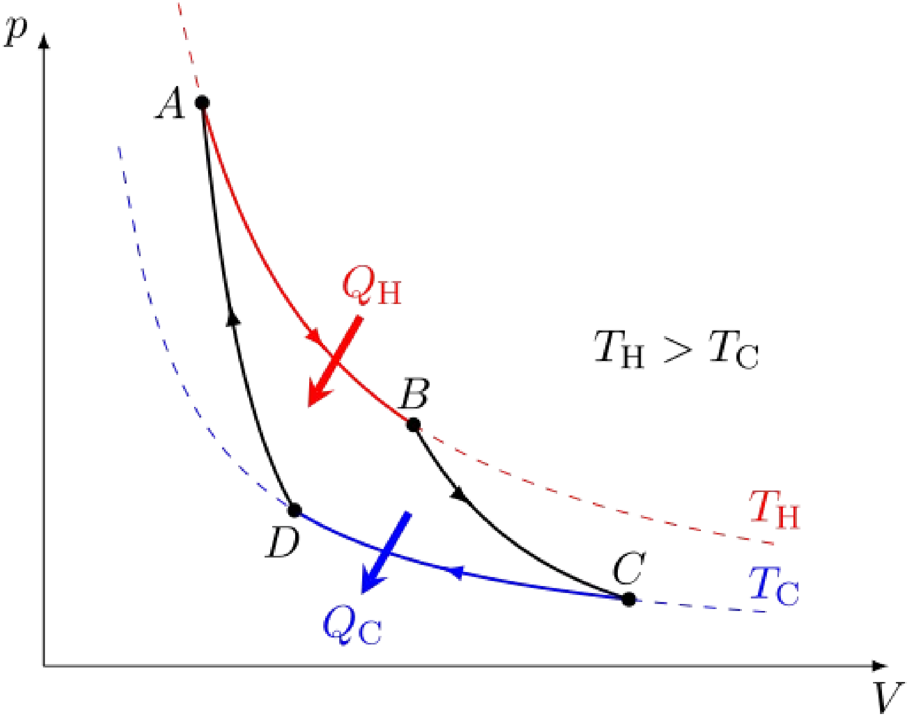
Port scanning and network side channels

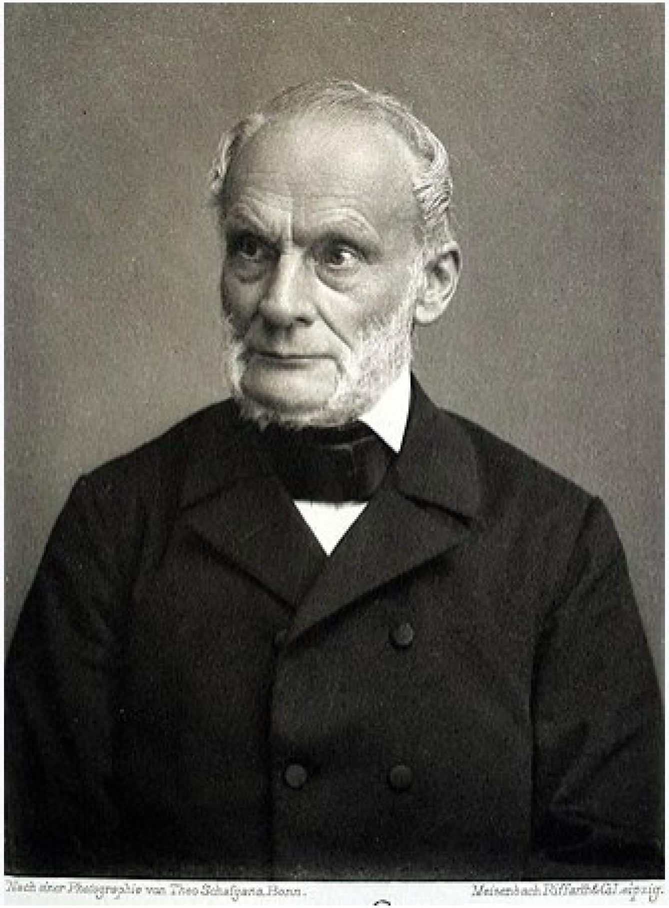
CSE 468 Fall 2025

jedimaestro@asu.edu



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

Rudolf Clausius


Nach einer Photographie von Theo Schafgans.Bonn.
Meisenbach Riffarth&Co Leipzig.

“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

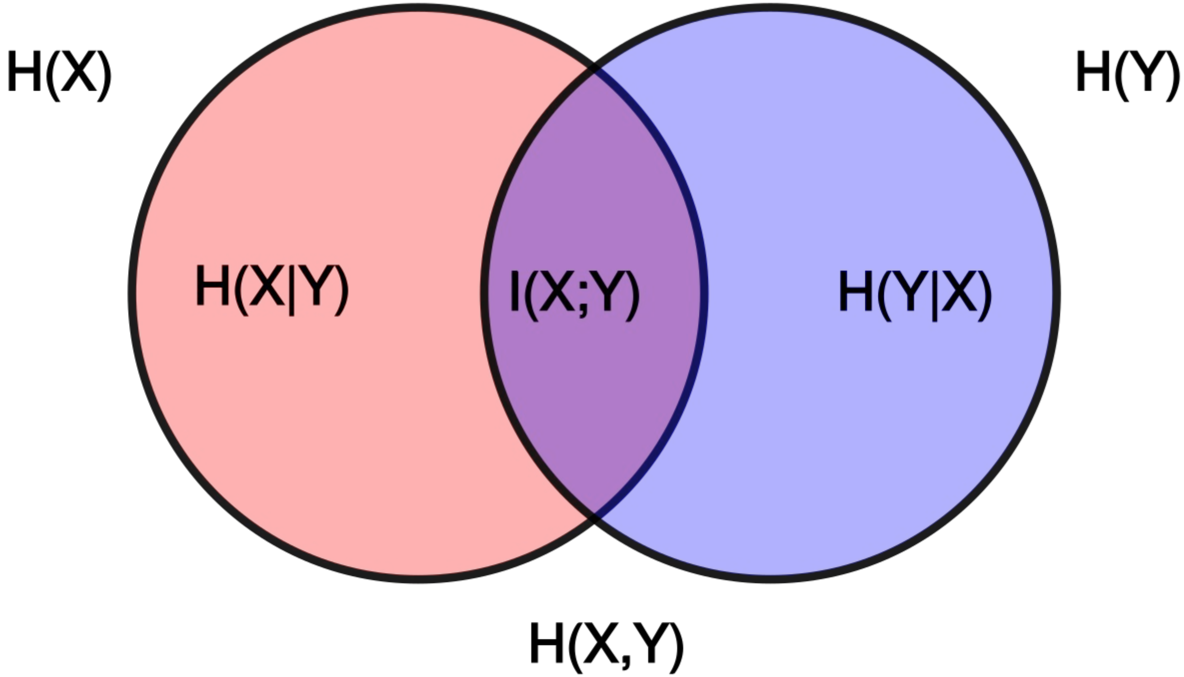
Entropy

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

$$H = -\sum_{i} p_i \log_2(p_i)$$

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

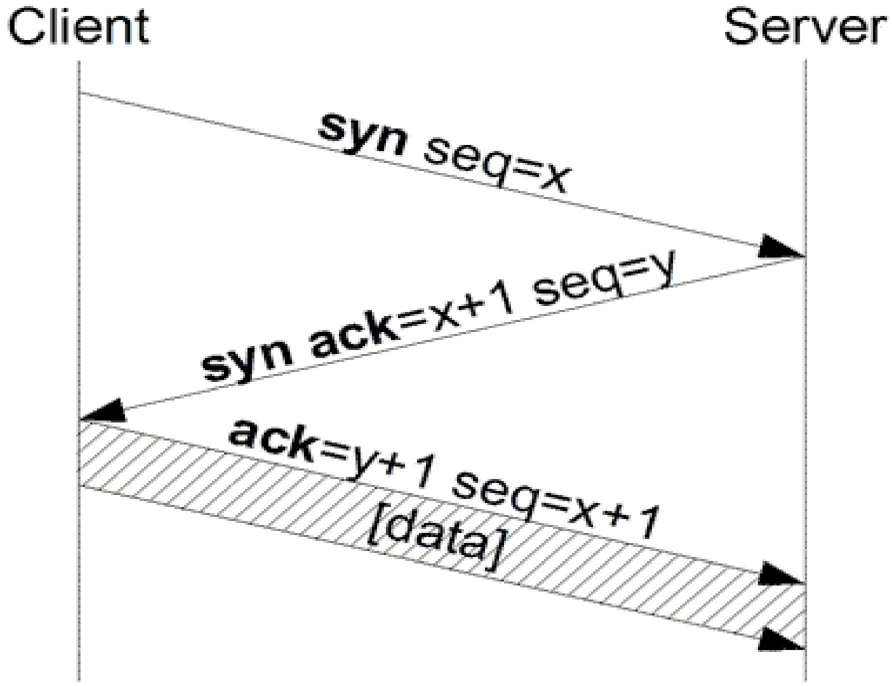


Image from Wikipedia

Open port == listening

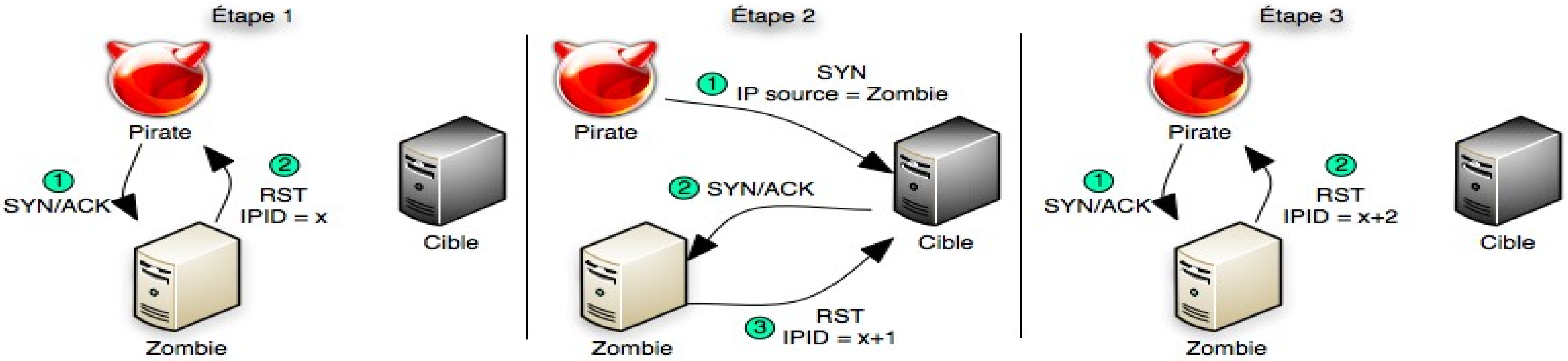
- If you send a SYN packet to port 80 (the HTTP port) on a remote host and that host replies with a SYN/ACK, then we say that port 80 on that machine is “open”
  - In this example, that probably means it's a web server
- If it responds with a RST, we say it's “closed”
- If there is evidence of filtering (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 detection (research on your own)
- Stealth, info about middleboxes, etc.

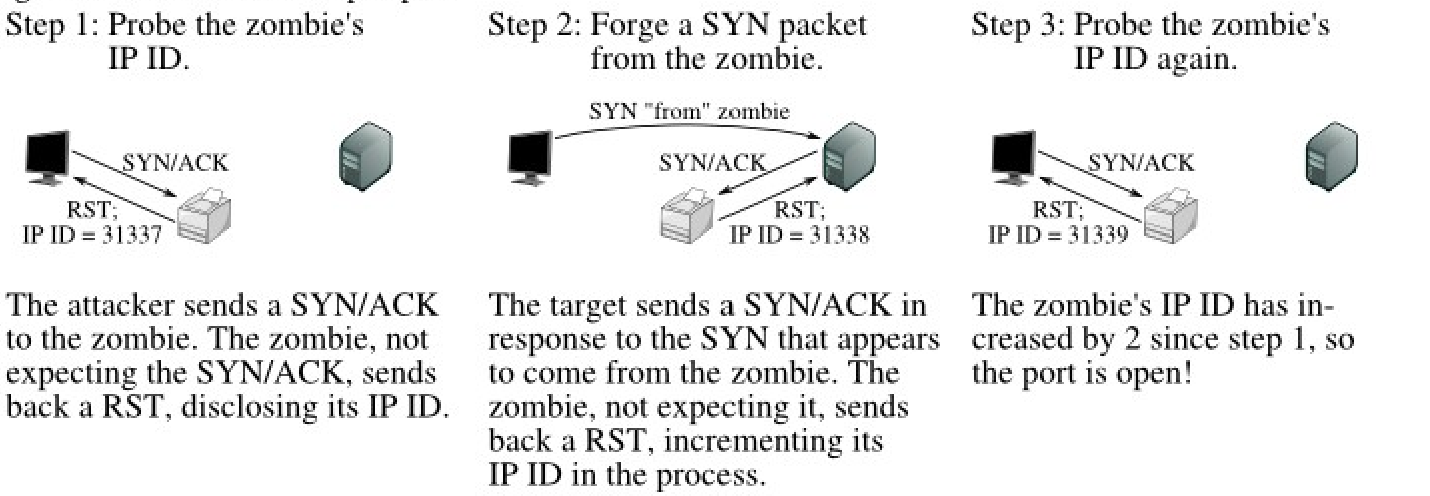
Idle scan

- 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



Par Shiningfm — Travail personnel, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=8273541

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.

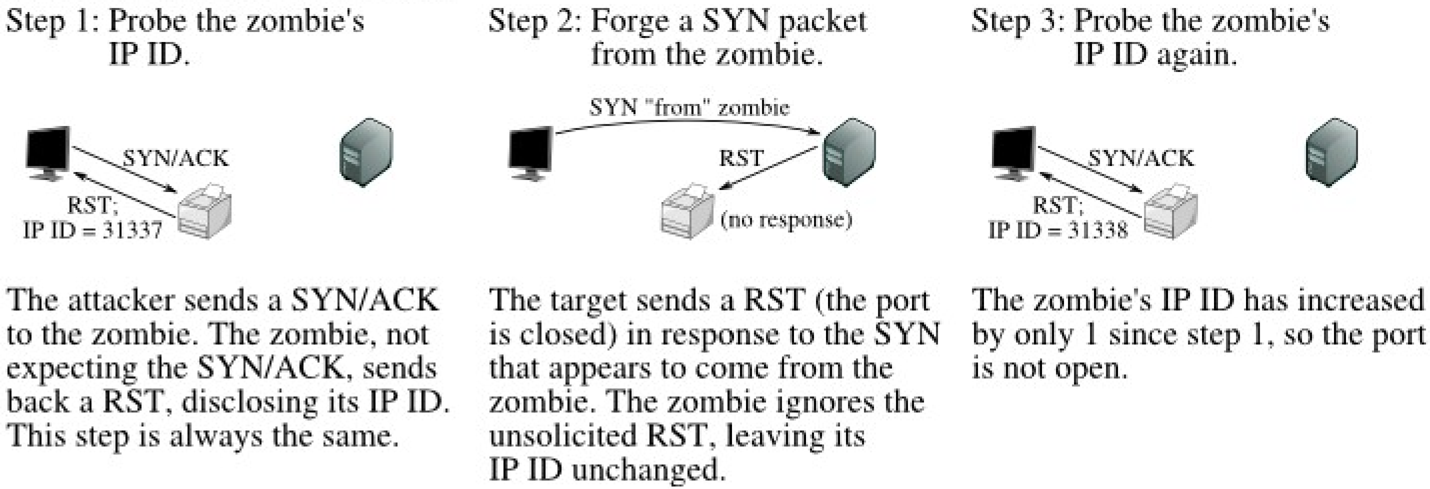
The target sends a SYN/ACK in response to the SYN that appears to come from the zombie. The zombie, not expecting it, sends back a RST, incrementing its IP ID in the process.

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

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

$$1 = 1 - 0$$

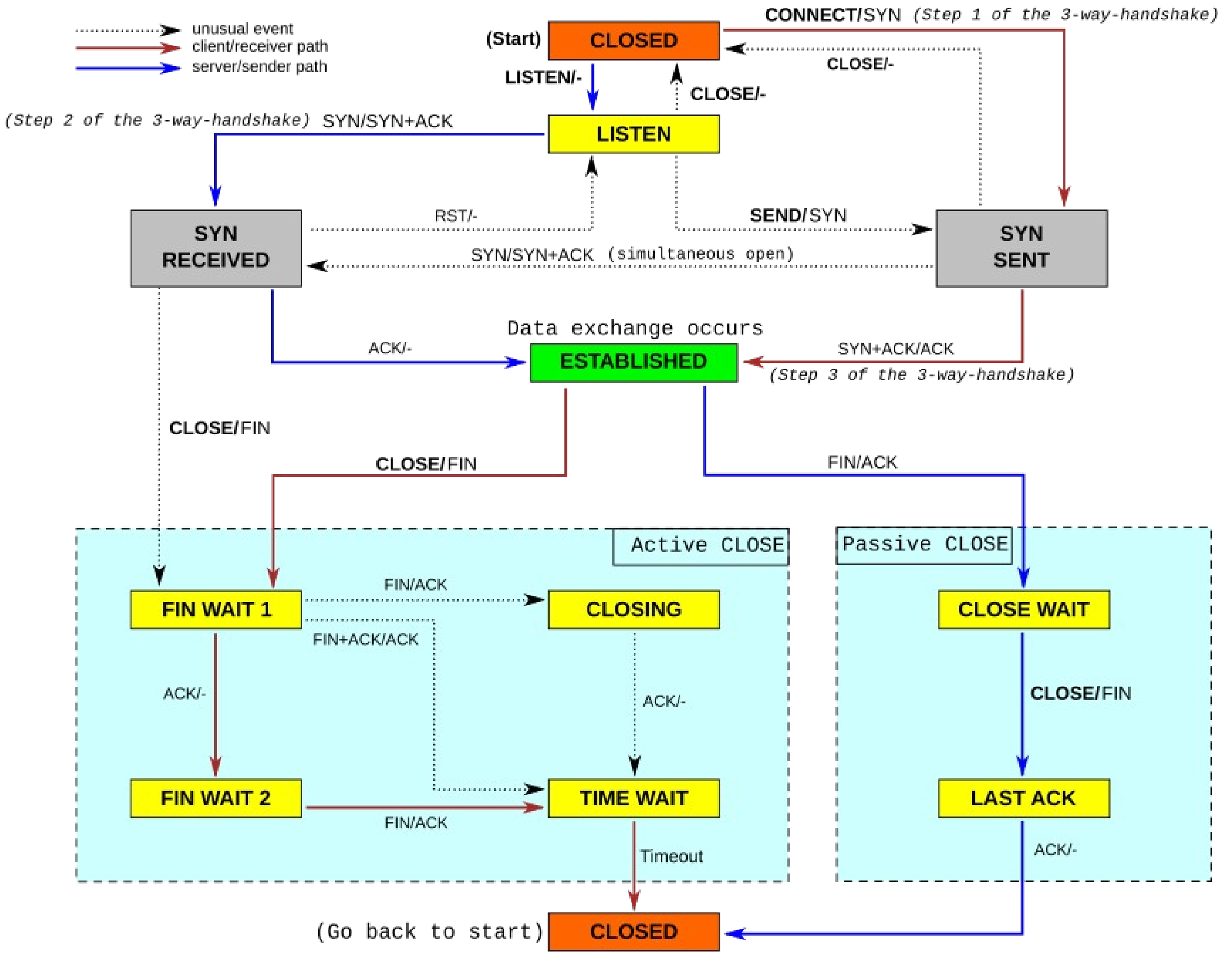
More examples of network side channels...

- 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

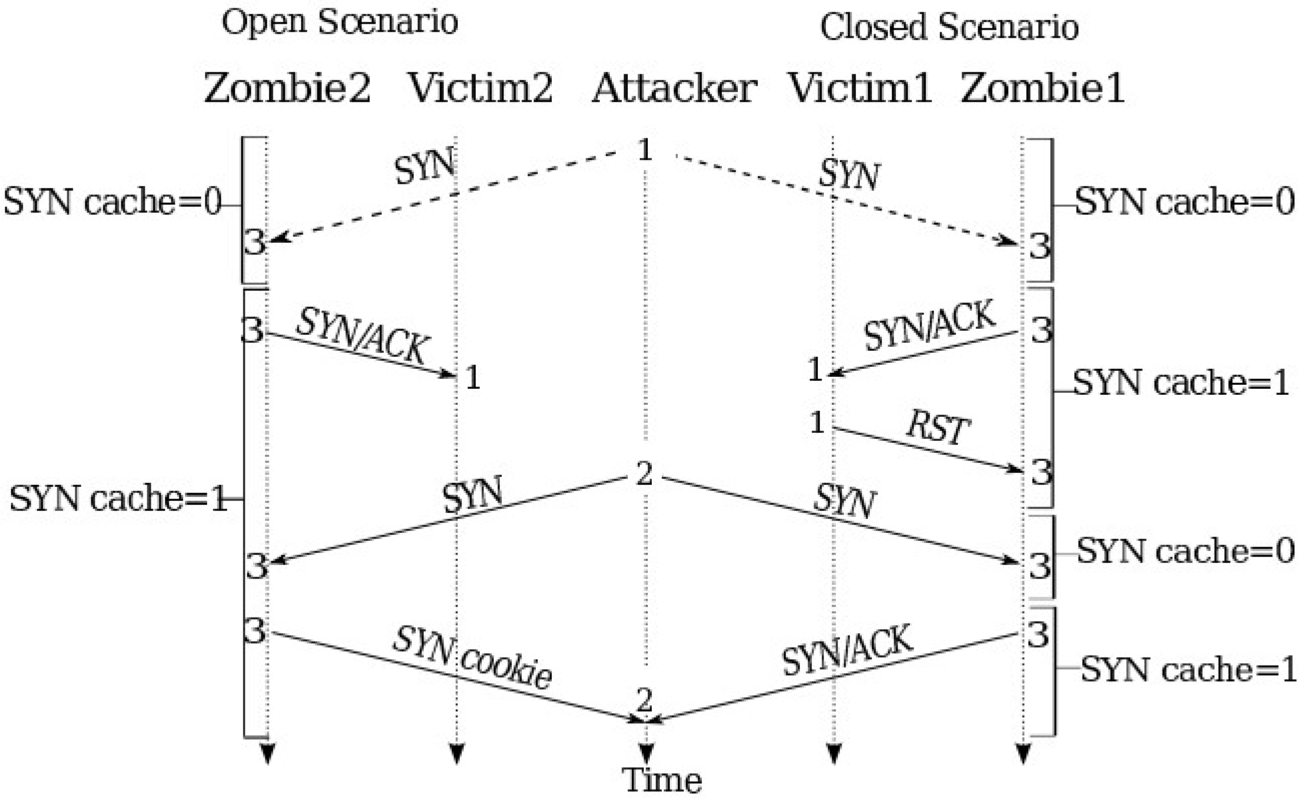
- Exhaust some kind of resource, *e.g.*:
  - Optimistic ACK to exhaust bandwidth
    - See https://homes.cs.washington.edu/~tom/pubs/CCR99.pdf
  - PING of death (*e.g.*, large PING) causes crash
  - Exhaust CPU in layer 7
  - More examples: http://www.isi.edu/~mirkovic/bench/attacks.html
  - SYN flood: 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

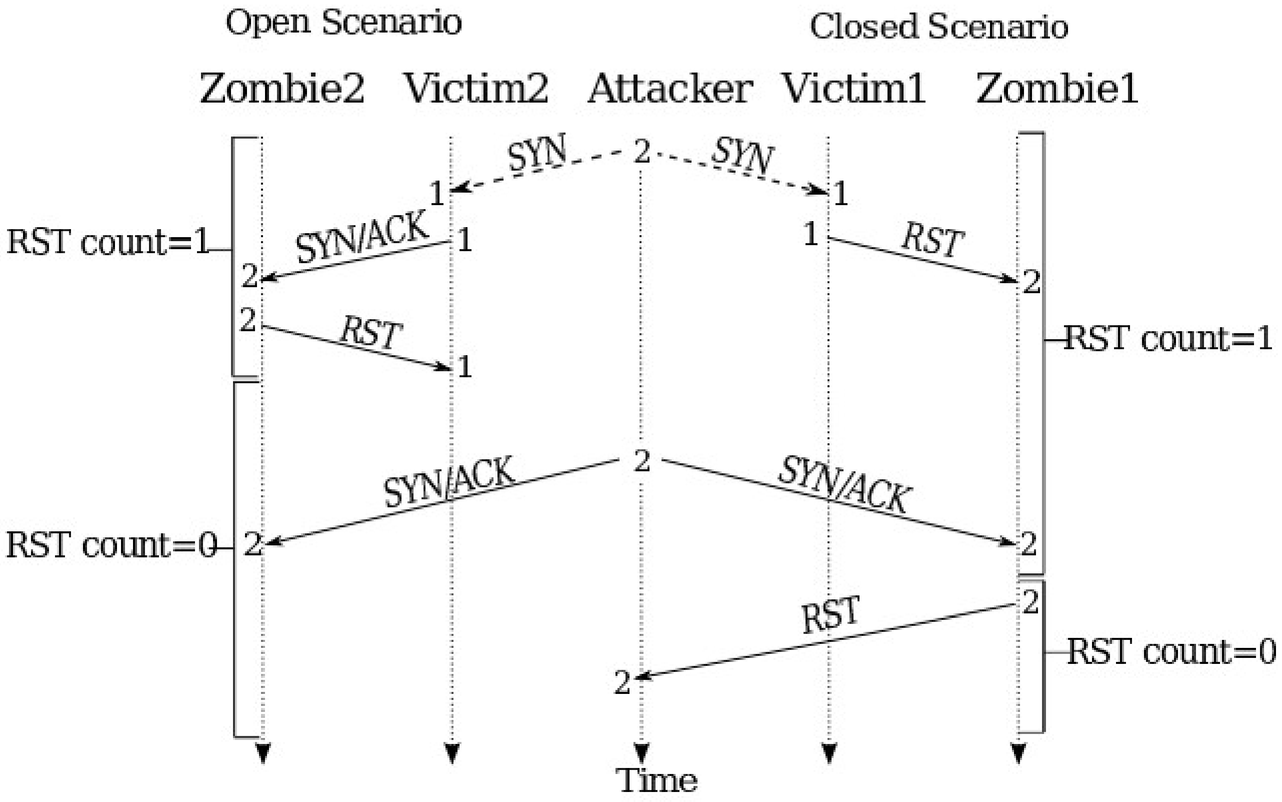


SYN cookies and SYN backlogs

- 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)



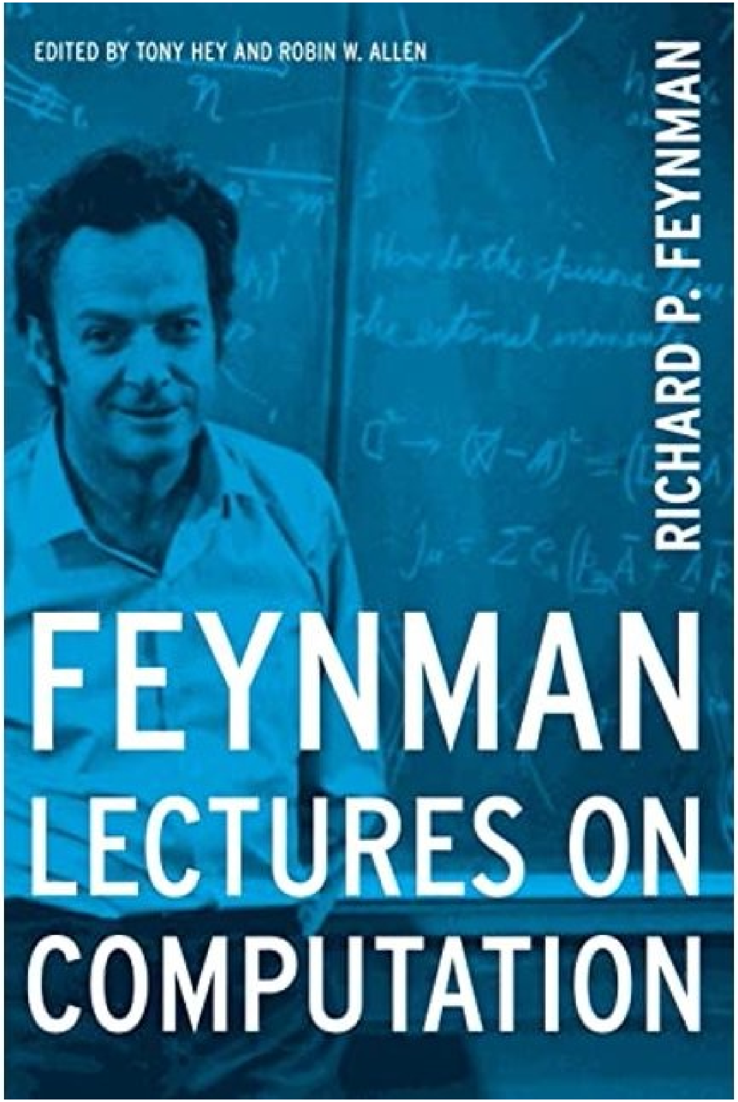
From… https://jedcrandall.github.io/usenix10.pdf



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