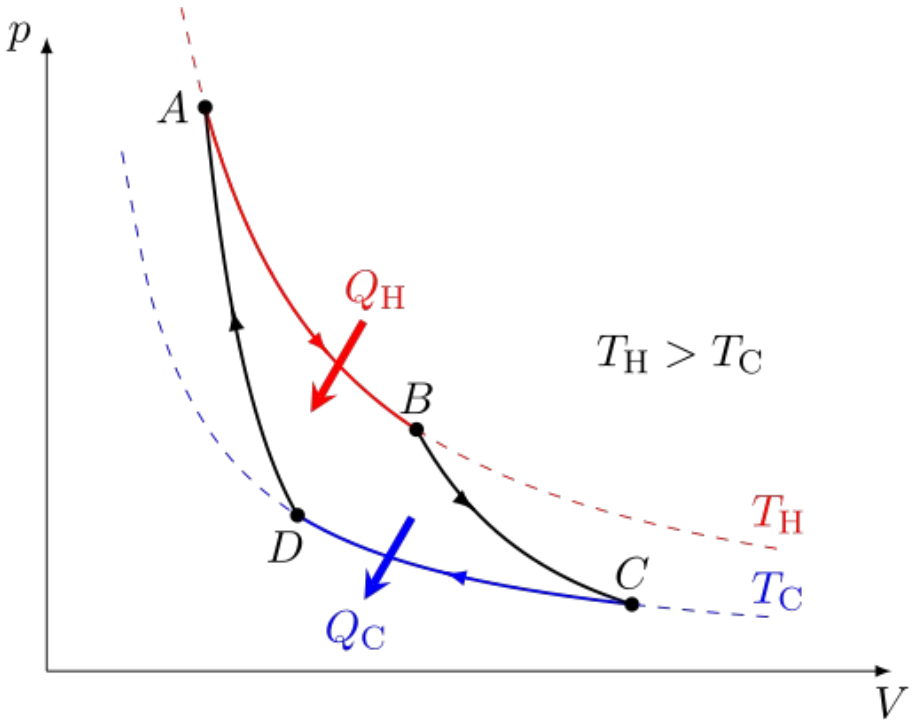
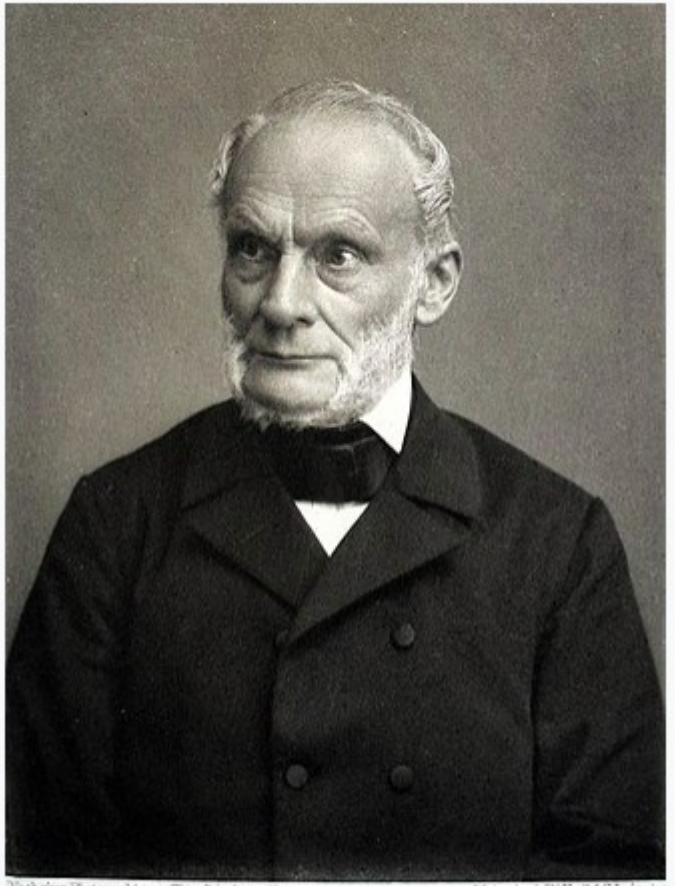
Port scanning and network side channels

CSE 548 Spring 2026
jedimaestro@asu.edu



https://en.wikipedia.org/wiki/Nicolas_L%C3%A9onard_Sadi_Carnot
https://en.wikipedia.org/wiki/Carnot_heat_engine

Rudolf Clausius



“entropy”

(from Greek ἐν en "in" and τροπή tropē "transformation")

Like energy, but you can’t use it.

https://en.wikipedia.org/wiki/Rudolf_Clausius

Entropy

- Statistical foundation by Gibbs, Boltzmann, Maxwell, Planck, *etc.*
- Directly inspired the name of entropy in Shannon’s information theory

$$H = -\sum_i p_i \log_2 (p_i)$$

Requirements (Shannon, 1948)

1) $I(p) \geq 0$ (information is non-negative, $p \geq 1$)
2) $I(1) = 0$ (events that always occur carry no information)
3) $I(p_1 p_2) = I(p_1) + I(p_2)$ (information due to independent events is additive)

Also, continuity, symmetry, and maximum when all possible events are equiprobable.

$$I(p) = \log(1/p)$$

1) $I(1/2) = 0.30102999566...$
2) $I(1) = 0$
3) $I(1/2) + I(1/3) = \log(2) + \log(3) = 0.77815125038...$

Joint probability: $I(1/6) = 0.77815125038...$

Continuity: $I(1/2.01) = 0.30319605742...$

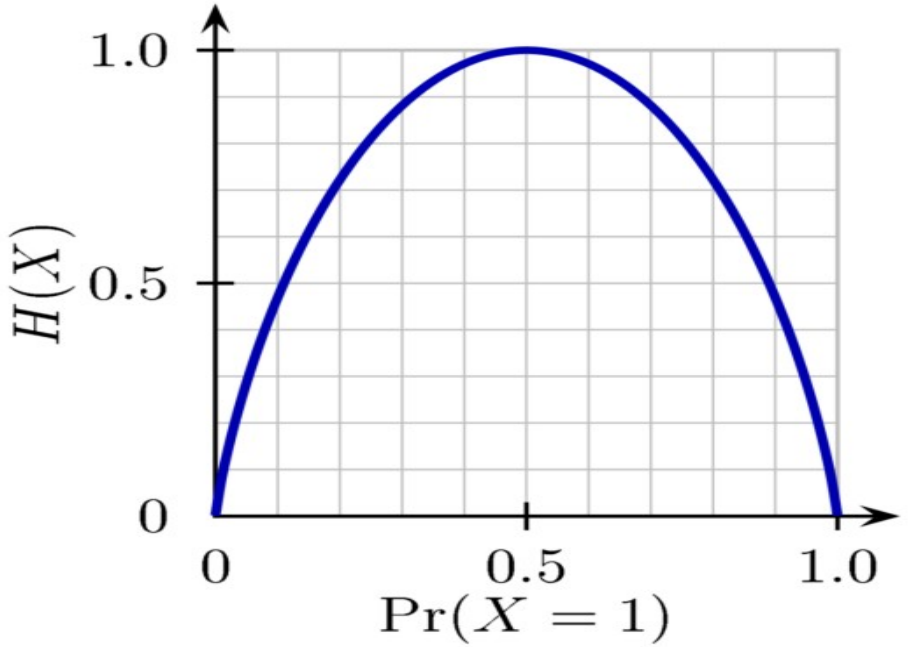
Symmetry: $\log(3) + \log(2) = 0.77815125038...$

Maximum: $\log 3 + \log 3 + \log 3 = 1.43136376416...$

$\log 2 + \log 4 + \log 4 = 1.20411998266...$

Information = Entropy = Surprise

$$H[p] = -\sum_{i=1}^{k} p_i \log p_i$$



Side note – Differential Entropy

https://en.wikipedia.org/wiki/Differential_entropy

$$h(X) = \mathbf{E}[-\log(f(X))] = -\int_{\mathcal{X}} f(x)\log f(x)\,dx$$

f(x) is a probability density function for the signal, the more the signal “jumps around” the higher the entropy, therefore modulating higher frequencies
means more entropy and therefore more bandwidth.

Not a pop quiz #1

- When a 3yo walks by with a stepstool…
  - 4 times out of 10 it’s to get something they’re not supposed to have
  - 2 times out of 10 it’s to climb up to somewhere they’re not supposed to be
  - 1 time out of 10 it’s to wash their hands
  - 1 time out of 10 it’s to get something they are allowed to have
  - 1 time out of 10 it’s to use as a dollhouse
  - 1 time out of 10 it’s to turn over and use as a storage bin
- What is the entropy of each instance of 3yo stepstool habits?

Answer

Input:

$$-0.4 \log_2(0.4) - 0.2 \log_2(0.2) - 4\,(0.1 \log_2(0.1))$$

Result:

$$2.32193\ldots$$

Not a pop quiz #2

- There are three possible states the Tempe weather could be in during any given hour on a summer day (very hot and bright out, very hot and it’s nighttime, monsoonal rains). What probability distribution over these events would give the maximum entropy in terms of what you might observe in a randomly chosen hour from the summer?

Answer

Input:

$$-\frac{1}{3}\log_2\left(\frac{1}{3}\right)-\frac{1}{3}\log_2\left(\frac{1}{3}\right)-\frac{1}{3}\log_2\left(\frac{1}{3}\right)$$

Exact result:

$$\frac{\log(3)}{\log(2)}$$

$\log(x)$ is the natural logarithm

Decimal approximation:

More digits

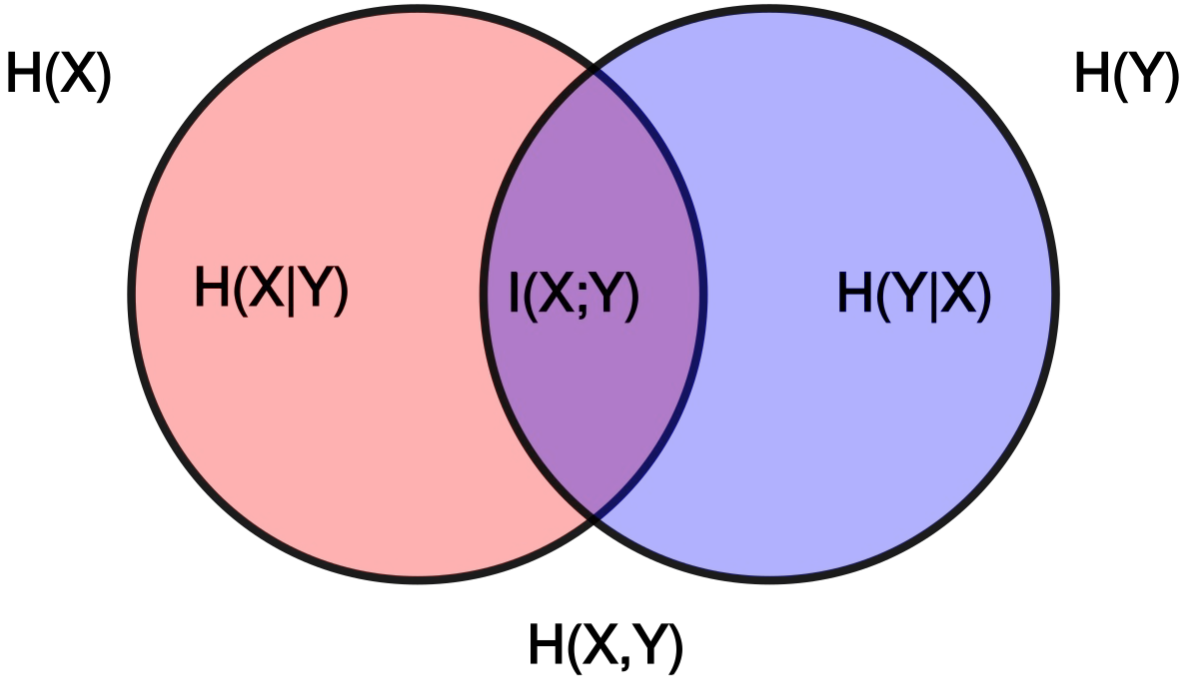
1.584962500721156181453738943947816508759814407692481060455...

Not a pop quiz #3

You have 12 coins, one is counterfeit. The counterfeit is either slightly heavier or slightly lighter, otherwise it’s impossible to tell. You have a balance. Using the balance the fewest number of times, find the counterfeit coin.

https://en.wikipedia.org/wiki/Mutual_information

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$



TCP 3-way handshake (review)

- SYN: I'd like to open a connection with you, here's my initial sequence number (ISN)
- SYN/ACK: Okay, I acknowledge your ISN and here's mine
- I ACK your ISN

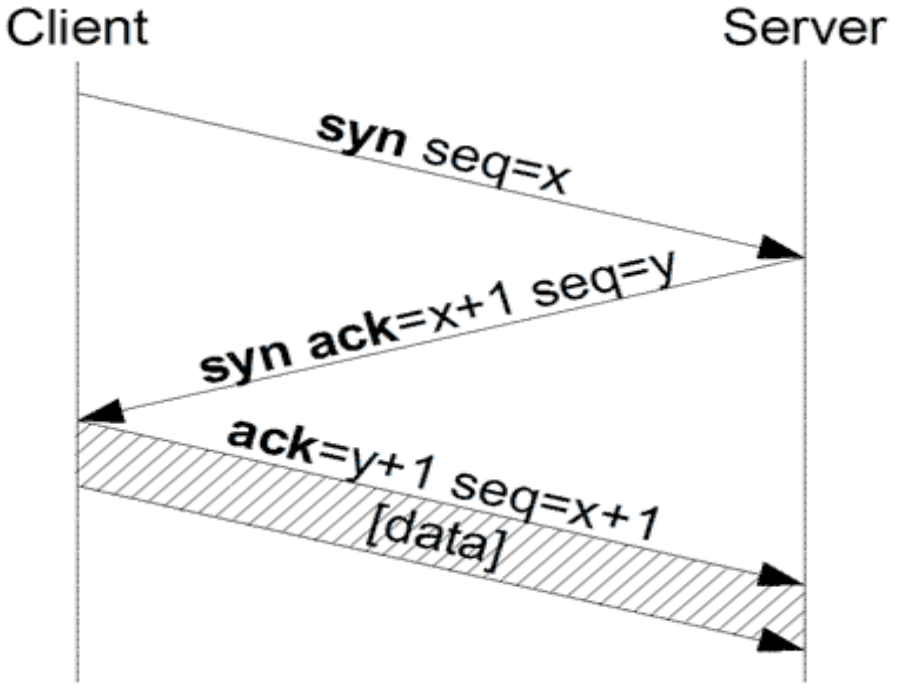


Image from Wikipedia

Port scanning

- Related terms
  - War dialing
  - banners
- Side channels
  - Infer things off-path

Open port == listening

- If you send a SYN packet to port 80 (the HTTP port) on a remote host and that host replies with a SYN/ACK, then we say that port 80 on that machine is “open”
  - In this example, that probably means it's a web server
- If it responds with a RST, we say it's “closed”
- If there is evidence of filtering (no response or ICMP==Internet Control Message Protocol error), we say it's “filtered”
  - UDP is more complicated: open|filtered *vs.* closed

Things nmap can do

- Is a port open? Closed? Filtered?
  - Many ports on one machine is a “vertical scan”
- For a /24 network, which machines are up? Which machines have port 80 open?
  - One port for a range of machines is a “horizontal scan”
- OS and service detection (research on your own)
- Stealth, info about middleboxes, etc.

Faster alternatives to nmap...

Paketto Simplified (1.0)

November 18, 2002 Dan Kaminsky Leave a comment Go to comments

SCANRAND

========

Really, really fast port scanner, that can also trace network paths. Port scanning is simply the act of asking a machine if you can start up a conversation with a certain port of its, and marking down “yes” or “no” depending on the response. Normally, there’s lots of overhead as you keep track of who you sent requests to and thus who you’re expected responses from. Overhead, or “state”, makes things slow. So scanrand is stateless — right when you start up, it splits in two. One half asks everyone, “Heh! What are you hosting!” The other half picks up responses, “Hmmm, some guy just said he has a web server.”

Now, there’s a problem: If someone knows I’m not keeping track of who I’m scanning, they can just throw fake responses back at me. But TCP lets me embed a little signature with every connection request — the “Sequence Number”. This number will be returned to me when I get a valid response from a host that I scanned. So I take the IP and the port of the machine I scan, encrypt it into the sequence, and send off the request. When I get the response back, I look at the ACKnowledgement, compare it to the IP and port of the machine that’s talking to me, and immediately know whether I ever scanned this guy in the first place.

Unicornscan (2005)

https://www.kali.org/tools/unicornscan/

```
root@kali:~# unicornscan -mTsf -Iv -r 1000 192.168.0.102:a
adding 192.168.0.102/32 mode `TCPscan' ports `a' pps 1000
using interface(s) eth0
scaning 1.00e+00 total hosts with 6.55e+04 total packets, should take a little lo
connected 192.168.103.227:23221 -> 192.168.0.102:445
TCP open 192.168.0.102:445  ttl 128
connected 192.168.103.227:50006 -> 192.168.0.102:443
TCP open 192.168.0.102:443  ttl 128
connected 192.168.103.227:54487 -> 192.168.0.102:161
TCP open 192.168.0.102:161  ttl 128
connected 192.168.103.227:47765 -> 192.168.0.102:80
TCP open 192.168.0.102:80  ttl 128
connected 192.168.103.227:4267 -> 192.168.0.102:1884
TCP open 192.168.0.102:139  ttl 128
sender statistics 963.9 pps with 65536 packets sent total
listener statistics 131180 packets recieved 0 packets droped and 0 interface drop
TCP open                 http[   80]     from 192.168.0.102  ttl 128
TCP open          netbios-ssn[  139]     from 192.168.0.102  ttl 128
TCP open                 snmp[  161]     from 192.168.0.102  ttl 128
TCP open                https[  443]     from 192.168.0.102  ttl 128
TCP open        microsoft-ds[  445]     from 192.168.0.102  ttl 128
root@kali:~#
```

Zmap (2013)

- Fast scanner designed for horizontally scanning the entire Internet
- $2^{32} + 15 = 4294967311$ (prime number)
- Required reading, so omitting details here

Side channels...

Idle scan

- Every IP packet sent has an IP identifier
  - In case it gets fragmented along the way
- Old machines (or just that are configured that way) use a globally incrementing IPID that is shared state for all destinations

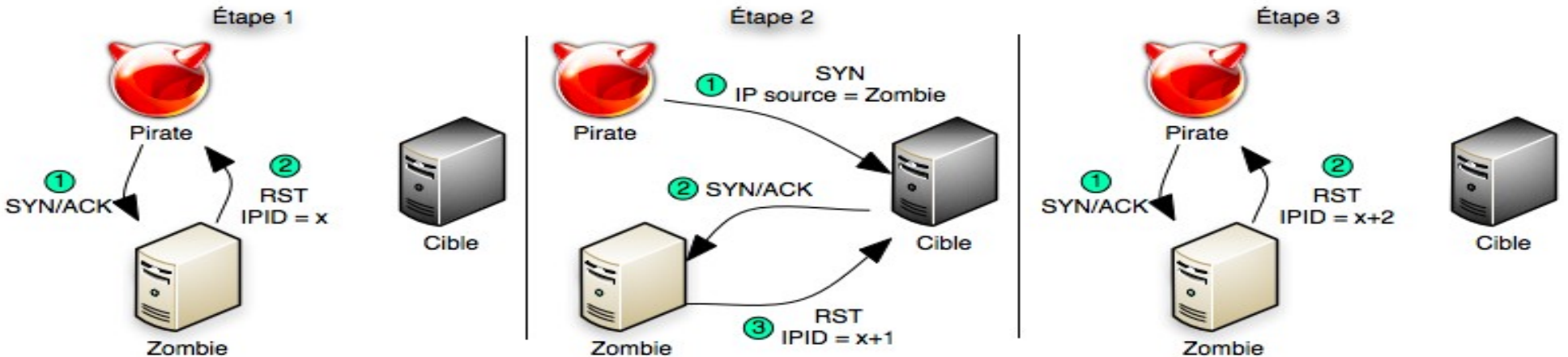


Figure 5.6. Idle scan of an open port

Step 1: Probe the zombie's IP ID.

The attacker sends a SYN/ACK to the zombie. The zombie, not expecting the SYN/ACK, sends back a RST, disclosing its IP ID.

Step 2: Forge a SYN packet from the zombie.

The target sends a SYN/ACK in response to the SYN that appears to come from the zombie. The zombie, not expecting it, sends back a RST, incrementing its IP ID in the process.

Step 3: Probe the zombie's IP ID again.

The zombie's IP ID has increased by 2 since step 1, so the port is open!

https://nmap.org/book/idlescan.html

Figure 5.7. Idle scan of a closed port

Step 1: Probe the zombie's IP ID.

The attacker sends a SYN/ACK to the zombie. The zombie, not expecting the SYN/ACK, sends back a RST, disclosing its IP ID. This step is always the same.

Step 2: Forge a SYN packet from the zombie.

The target sends a RST (the port is closed) in response to the SYN that appears to come from the zombie. The zombie ignores the unsolicited RST, leaving its IP ID unchanged.

Step 3: Probe the zombie's IP ID again.

The zombie's IP ID has increased by only 1 since step 1, so the port is not open.

Assuming a 50%/50% chance of the target's port being open and no noise (*i.e.*, the zombie is idle), what's the mutual information between the port status and the IPID the attacker sees in the last step?

H(X) is the entropy of the port status
H(Y) is the entropy of the IPID

Mutual information
$I(X;Y) = H(X) – H(X|Y)$

$1 = 1 - 0$

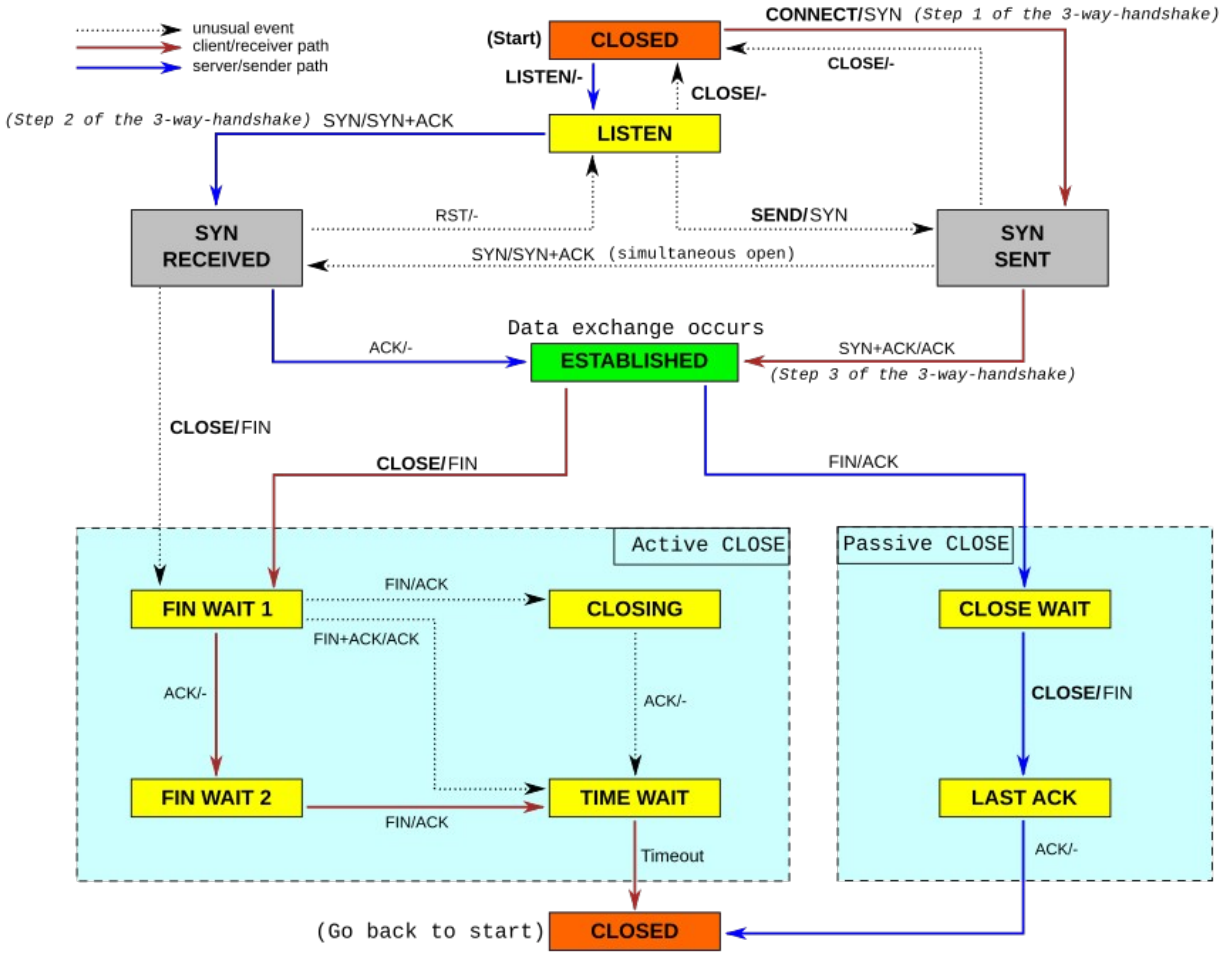
More examples of network side channels...

- DoS and SYN backlog basics
  - A side channel based on the SYN backlog
- Counting packets off-path
- Blind off-path TCP hijacking

DoS in general

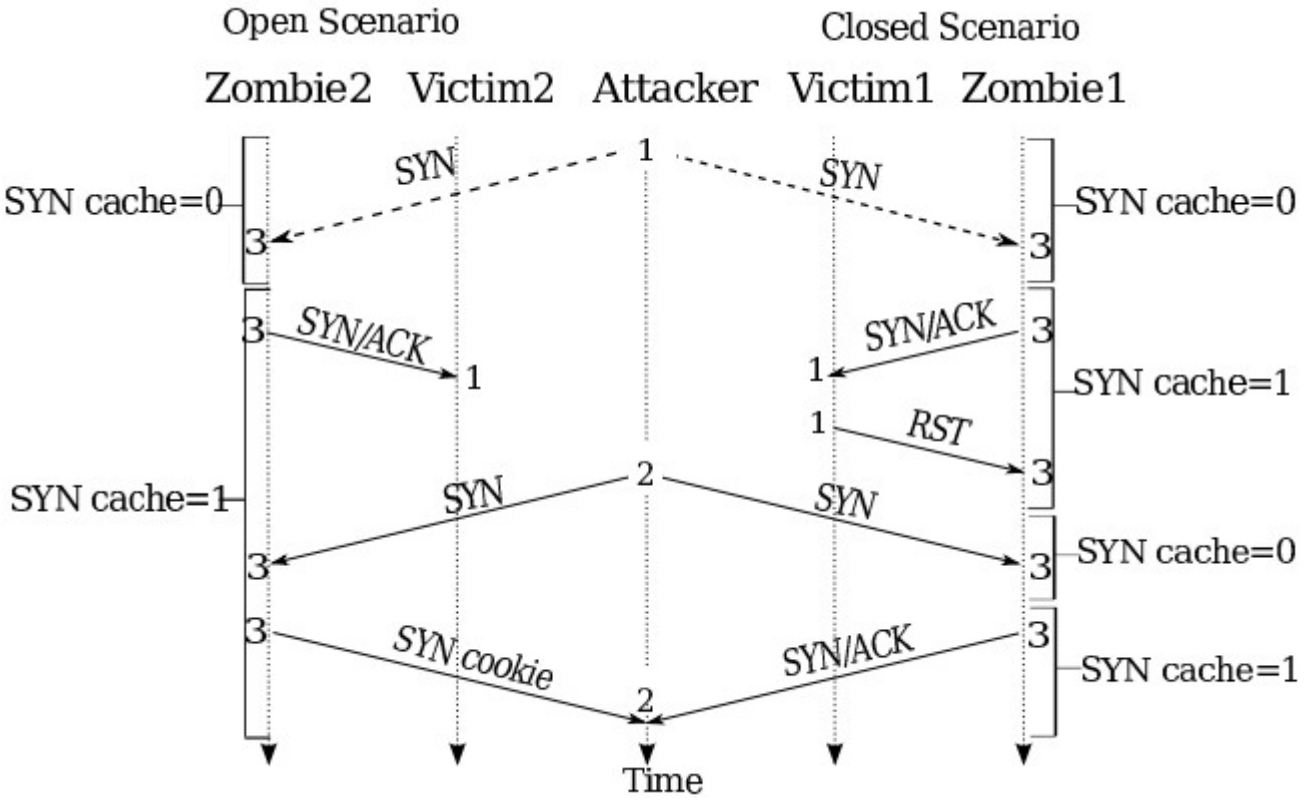
- Exhaust some kind of resource, *e.g.*:
  - Optimistic ACK to exhaust bandwidth
    - See https://homes.cs.washington.edu/~tom/pubs/CCR99.pdf
  - PING of death (*e.g.*, large PING) causes crash
  - Exhaust CPU in layer 7
  - More examples: http://www.isi.edu/~mirkovic/bench/attacks.html
  - SYN flood: Older hosts had either a fixed amount of half-open connections they could keep track of or no limitations at all; attack is to send lots of SYNs and never ACK or RST
    - Defenses: SYN backlog policies and SYN cookies

https://commons.wikimedia.org/wiki/File:Tcp_state_diagram_fixed_new.svg

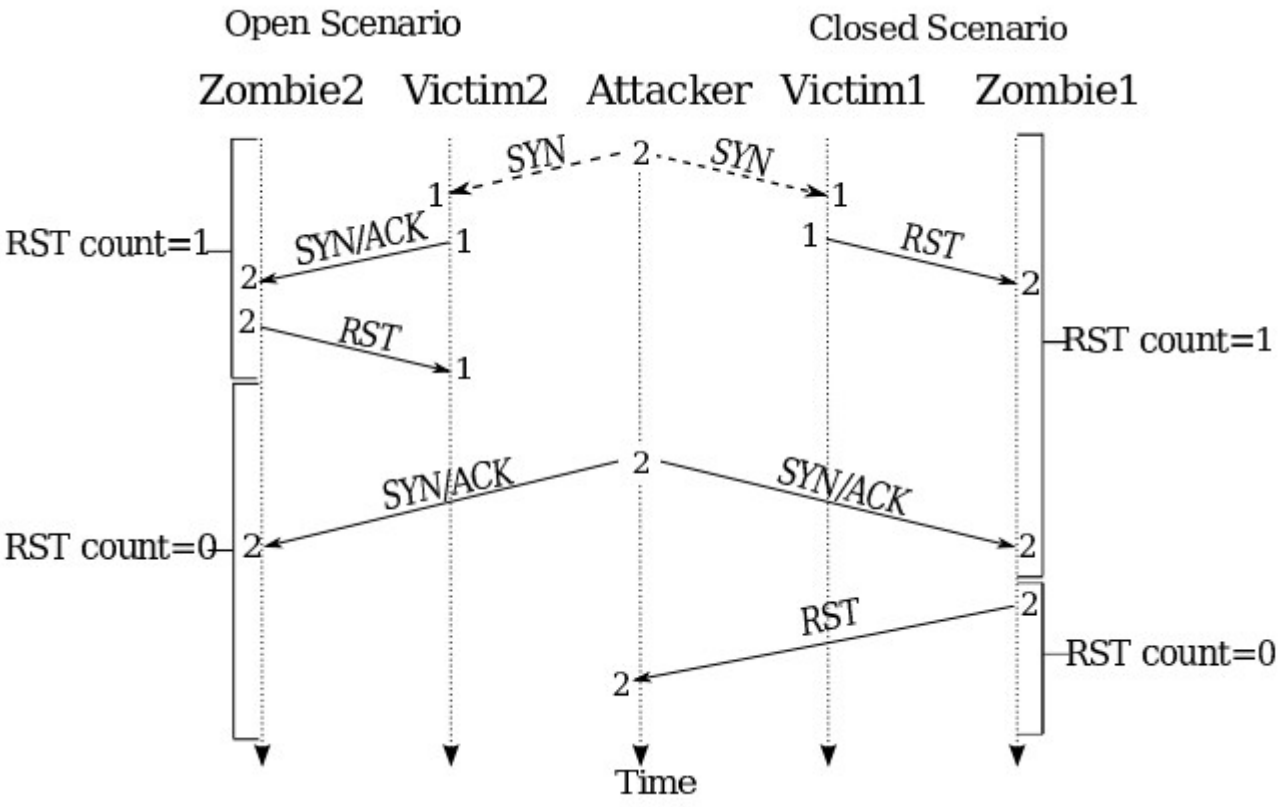


SYN cookies and SYN backlogs

- SYN cookies
  - Special kind of SYN/ACK
  - See https://cr.yp.to/syncookies.html
  - Can confirm ACK number and reconstruct the necessary state for a connection without having kept any state after sending the SYN cookie
    - Tuple info (source and destination IP addresses and ports) are hashed
- SYN backlog examples
  - Linux reserves ½, ¼, 1/8th, and so on for successively older SYNs, prunes 5 times a second
  - FreeBSD has 512 buckets of 30, you can't predict what bucket you fall into (in theory)



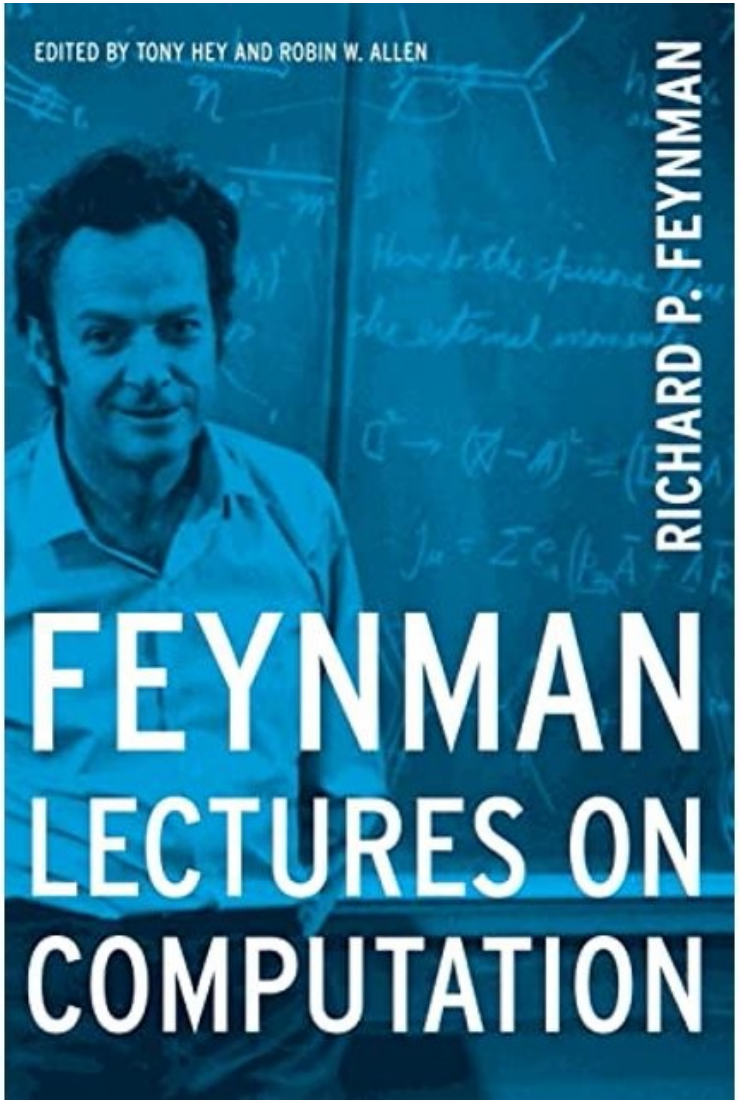
From… https://jedcrandall.github.io/usenix10.pdf



From… https://jedcrandall.github.io/usenix10.pdf

References

- *NMAP NETWORK SCANNING*, by Gordon “Fyodor” Lyon
- Google “nmap”, “idle scan”, etc.
- Other references were linked to inline


EDITED BY TONY HEY AND ROBIN W. ALLEN
RICHARD P. FEYNMAN
FEYNMAN
LECTURES ON
COMPUTATION