Forensics II

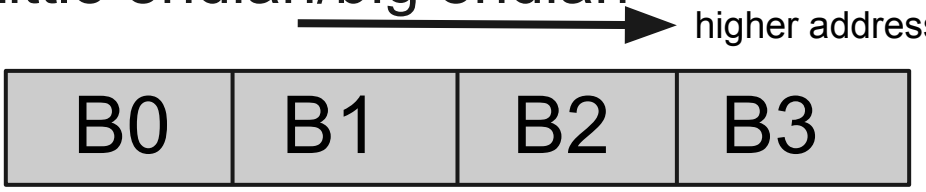
98-212 Week of Jan 28th

Before we start...

- Less than 20% of people in the class have scored points
  - What's up with that?
- Good news
  - Worst is over (seriously stego is obnoxious)
  - We put it first to get it out of the way
  - We'll also try to include more info in our slides for you to refer to later
- Bad news
  - Still a little bit more forensics to talk about
  - But it's more reasonable than stego!

Encodings

- An important thing to think about
  - Didn't really fit anywhere else in terms of lectures
  - Recognizing encodings is very useful
- Multibyte values
  - Integers (4 bytes)
  - Longs (8 bytes)
  - Shorts (2 bytes)
  - Strings (* bytes)
  - Nibble (1/2 byte, or 1 hex character)
  - Little endian/big endian



Encodings

- Hexadecimal
  - Hopefully people are very comfortable with hex
  - Common uses:
    - Reading/writing raw binary
    - hex editors: *you should have one*
      - Windows: hexplorer or 010 editor
      - Linux: bless, ghex2
      - OSX: hex fiend, 0xED
  - Very widely used for binary information
  - python <3.0: "6578616d706c65".decode("hex")
  - xxd command line tool

Encodings

- Base64
  - Characterset is ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/
  - Commonly used for encoding binary data in a printable form
    - email attachments, pgp keys, web cookies
  - More efficient than hex
  - Sometimes padded (with "="s), or using - and _
  - Example GX/dB640lntDNnAN3sRdOYJS+ur4JzyEy0e03483DHMYB6/94S9lZE zPFDh8oMregppA79Cf1oTECljyPBdk9ctIzkFq+EFdA2UmvKIPbKM=
  - Again, python: s.decode("base64")
  - base64 command line tool

Encodings

- ASCII - American Standard Code for Information Interchange
  - Most common representation for printable characters
  - Important values:
    - 0x20 (32): ' '
    - 0x21 - 0x2f (33-47): symbols
    - 0x30 - 0x39 (48-57): '0' - '9'
    - 0x41 - 0x5a (65-90): 'A' - 'Z'
    - 0x61 - 0x7a (97-122): 'a' - 'z'
    - 0x80 and above: *not printable!*
  - Good to be able to know if a value should be text or not!
  - "man ascii" for a chart on unix machines

Encodings

```
4885c07408bf380e60c9ffe0c9c39090554
889e54881ec30
646f657320616e796f6e65206b6e6f77206
9662074686973206973204153434949
5305bb520c2959e7c4ba1b4a9aaa7c69622
8d4e91da0c5ee4b
23696e636c756465203c737464696f2e683
e0a0a696e74206d61696e2869
```

Encodings

Questions?

Memory Forensics

- Basic idea:
  a. Get a dump of RAM from a running system
  b. Search for: passwords, browser history, encryption keys, running process information, etc
  c. Profit!?!

Memory Forensics

- Real world examples:
  - Firewire "feature" and DMA
  - Cold boot attack
  - Coredump of a crashed process
- Not only used to search for user data
  - Debugging
  - Searching for malware (especially rootkits)
    - Why?

How do you read a memory dump?

- Any ideas?

How do you read a memory dump?

- Easy:
  - Strings or grep
  - Vim or less
  - Hex editor
- Medium:
  - GDB
  - IDA
- Advanced:
  - Volatility
  - Specific tools for different uses (eg AESKeyFinder)

Examples!

Take away message

- Viewing things raw works pretty well
  - If you know what you're looking for, '`strings | less`' or something can work very well!
- ...but sometimes you will need more advanced tools
  - If you don't know what you're looking for, you probably need to use gdb, IDA, or a hex editor
  - Make heavy use of "find" commands

Why is memory forensics challenging?

- Virtual/physical memory can be messy
- Understanding structures in memory can be difficult
- *A lot* of data to look at
  - Similar to disk image forensics

Memory forensics

Questions up to this point?

Network Forensics

- Actually a kind of fun topic!
- Basic idea:
  - You are given a network capture (packet dump)
  - Look through the packet dump and find the suspicious information!

Network Forensics

- Where do you get a packet dump?
  - tcpdump
  - libpcap
  - tons of other tools!

Network Forensics

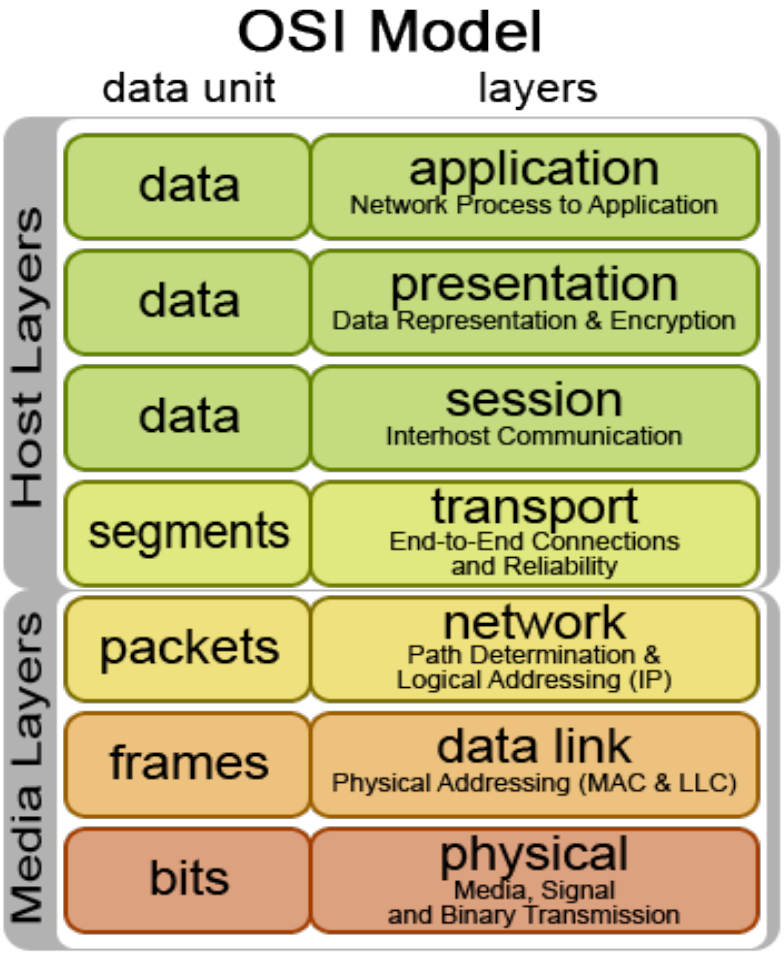
- Real world examples:
  - Incident response (similar to disk image forensics)
    - We know we were hacked, what did the hackers take?
    - How did they get in?
  - Packet monitoring of suspected criminals/terrorists
  - Intrusion detection/prevention system (IDS/IPS)
    - Does network analysis in real time to (try to) detect/prevent attacks
  - Reverse engineering
    - Reading obfuscated code can be a pain
    - Sometimes communication is the weakest link
  - Debugging broken networking equipment

Network Forensics

- CTF style problems:
  - Try to recover key transmitted in some weird way
    - Usually obscure/cute protocols
    - Sometimes horrible stego :(
  - Extract files cleanly from packet stream
- Attack-Defense CTF:
  - Commonly you are given the ability to log packets
  - This allows you to look for exploits used against your own server (and "reflect" them)

A tiny bit about networking

- Standard description of network "layers"
- Different devices/ protocols at each layer

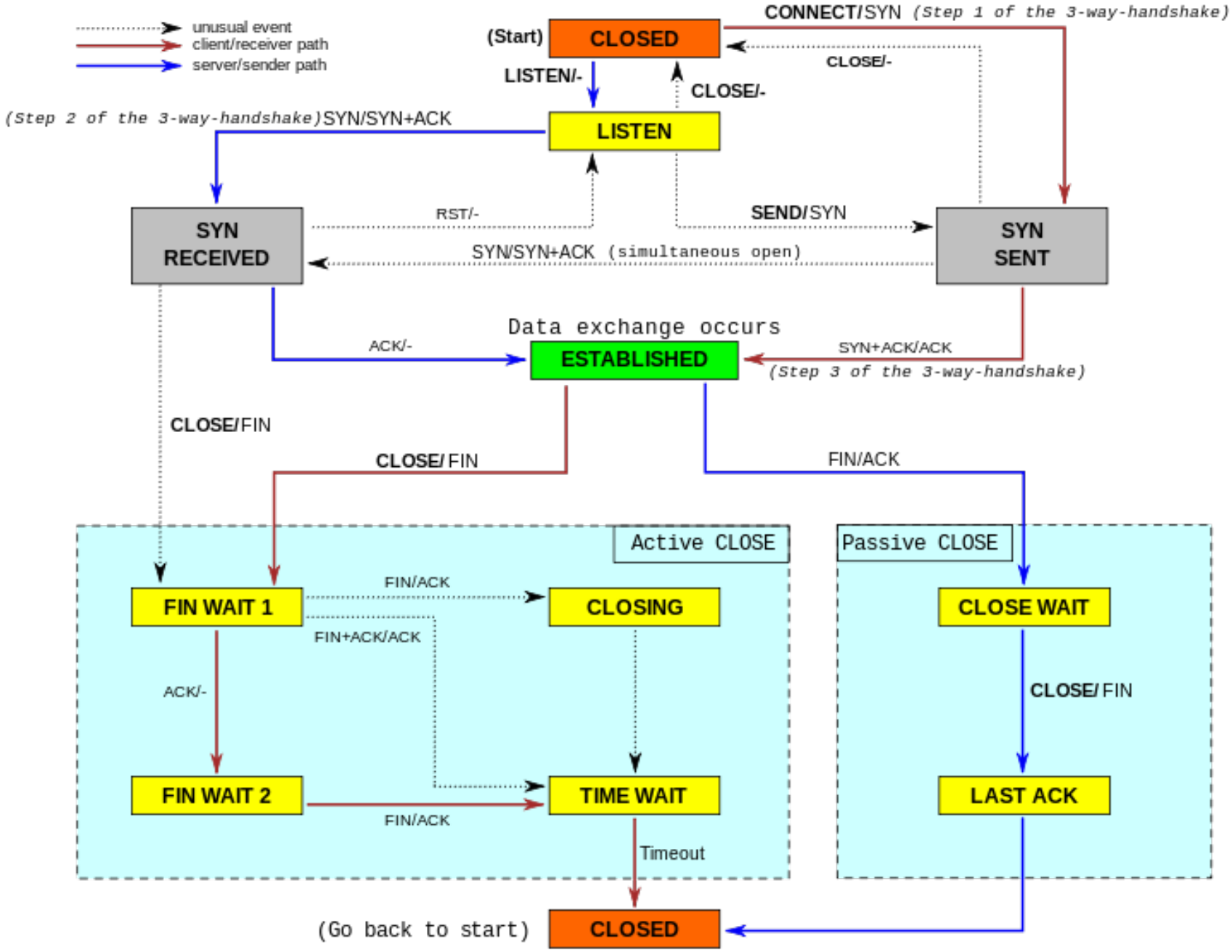


Networking protocols

- UDP
  - Packets have source and destination port for keeping in touch
  - Once a packet is sent, it may never come back
  - Things may arrive out of order
  - *Very* simple protocol
  - "stateless" and "unidirectional"
  - Used often for streaming

Networking protocols

- TCP
  - Most familiar protocol (websites, bittorrent, chat, etc)
  - Source and destination ports, but also sequence numbers, flags, "urgent pointers", "window sizes"
  - *Stateful*
    - Requires a 3 part handshake to establish connection
    - "sessions" or "conversations" are possible
  - Very robust:
    - If client doesn't get data, try to resend it
    - Should be difficult to hijack a "session"


unusual event
client/receiver path
server/sender path
(Start)
CLOSED
CONNECT/SYN (Step 1 of the 3-way-handshake)
CLOSE/-
LISTEN/-
CLOSE/-
(Step 2 of the 3-way-handshake)SYN/SYN+ACK
LISTEN
SYN RECEIVED
RST/-
SEND/SYN
SYN SENT
SYN/SYN+ACK (simultaneous open)
Data exchange occurs
ACK/-
ESTABLISHED
SYN+ACK/ACK
(Step 3 of the 3-way-handshake)
CLOSE/FIN
CLOSE/ FIN
FIN/ACK
Active CLOSE
Passive CLOSE
FIN WAIT 1
FIN/ACK
CLOSING
FIN+ACK/ACK
CLOSE WAIT
ACK/-
CLOSE/ FIN
FIN WAIT 2
TIME WAIT
LAST ACK
FIN/ACK
Timeout
(Go back to start)
CLOSED

Other protocols

- ICMP (ping)
- HTTP and FTP (on top of TCP)
- DNS (on top of UDP... usually)
- Hundreds more.....

Network Forensics

- Excellent tool: Wireshark
- You should all use wireshark
  - works on all platforms, fancy GUI, free and open source

- Demo!

Network Forensics

- For a lot of things, `statistics -> conversations list` gives a quick overview of what happened
- Right click on a packet and `follow stream` is awesome for most TCP conversations
- If you know what you're looking for, you can use `edit -> search`
- Everyone uses wireshark, you can find tutorials online!

Network Forensics Difficulties

- Understanding protocols
  - Wireshark handles a lot of this for you
  - Sometimes wireshark can be fooled
- Reconstructing data
  - Packetized protocols may reorder data
  - Packets might also overlap!
  - (wireshark will do most of this)
    - Other tools: tcpxtract, tcpflow, streams
- Encryption/compression
  - SSL or SSH traffic is encrypted
  - Most websites transmit data with gzip

Network Forensics Difficulties

- Malicious environments
  - Wireshark is a complicated piece of software
    - Written in C (that means bugs)
    - Disabling "dissectors" you don't need is a good idea
  - This is more common than you might think

Network Forensics Difficulties

- Quick example... (from Defcon)

Network Forensics

- Questions?