information.
- Pretending to be part of an organization they're not, exploiting the helpful nature of people ("I forgot my password"), distractions to grab information somehow, etc.
- Other ways you could get info?
  - Dumpster diving
  - Eavesdropping
  - Lots of things placed online that shouldn't be there — Google, social media, etc.
- Wiretapping
  - Two types: passive and active.
  - Passive wiretapping is basically just eavesdropping. When a message is sent, a node could read the destination data — but there is *nothing* stopping Eve from looking at the data!
  - The analogy is an envelope with an address and a non-sealed back.
  - Active wiretapping will require modification/fabrication of communication.
  - For example, Mallory could modify a message sending money from one account to another. That is, Mallory is usually a MITM during an active wiretap attack.
  - One can also eavesdrop while communication is flowing through a link; we call "promiscuous sniffing".
  - We should *always* assume someone is eavesdropping the data!
  - The degree of vulnerability would depend on the communication media:
    - For example, copper cables mean that a physically close attacker could eavesdrop without making physical contact, or just cut the cable open/splice in another cable.
    - Coaxial cables help shield some of this signal from leaking out compared to twisted pair cables.
    - Optical fibre would be harder, as there is no inductance and signal loss caused by splicing would be noticeable.
    - Unbound transmission is through the air — WiFi, microwaves, radio, etc.
    - This is versus bounded, like cables.
    - How could we protect something like WiFi? Problems are:
      - It is easy to intercept with anything that can use WiFi.

      - It’s easy to read packet info like destination and source IP addresses — even at a distance!
      - Physical barriers are useless for a wireless network.
      - Wireless APs can also be faked; one could use a router that is not actually owned by the network you are connecting to to steal credentials.
  - When we transmit data, how do we choose what medium?
    - Is it sensitive? If it is, we probably don’t want to use an unbounded medium.
    - Are there *segments* of the network carrying sensitive data?
    - Would one notice of an intruder is eavesdropping? For example, using barriers that would make an attack obvious due to damage to said barrier.
    - Are backbone segments accessible? Can an intruder actually attack said parts of the network?

3.3 Impersonation

- A person could try to log into a machine that does not belong to them by pretending to be an owner.
- Steal passwords, guess, social engineering, sniff password, etc.
- Or pretend to act like a machine itself.