Network & Web Security

Packet Sniffing and Spoofing (Part II)

Professor Travis Peters
CSCI 476 - Computer Security
Spring 2020

Some slides and figures adapted from Wenliang (Kevin) Du's
Computer & Internet Security: A Hands-on Approach (2nd Edition).
Thank you Kevin and all of the others that have contributed to the SEED resources!

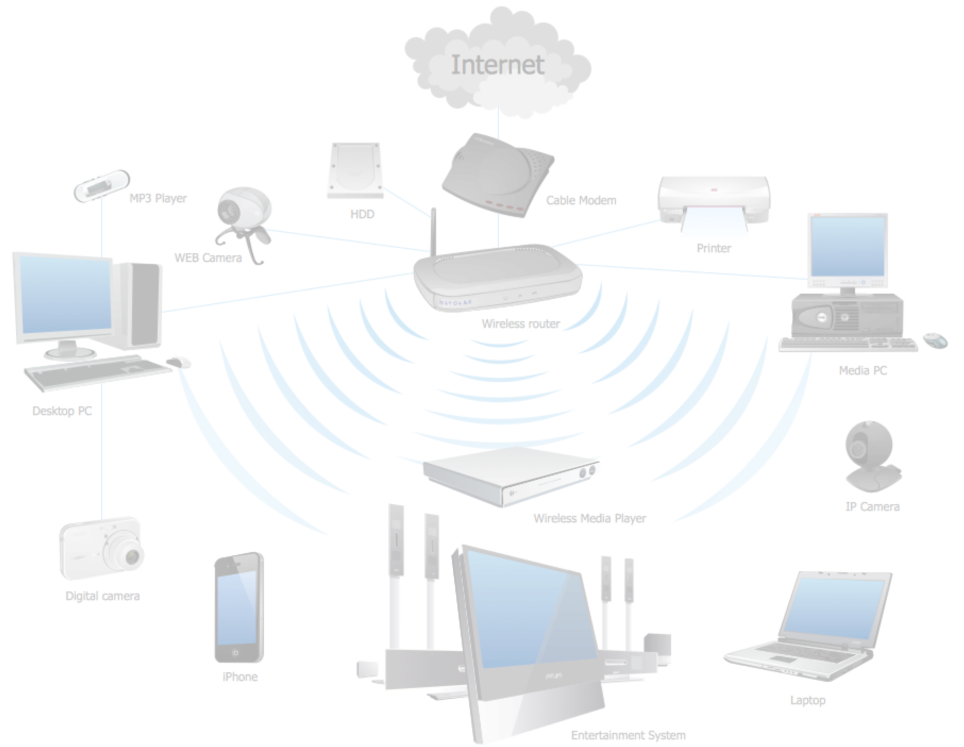
Today

Announcements

- Lab 06 Due Today!!
- Lab 07 Up! (converted to webpage; not a PDF)

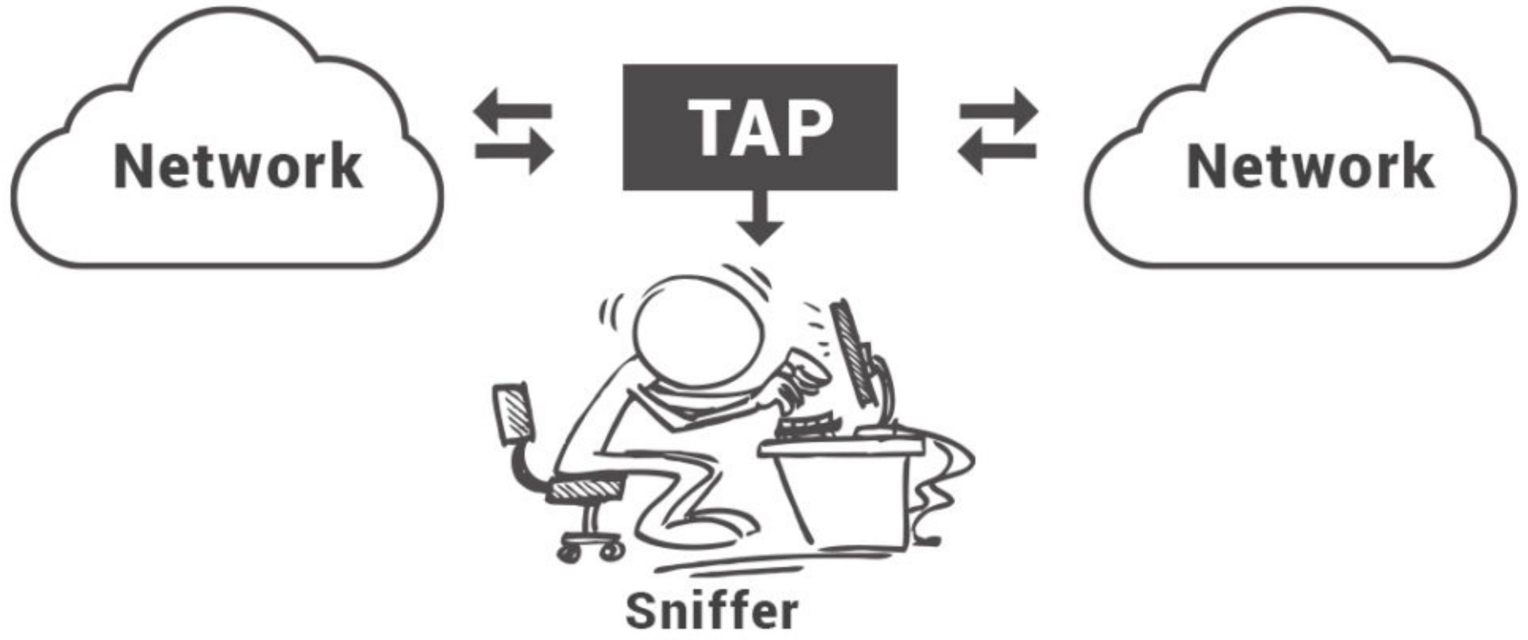
Goals & Learning Objectives

1. Network Basics

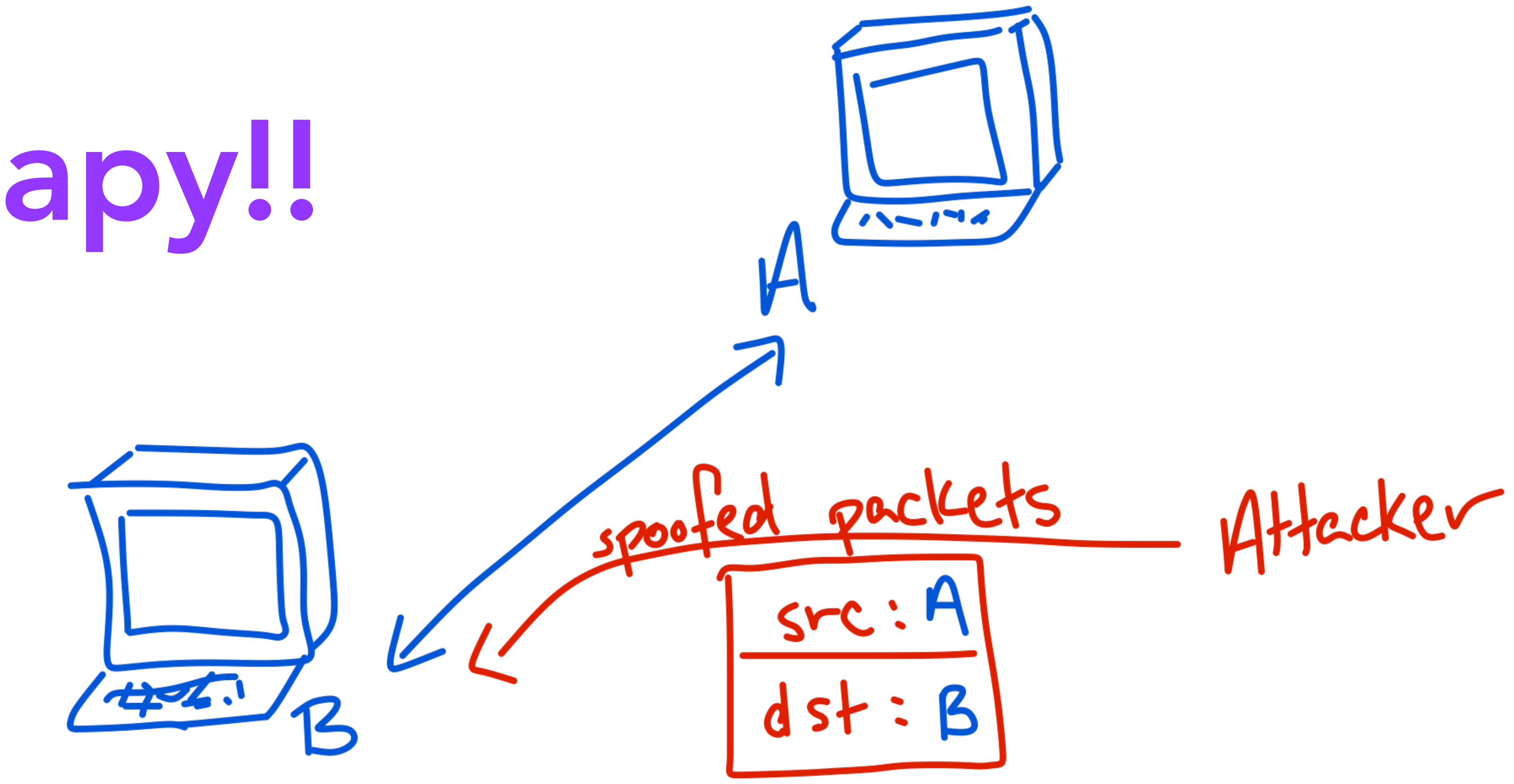


2. Packet *sniffing* **AND** 3. Packet *spoofing*

w/ Scapy!!



How can we build a sniffer?



How can we spoof packets?

Sniffing and Then Spoofing Using Scapy

Example: Sniff+Spoof ICMP packets

```
#!/usr/bin/python3
from scapy.all import *

def spoof_pkt(pkt):
  if ICMP in pkt and pkt[ICMP].type == 8:
     print("Original Packet.........")
     print("Source IP : ", pkt[IP].src)
     print("Destination IP :", pkt[IP].dst)

     ip = IP(src=pkt[IP].dst, dst=pkt[IP].src, ihl=pkt[IP].ihl)
     icmp = ICMP(type=0, id=pkt[ICMP].id, seq=pkt[ICMP].seq)
     data = pkt[Raw].load
     newpkt = ip/icmp/data

     print("Spoofed Packet.........")
     print("Source IP : ", newpkt[IP].src)
     print("Destination IP :", newpkt[IP].dst)

     send(newpkt,verbose=0)

pkt = sniff(filter='icmp and src host 10.0.2.69',prn=spoof_pkt)
```

sniff_spoof_icmp.py

Activities

- Working with IP packets
  - Create an IP Packet
  - Set the source address to **1.2.3.4**
  - Set the destination address to **10.152.183.104** ***(you must be on MSU Secure)***
  - Print out the packet
  - Send a packet to **10.152.183.104**
  - Send multiple such packets to my machine

- Stack packet layers
  - Create an IP packet with an ICMP packet
  - Create an IP packet with a UDP that contains your name

- Send me 10 UDP packets where you increment a counter in the payload (e.g., 1, 2, 3, …)

- ~~**(With a partner)** Write a sniffer to capture only certain packets~~
  - ~~Create a sniffer that captures only ICMP packets~~
  - ~~Send ICMP packets to a friend~~

Packet Spoofing: Scapy vs. C

We're not prepared for the end of Moore's Law
https://www.technologyreview.com/s/615226/were-not-prepared-for-the-end-of-moores-law/

"Thompson and his colleagues showed that they could get a computationally intensive calculation to run some 47 times faster just by switching from Python, a popular general-purpose programming language, to the more efficient C. That's because C, while it requires more work from the programmer, greatly reduces the required number of operations, making a program run much faster. Further tailoring the code to take full advantage of a chip with 18 processing cores sped things up even more. In just 0.41 seconds, the researchers got a result that took seven hours with Python code."

- Python + Scapy
  - **pros:** constructing packets is very simple
  - **cons:** much slower than C code
- C Program (using raw socket)
  - **pros:** much faster
  - **cons:** constructing packets is complicated
- Hybrid Approach
  - Using Scapy to construct packets
  - Using C to slightly modify packets and then send packets
  - (There are some examples in the course repo)

Summary

- Packet sniffing
  - Using raw sockets
  - Using PCAP APIs
- Packet spoofing
- Sniffing and then spoofing
- Endianness & Checksums — *see the textbook*

You Try!

Exam-like problems that you can use for practice!

- There are two typical approaches for a sniffer program to filter out unwanted packets. The first approach gets all the packets from the system, and then filters out unwanted ones, before presenting the results to users (or save to files). The second approach uses `pcap_setfilter` to set the filter. Please describe the differences of these two approaches.
- An integer 0xAABBCCDD is stored in a memory address starting from 0x1000. **(1)** If the machine is a Big-Endian machine, what is the value stored in addresses 0x1000, 0x1001, 0x1002, and 0x1003, respectively? **(2)** If the machine is a Little-Endian machine, how is this integer stored?
- Is it possible to spoof a packet with a size larger than 65535, which is the upper limit of the IP packet size (the length field in the IP header has only 16 bits)? Explain.
- In the past, one can send a broadcast packet to all the machines on a subnet. This is called *Directed Broadcast*. If the subnet is 10.0.2.0/24, the directed broadcast address is 10.0.2.255. *How can we use this feature to launch a denial-of-service attack on a victim?*
  - Hint: Basically, we would like to send a lot of packets to the target machine, but we cannot afford to do it ourselves, because the target has a larger bandwidth than us. We need to find a way to turn one packet into many.