
SECURITY (COMP0141): NETWORK SECURITY



HOW DOES THE INTERNET WORK?

goal: get Alice to that website!



`http://me.bob.com/hi.html`

HOW DOES **HTTP** WORK?

goal: get Alice to that website!



`http://me.bob.com/hi.html`

INTERNET PROTOCOL SUITE

Application layer: SMTP, FTP, SSH, **HTTP**, etc.

Transport layer: host-to-host communications (UDP, TCP, etc.)

Internet layer (IP): End-to-end routing of data packets

Link layer: Transmission of data within a local network (Ethernet)

Physical layer: Transmission of raw bits over a physical link

HOW DOES THE INTERNET WORK?

goal: get Alice to that website!

get IP address for me.bob.com
("domain name resolution")

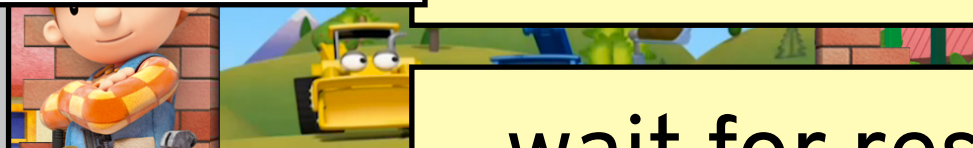
application layer

transport layer (TCP) send GET request to IP address hi.html

Internet layer (routing via the internet backbone)

Internet layer

wait for response from IP address,
then render hi.html and enjoy



HOW DOES **TCP/IP** WORK?

Application layer: SMTP, FTP, SSH, **HTTP**, etc.

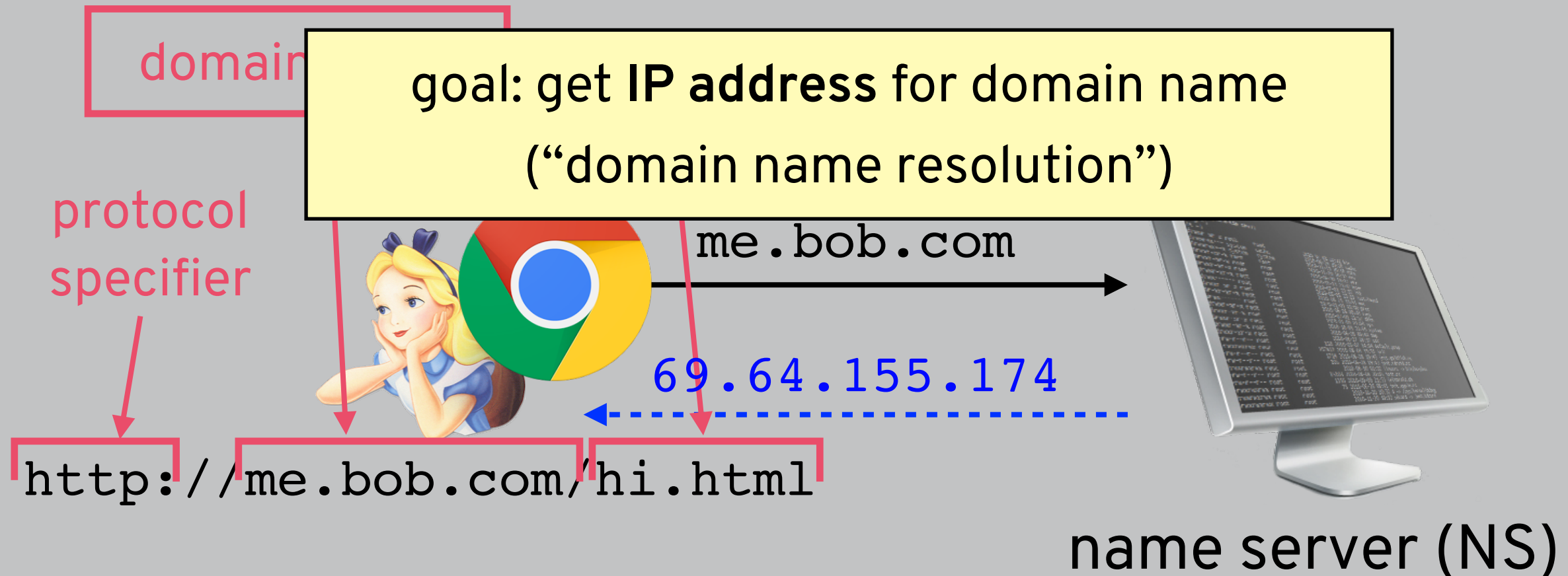
Transport layer: host-to-host communications (UDP, TCP, etc.)

Internet layer (IP): End-to-end routing of data packets

Link layer: Transmission of data within a local network (Ethernet)

Physical layer: Transmission of raw bits over a physical link

STEP 1: FIND CONTENT HOST



DOMAIN NAME SYSTEM

FAQs

q: do we really do this every time we go to a website?

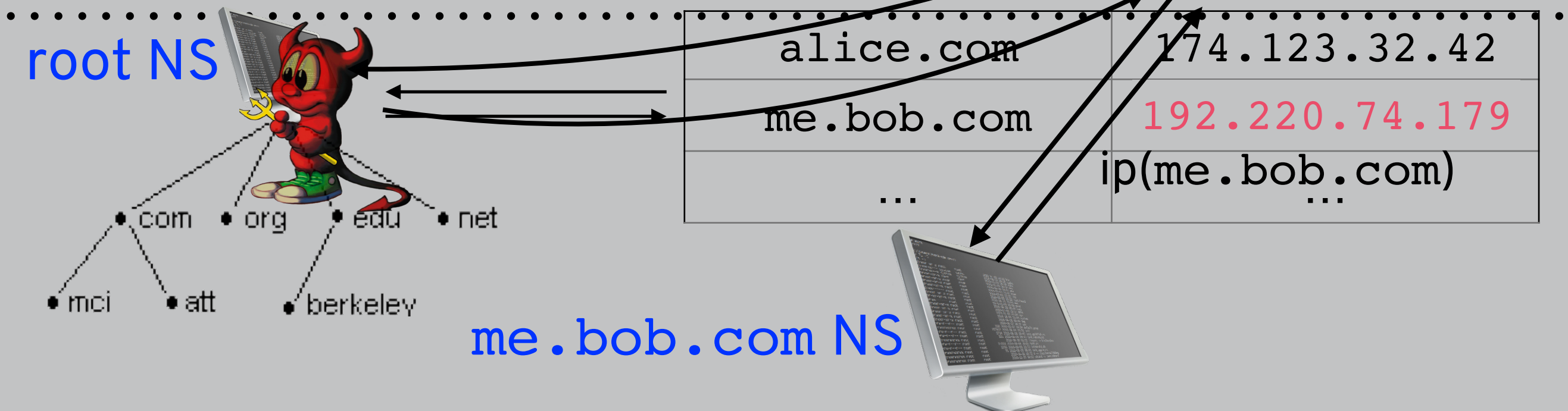
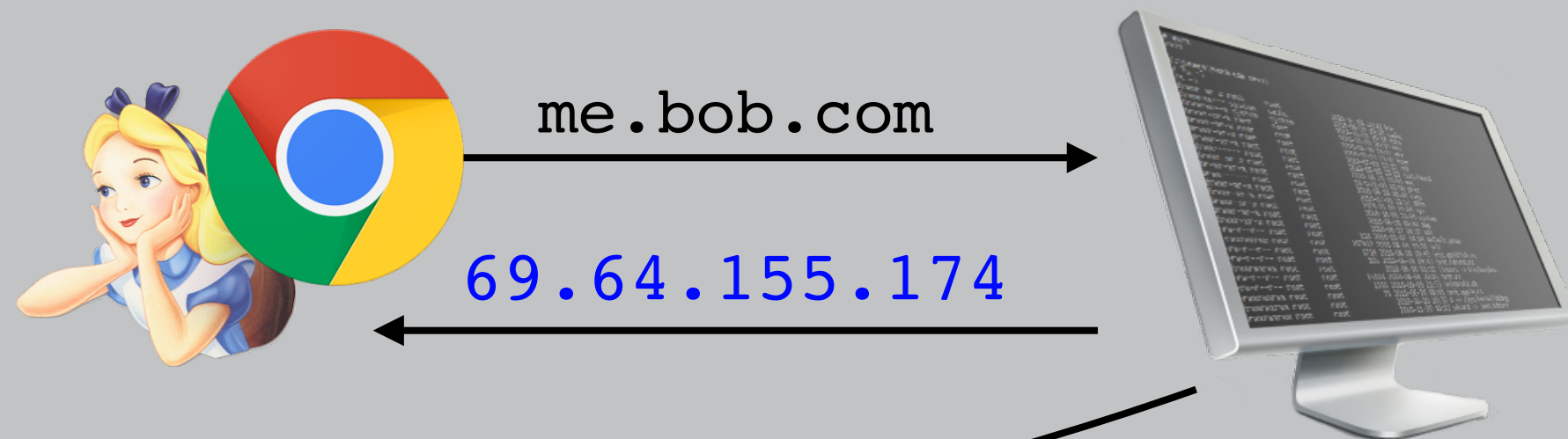
a: no! DNS results are cached by your browser.

DNS responses and negative queries are cached, and cached data expires according to TTL (time to live) provided by owner of data

These results are also cached by the resolver itself

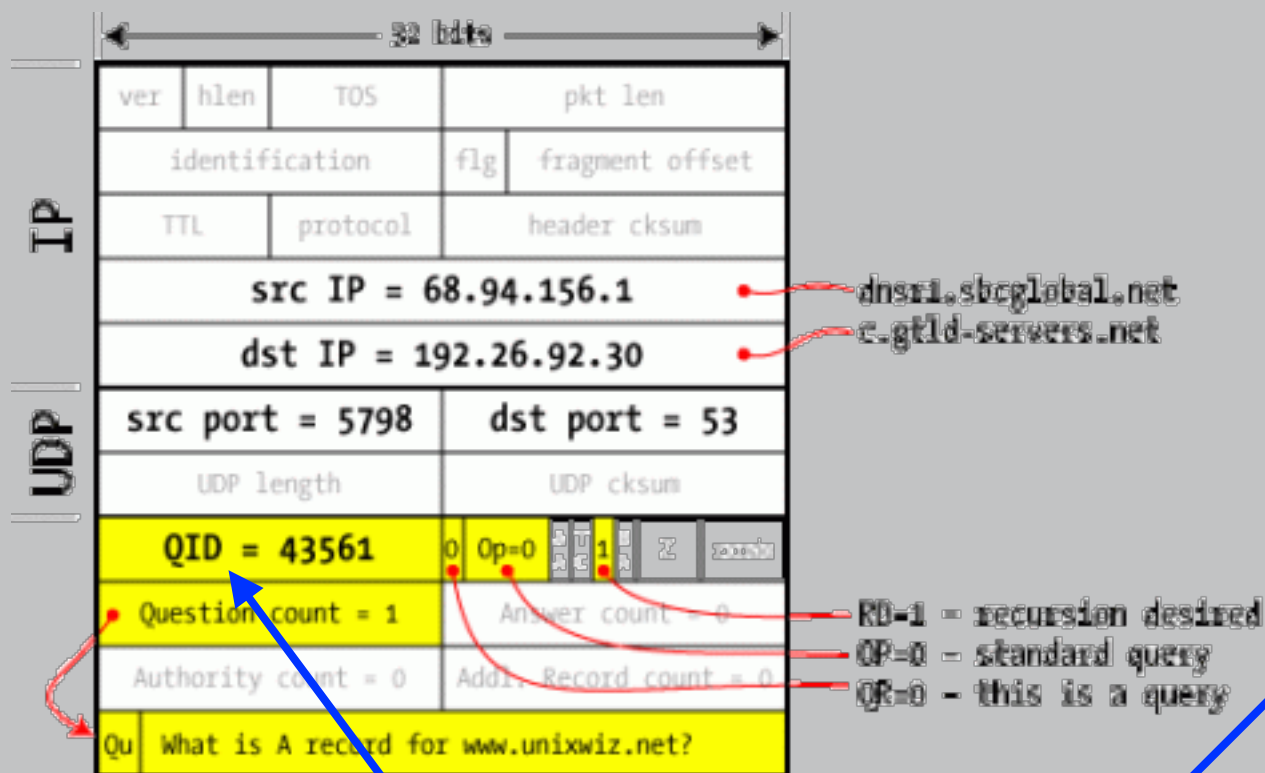
This opens up the potential for **DNS cache poisoning** because the NS does not validate that DNS entries are from authoritative sources

DNS SPOOFING

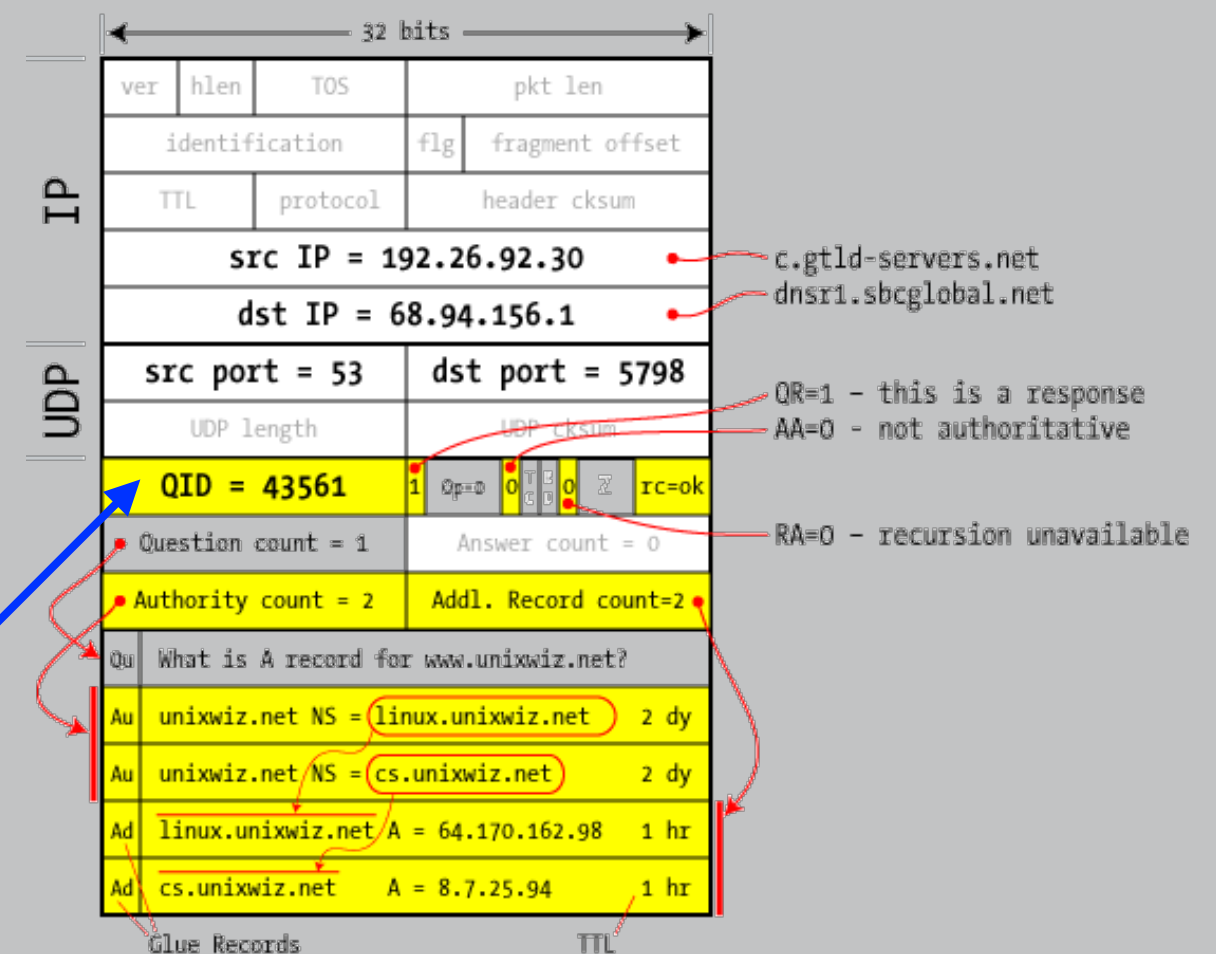


DNS SPOOFING

query packet



response packet

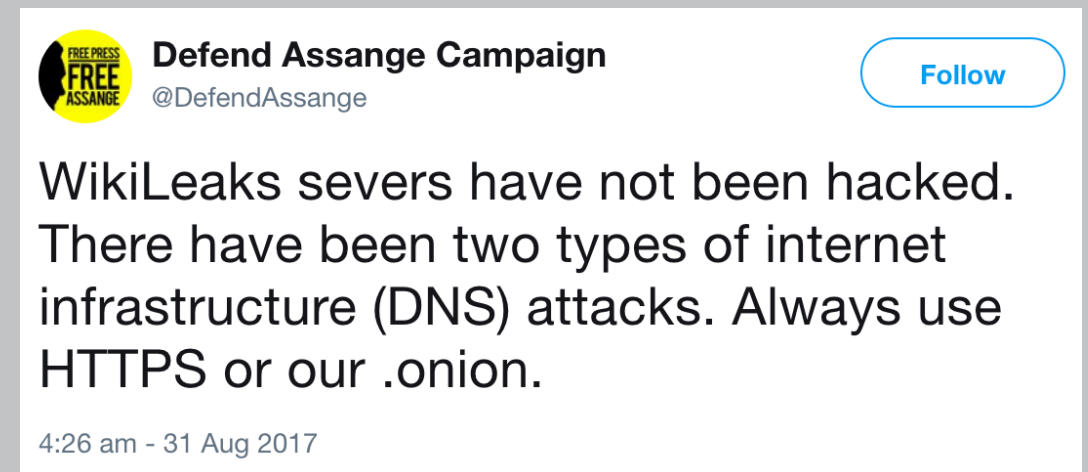


attacker needs these
values to match

DNS SPOOFING

Many examples of this being done in practice:

- 2000: `hilary2000.org` sent to `hilaryno.com`
- 2004: Google and Amazon sent to an online pharmacy
- 2016: all 36 of a Brazilian bank's domains sent to phishing sites ([pharming](#))
- 2017: Wikileaks visitors sent to attacker-controlled page

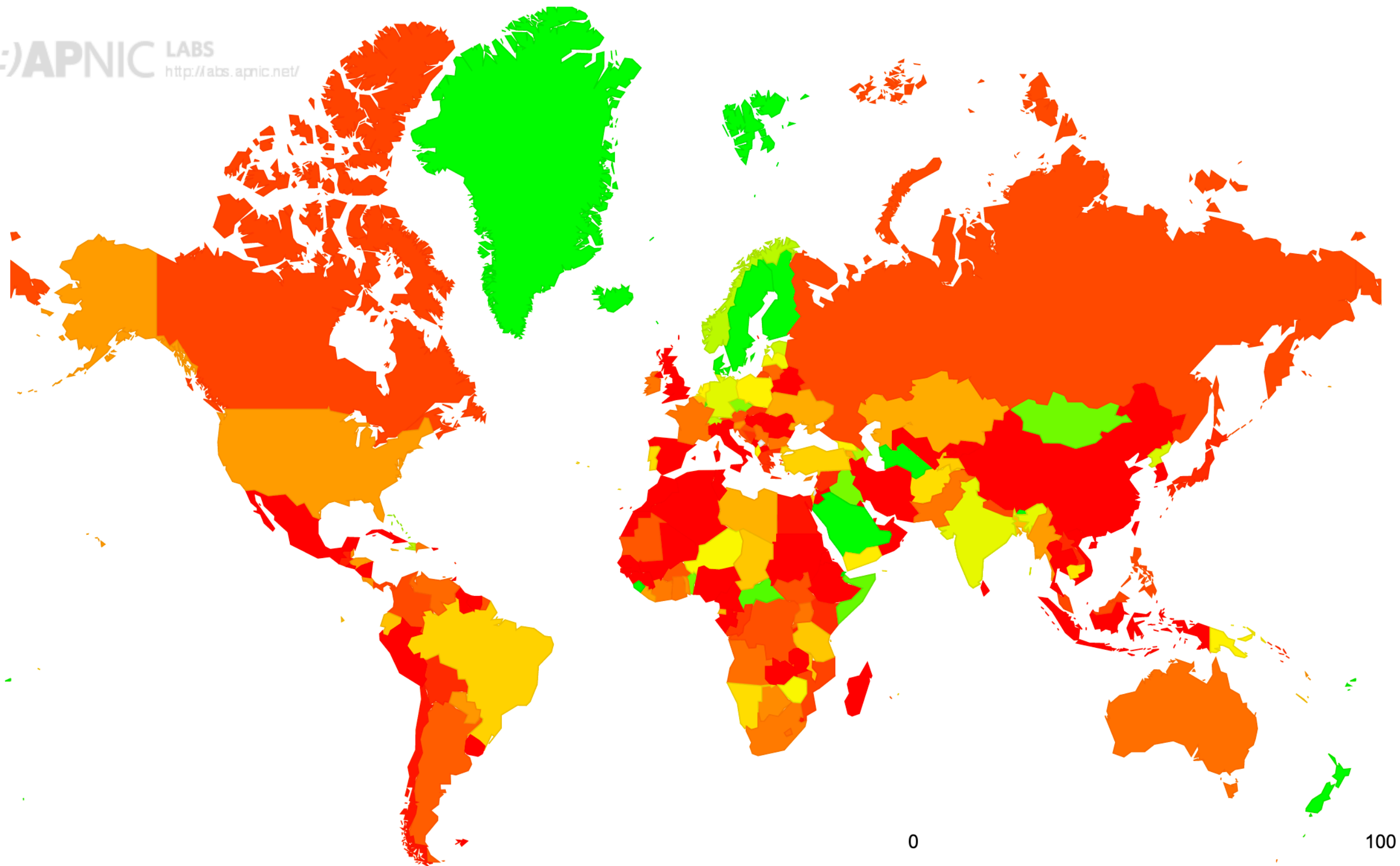


Solutions include:

- Randomising source port (more randomness for QID)
- Ignoring unnecessary responses
- **DNSSEC: authenticate responses with digital signature**

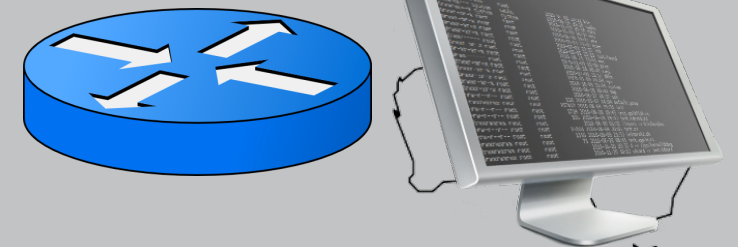
DNSSEC Validation Rate by country (%)

(::)APNIC LABS
<http://labs.apnic.net/>



STEP 2: REQUEST CONTENT

`http://me.bob.com/hi.html`
`69.64.155.174`

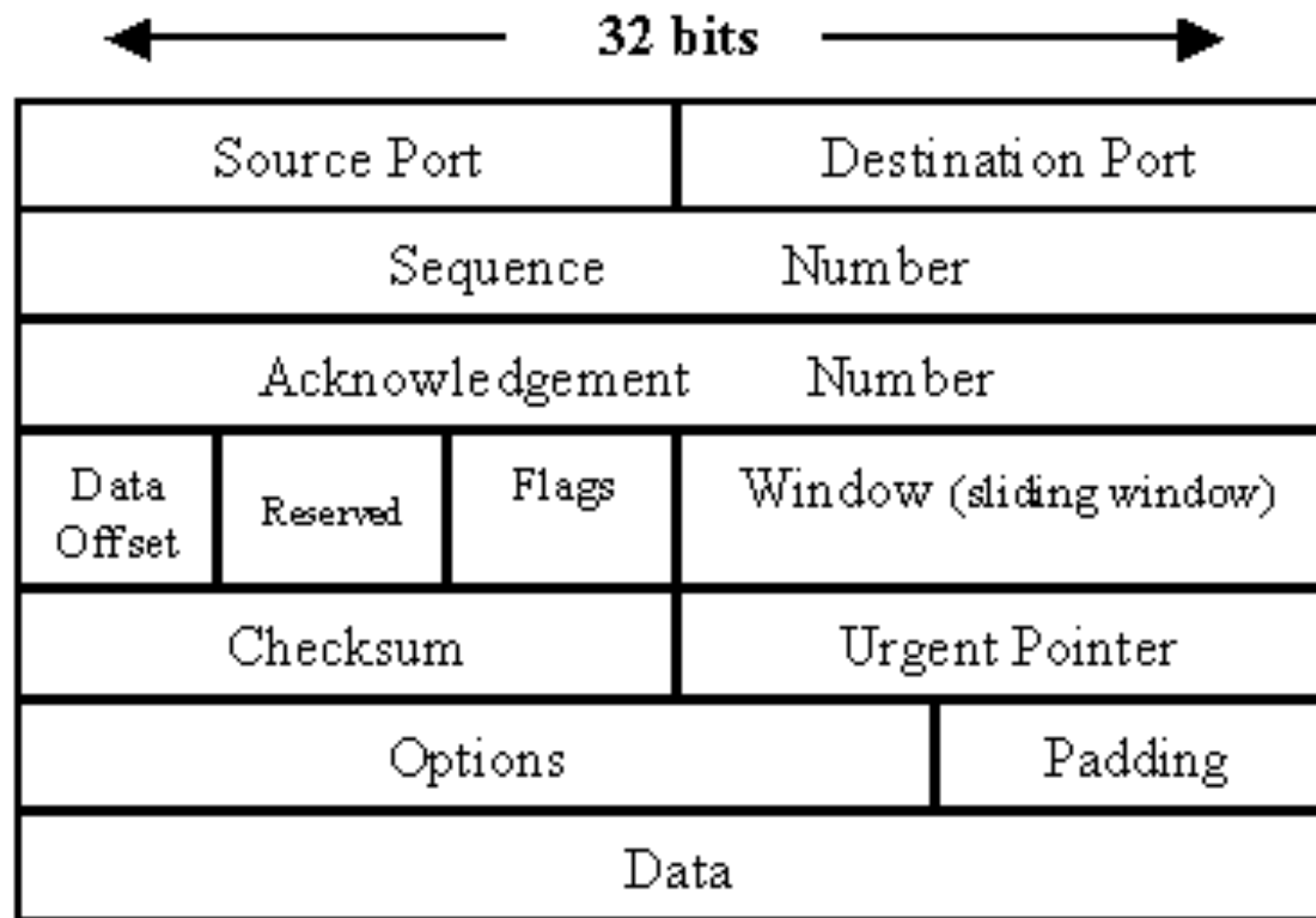


`me.bob.com`

IP PACKET

4-bit version	4-bit Header len	8-bit type of service	16-bit total length (in bytes)	
16-bit identification			3-bit flags	13-bit fragment offset
8-bit time to live (TTL)		8-bit protocol	16-bit header checksum	
Alice's IP address				
Bob's IP address				
Options (if any)				
“I want the content at hi.html”				

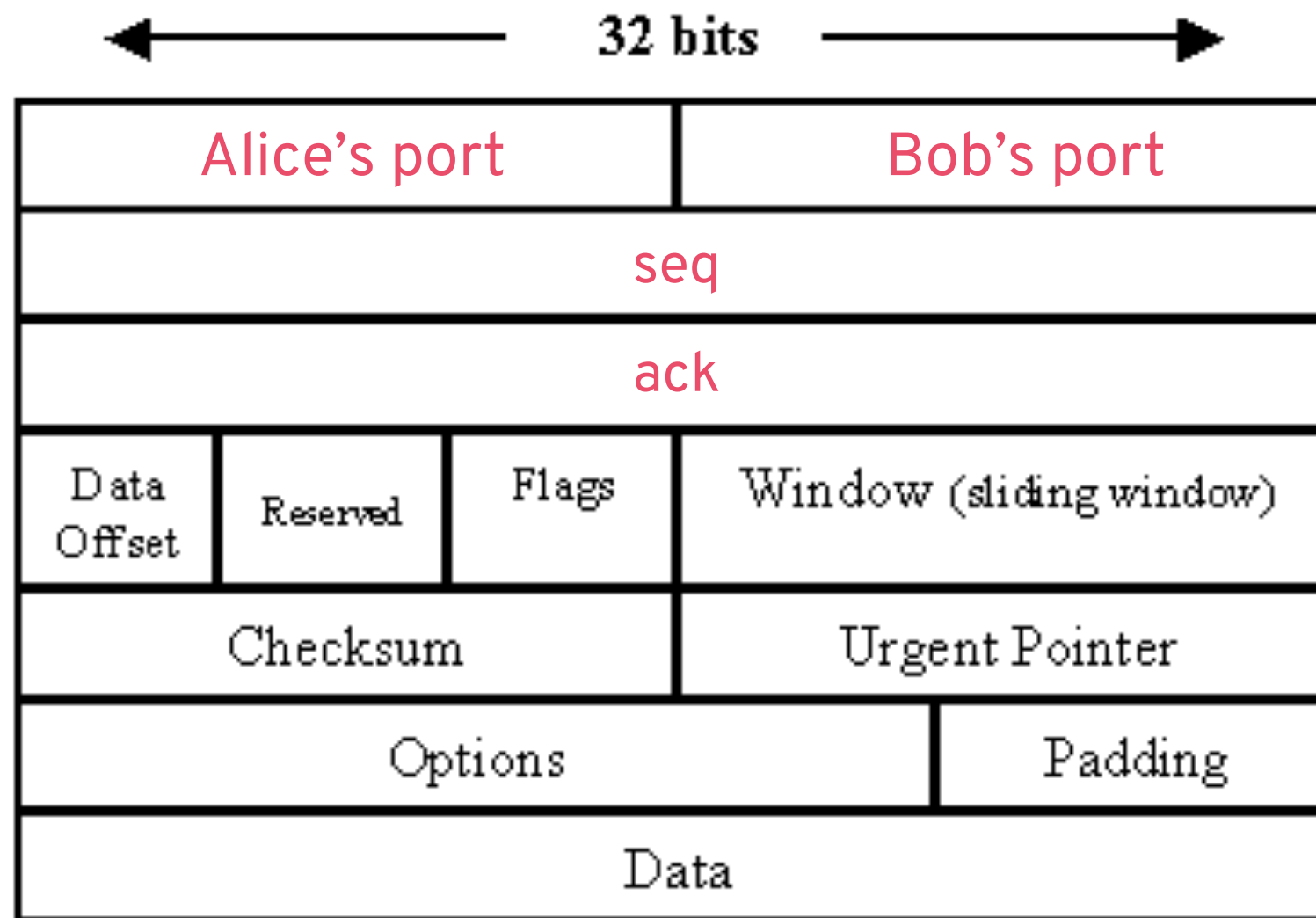
TCP PACKET



TCP

Used to establish a bi-directional **stateful** session between two endpoints identified by their IP address and **port**

TCP PACKET



TCP

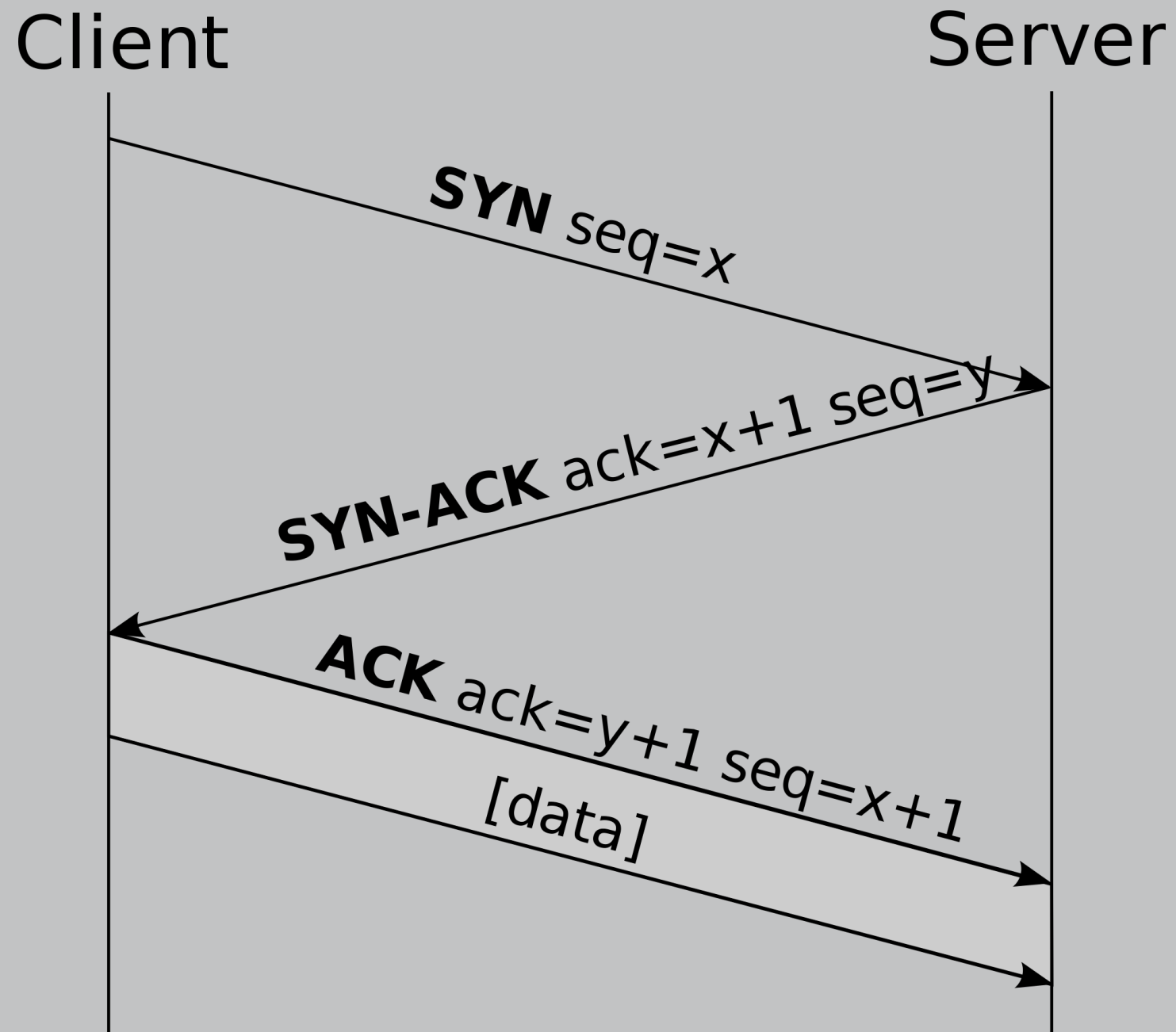
Used to establish a **stateful** bi-directional session between two endpoints identified by their IP address and **port**

- 22: SSH (remote access)
- 53: DNS
- 80: HTTP
- 443: HTTPS

Packets can contain special **flags**:

- SYN: I want to start a connection
- FIN: I want to close a connection
- ACK: I got your last packet

TCP HANDSHAKE



TCP

Used to establish a **stateful** bi-directional session between two endpoints identified by their IP address and **port**

- 22: SSH (remote access)
- 53: DNS
- 80: HTTP
- 443: HTTPS

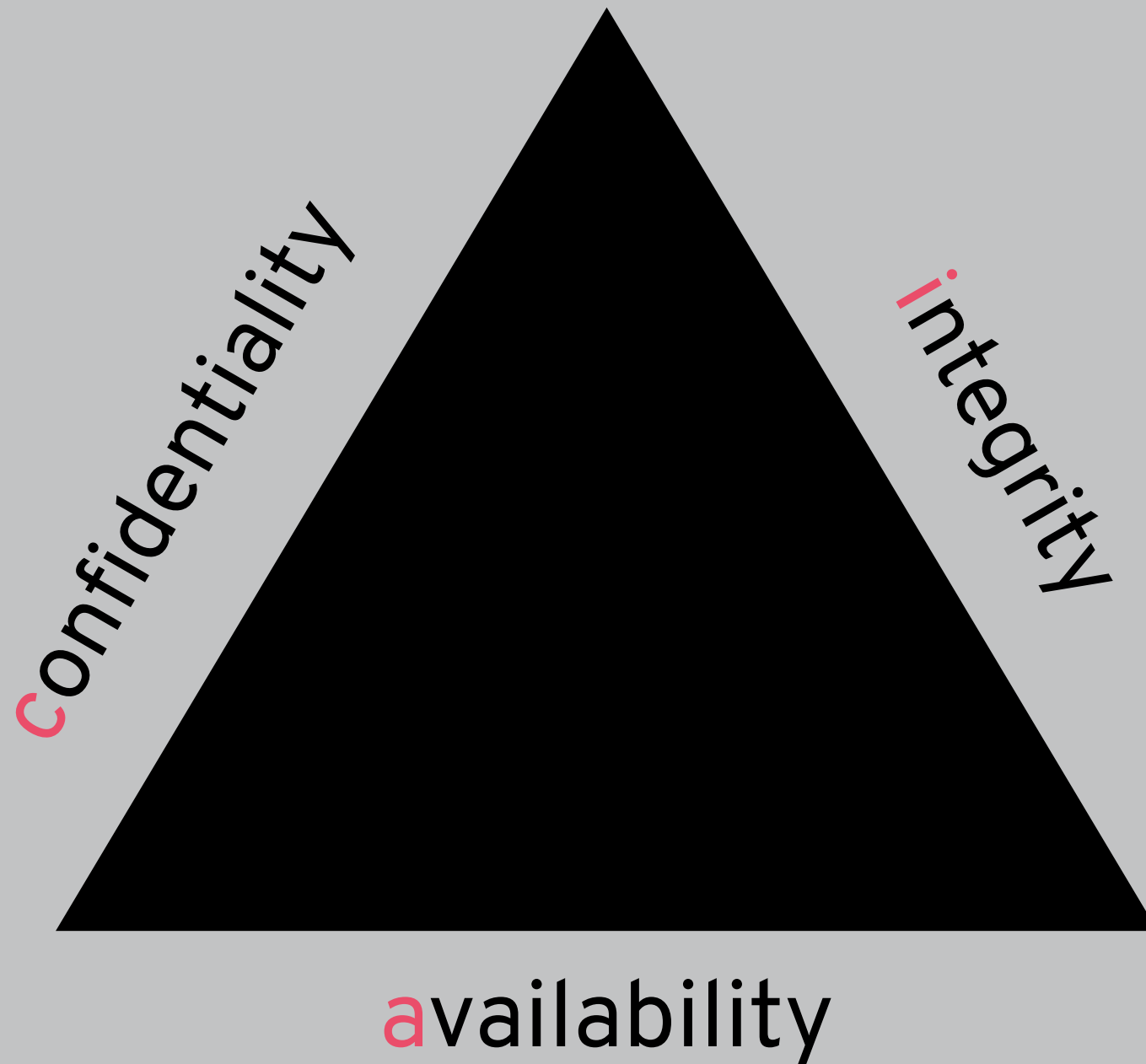
Packets can contain special **flags**:

- SYN: I want to start a connection
- FIN: I want to close a connection
- ACK: I got your last packet



TCP/IP **trust model** has evolved:

- 1970s: trusted network and trusted hosts
- 1980s: hosts may be compromised
- today: network may be compromised too

CIA TRIANGLE



CONFIDENTIALITY

4-bit version	4-bit Header len	8-bit type of service	16-bit total length (in bytes)	
16-bit identification			3-bit flags	13-bit fragment offset
8-bit time to live (TTL)	8-bit protocol		16-bit header checksum	
 <div>Bob's IP address</div> <div>Alice's IP address</div>				
Options (if any)				
<div><Content at hi.html (part 1 of N)></div> 				

anyone can read your web traffic

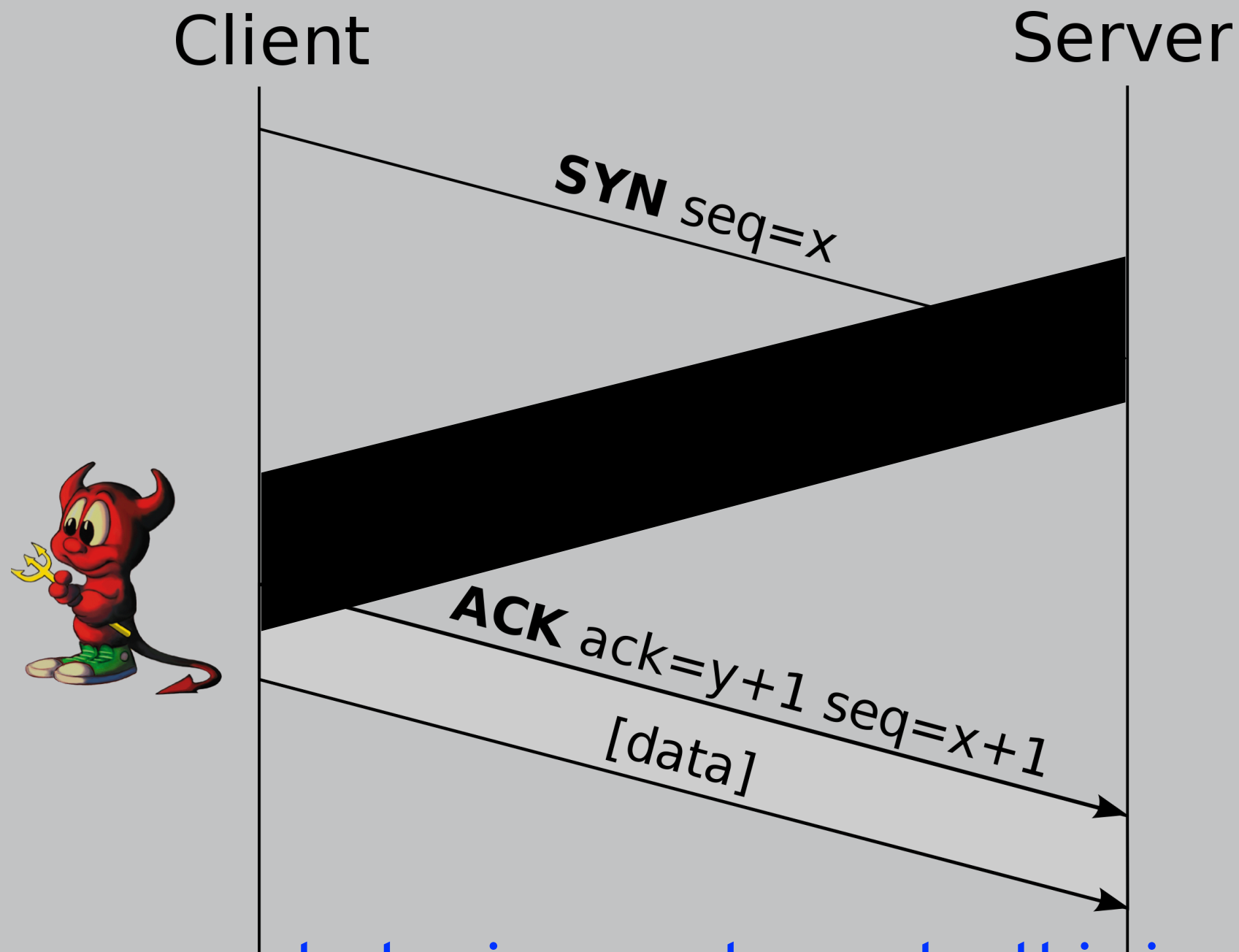
anyone can see who you're talking to

IP SPOOFING

4-bit version	4-bit Header len	8-bit type of service	16-bit total length (in bytes)	
16-bit identification			3-bit flags	13-bit fragment offset
8-bit time to live (TTL)		8-bit protocol	16-bit header checksum	
Professor Evil's Alice's IP address				
Bob's IP address				
Options (if any)				
“I want the content at hi.html”				

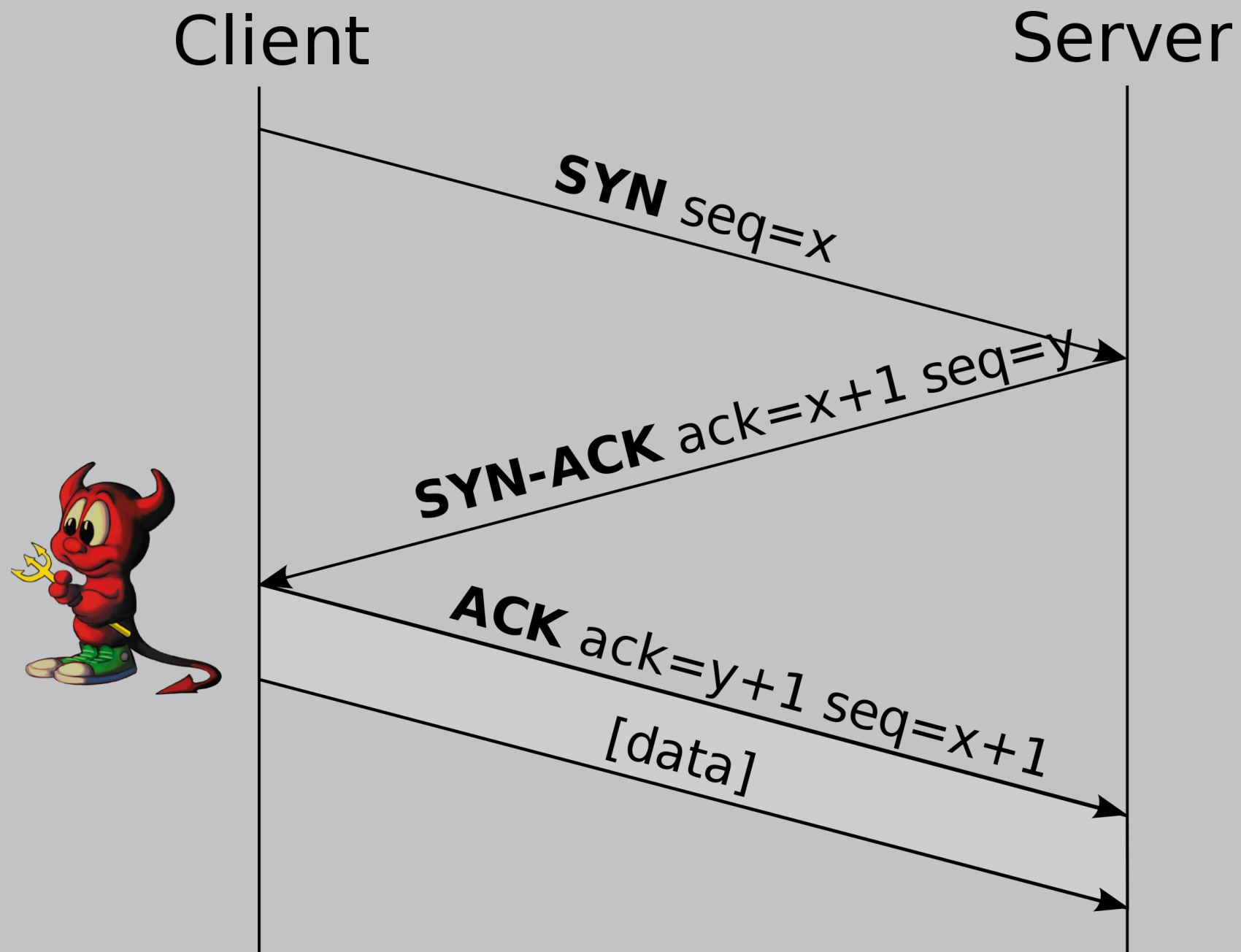
anyone can impersonate you (or anyone else)

INTEGRITY: TCP SPOOFING



if ack,seq generated using weak crypto, this is possible
(and both are only 4 bytes so can be brute-forced)

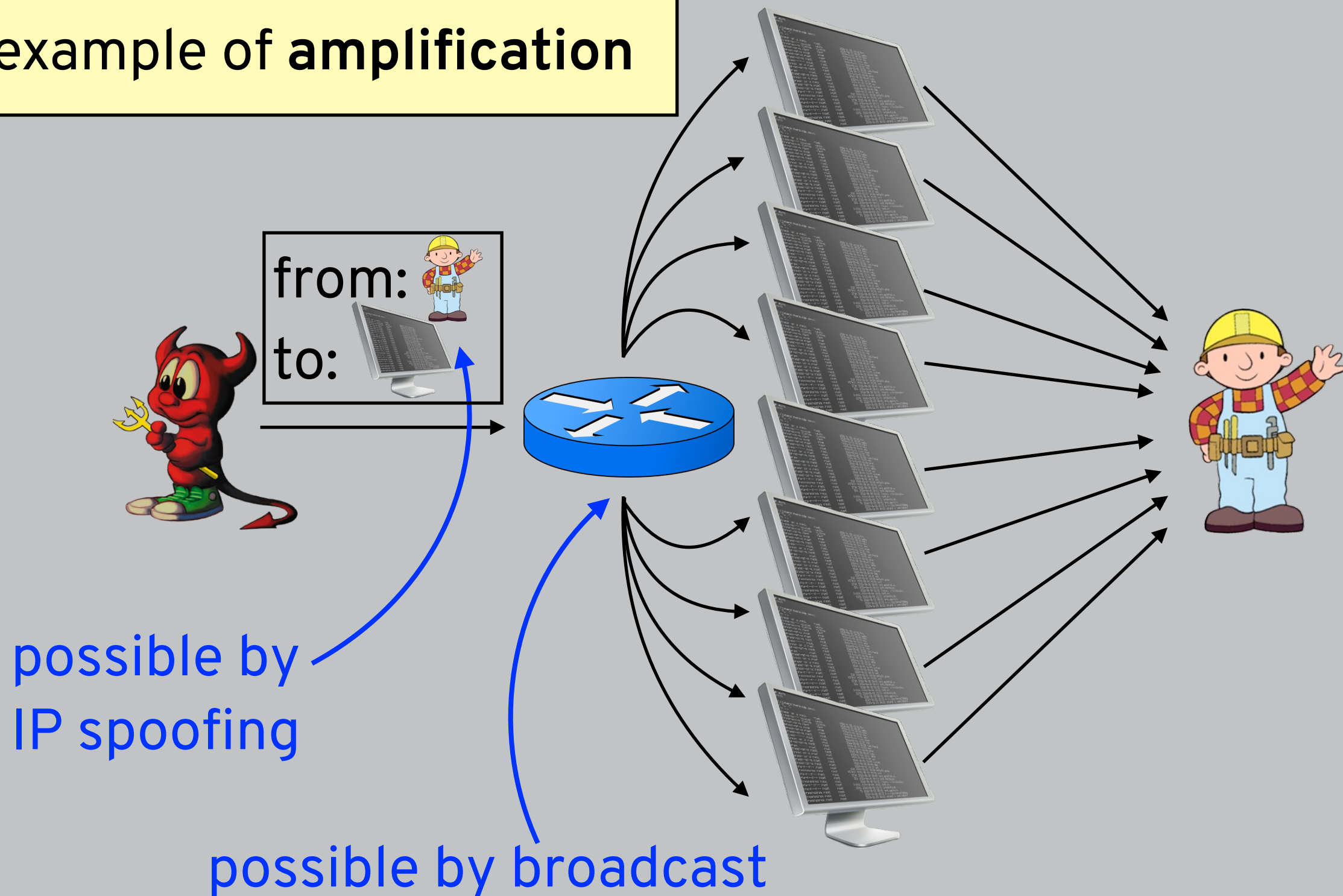
INTEGRITY: TCP HIJACKING



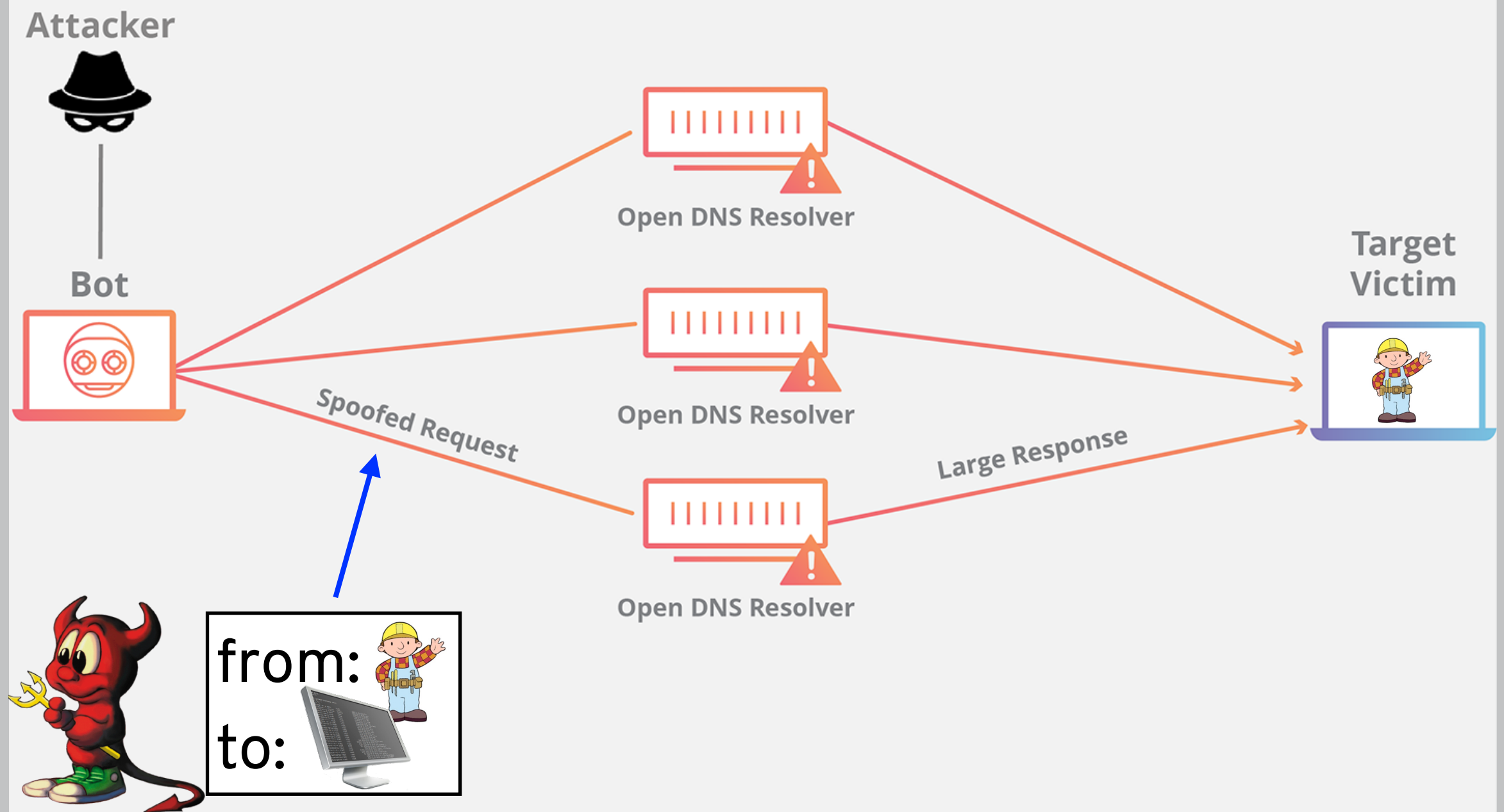
this is easy (just need MitM capabilities)

AVAILABILITY: SMURF ATTACK

example of amplification



AVAILABILITY: DNS AMPLIFICATION



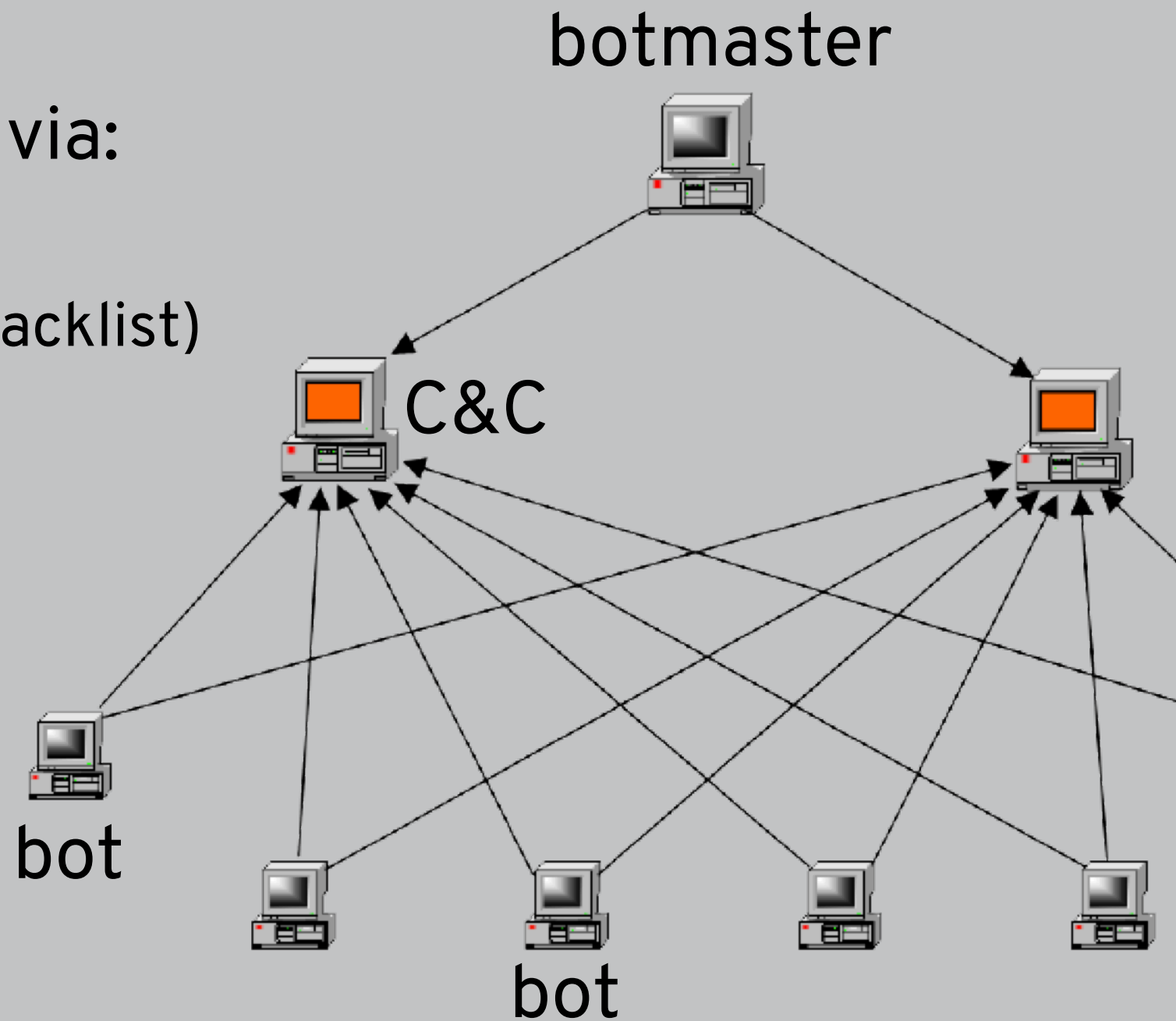
AVAILABILITY: BOTNETS

communication takes place via:

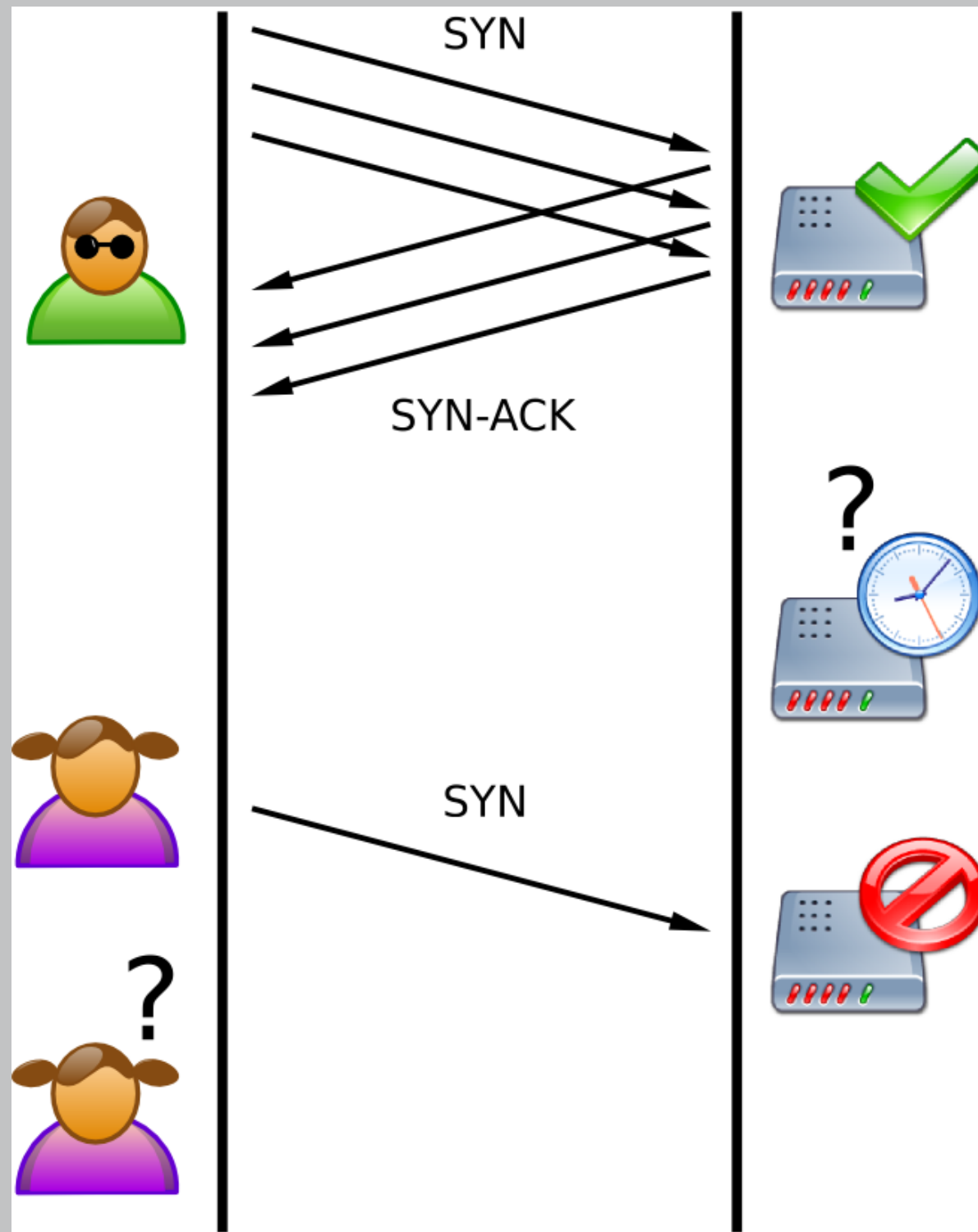
- IRC (easy to infiltrate)
- proprietary channels (easy to blacklist)

structure uses:

- multiple tiers (expensive)
- p2p (easy to infiltrate)
- fast flux/domain flux (hard!)



AVAILABILITY: SYN FLOOD



AVAILABILITY: SYN FLOOD

Several proposed countermeasures:

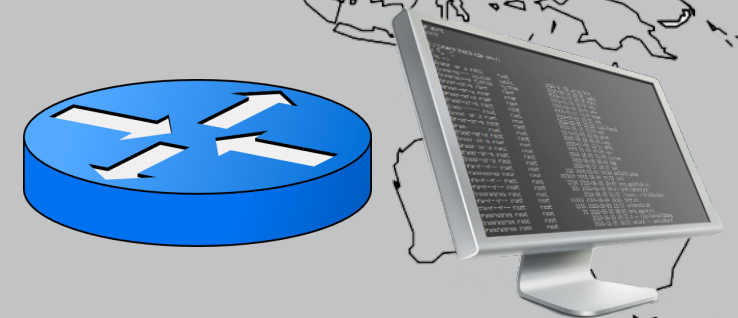
- Filtering (don't allow so many connections from same place)
- Reducing SYN-RECEIVED timer (don't wait so long to close connection)
- SYN cookies
- Firewalls and proxies

SYN cookies:

- Let t be slow timestamp (e.g., changes every minute)
- Let m be maximum segment size (MSS)
- Let $s = H(\text{IP addresses, ports, } t)$
- Initial seq ("SYN cookie") = 5 bits t + 3 bits m + 24 bits s
- This seq+1 is sent as ack in TCP ACK
- Server can check t within range, compute s and check equal

STEP 2: REQUEST CONTENT

`http://me.bob.com/hi.html`
`69.64.155.174`



`me.bob.com`