- Terms/Tools:
    - ARP table
        - Address Resolution Protocol
        - Contains both IP and MAC addresses for devices (like an address book)
        - Being altered in ARP Poisoning
    - MAC address
        - Media Access Control Address
    - ARP Poisoning
        - Man in the middle attack
        - Deceives victims into thinking they are the router
        - Deceives router into thinking they are the victim
    - Scapy
        - Python library for ARP poisoning, packet manipulation
    - Backbox
        - OS used for security and penetration testing
- Python Functions:
    - arping()
        - Returns MAC address for IP using ARP table
    - send()
        - Sends packets
        - Arguments
            - op = “type”
                - Type can be 1 or 2
                - 1 indicates sending a request

        - 2 indicates sending a response with information
    - pdst = “ip destination”
        - IP you’re communicating with/sending to
    - psrc = “ip source”
        - The IP you’re pretending to be from
    - hwdst = “hardware destination”
        - MAC address of the device you’re communicating/sending to
        - When restoring the ARP Table, make sure to restore to all devices by calling all MAC addresses using “ff:ff:ff:ff:ff:ff”
    - hwsrc = “hardware source”
        - MAC address of the device you’re actually from
        - Essentially making sure the IP you’re trying to be is connected to YOUR MAC address in the ARP table so you can intercept
    - count = “number of times”
        - How many times you’re sending packets
        - 3 is a good number
        - This argument is optional, you don’t need to specify a number
- Terminal Commands:
    - arp -a
        - Lists the entire ARP table with all IP and MAC addresses

- cd
  - Change directory
- ifconfig
  - Shows network information
  - First inet address is my own IP address
  - Mask address determines which space is available for device addresses
    - ".0" signifies that those are where other device IPs lie
    - 255.255.0.0 means the last two places are available
    - For an IP of 192.168.101.173, the last two places differ among devices on that network
    - So the first possible address is 192.168.100.1 (the second place's first possible value is 100, not 1)
    - The first possible address on the network is the router
- ip route show
  - Shows route to router (also displays router IP)

- Process
  - Set IP address of intended victim
  - Find the IP address of the router using either ifconfig or ip route show
  - Use arping to find the respective MAC address
  - **Enable IP forwarding by editing configuration**
  - Poison ARP table by pretending to be the router to the victim and pretending to be the victim to the router
    - ```
      def poison(routerIP, victimIP, routerMAC, victimMAC):
        send(ARP(op=2, pdst=victimIP, psrc=routerIP, hwdst=victimMAC))
        send(ARP(op=2, pdst=routerIP, psrc=victimIP, hwdst=routerMAC))
      ```
      ```

- Restore ARP table by telling all devices in the network that the original address and IPs match up like they were originally
    - ```
      def restore(routerIP, victimIP, routerMAC, victimMAC):
          send(ARP(op=2, pdst=routerIP, psrc=victimIP, hwdst="ff:ff:ff:ff:ff:ff", hwsrc=victimMAC), count=3)
          send(ARP(op=2, pdst=victimIP, psrc=routerIP, hwdst="ff:ff:ff:ff:ff:ff", hwsrc=routerMAC), count=3)
      ```
- Carry out spoof attack
      ```