Module 4

Network Security and Privacy

Network basics

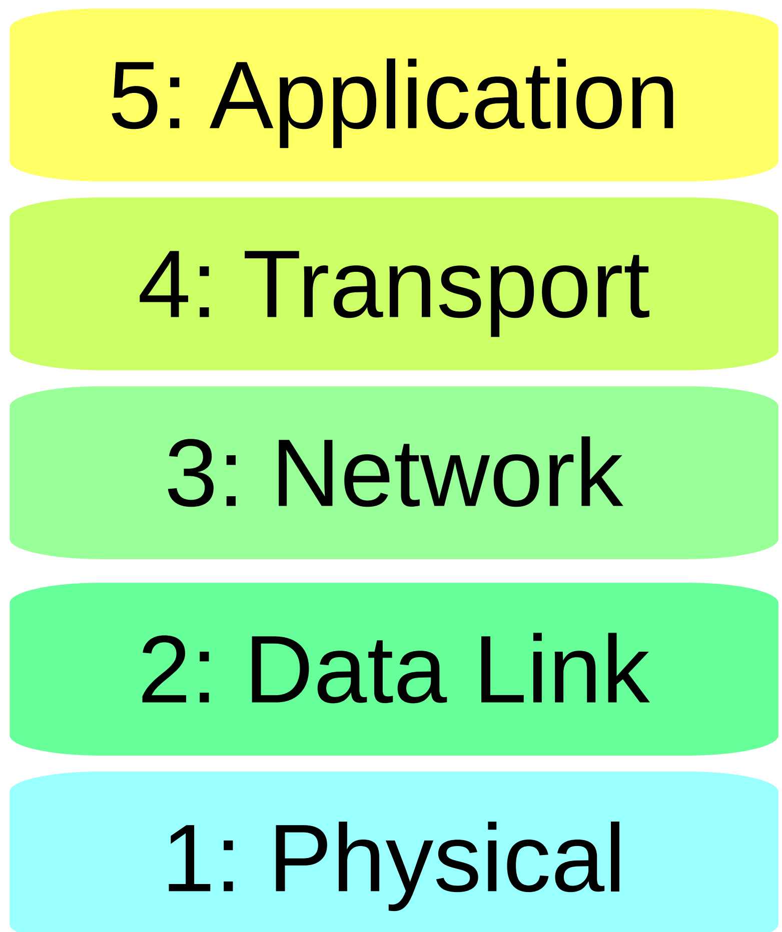
- Five-Layer Model:

  If Alice sends Bob (web server) a GET request:

  - It is an HTTP/HTTPS message (**Layer 5**)
  - It is wrapped in a TCP segment to deliver to the right port and guarantee order and correctness (**Layer 4**)
  - It is then wrapped in an IP packet to deliver to the right IP address (**Layer 3**)
  - It is then wrapped in a data link (e.g. Ethernet) frame for the MAC address of the closest router interface (**Layer 2**)
  - It is then sent onto the wire (**Layer 1**)

Network basics

- Five-Layer Model:



Security in the five-layer model

- Layer 5: TLS, Tor (this module), etc.
- Layer 4: TCP (poor security!)
- Layer 3: IP (poor security!)
- Layer 2: Ethernet (no assumed security)
- Layer 1: Physical (no assumed security)

Issues with TCP/IP Security

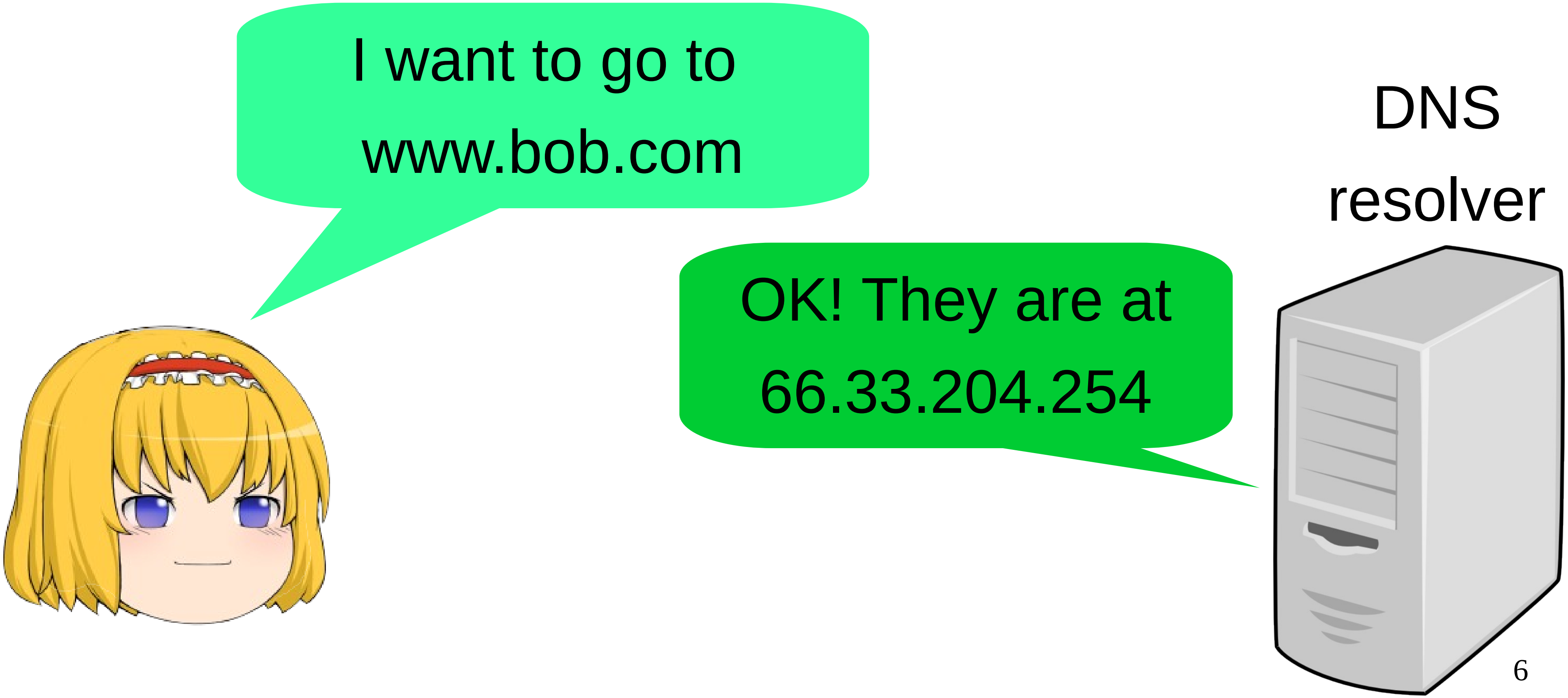
- No authentication of IP (send your packets to me, I am <web server>)
- No authentication of transmission path (send your packets to me, I will deliver them to <web server>)
- Various ways to achieve DoS (Denial of Service)
- Leveraging services for DDoS attacks
- (Others)

The early Internet was built on an assumption of honesty

Network basics

When you connect to a website:

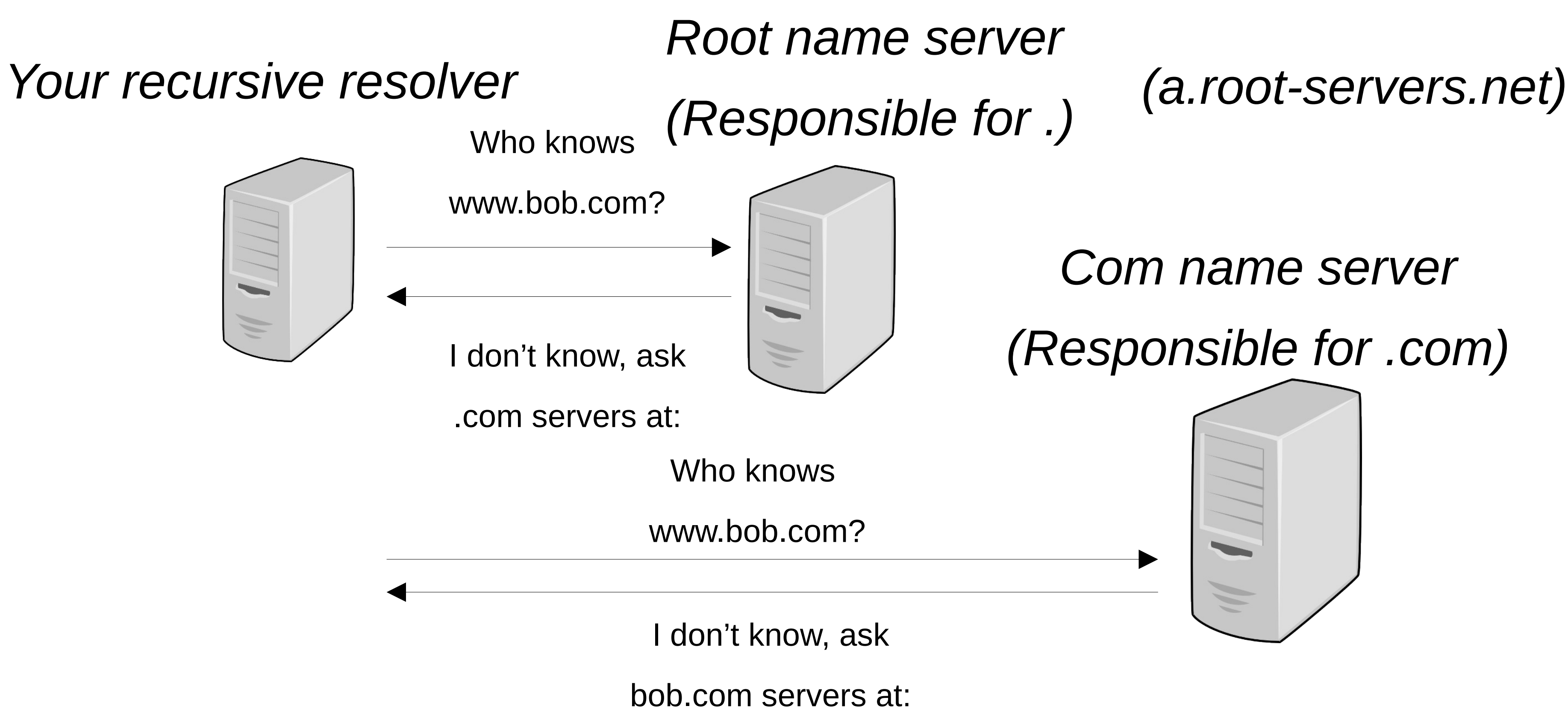
1. DNS resolution of **website name** into **IP address**
   (for humans) *(for machines)*



Network basics

How did the DNS resolver get the answer?

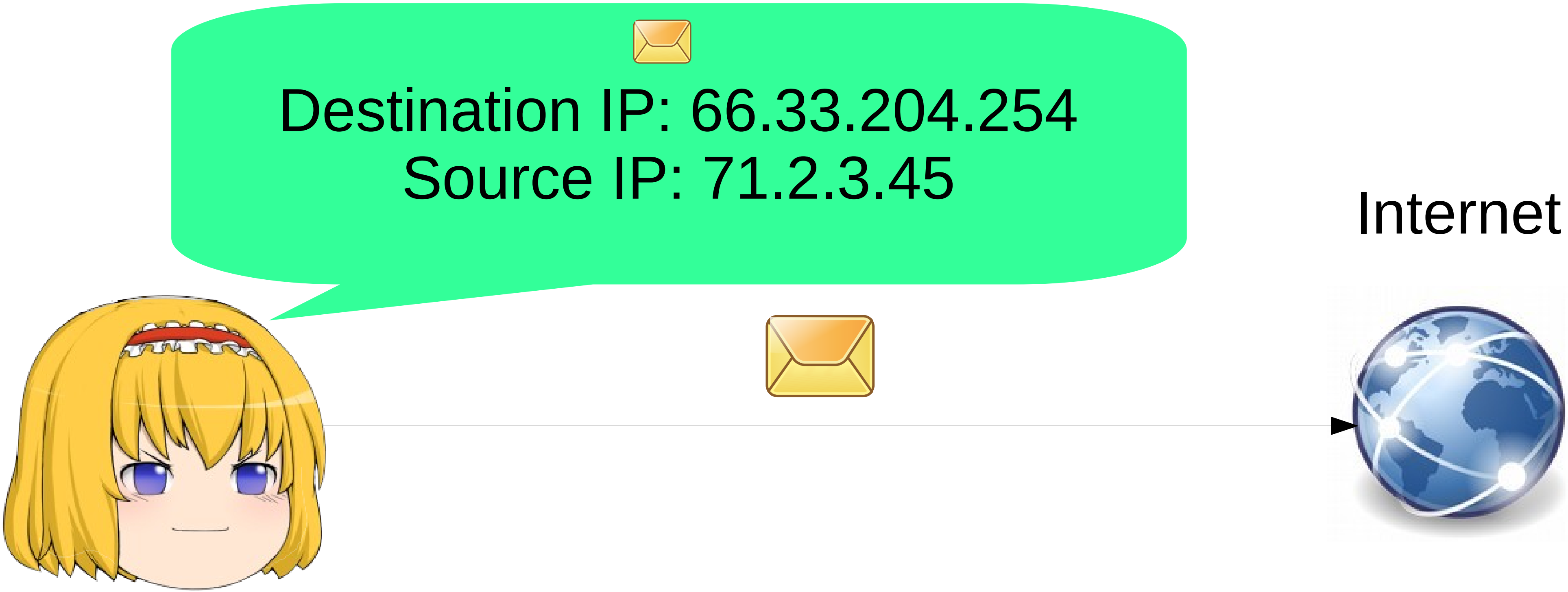
- **Recursive** resolution from root until an authority is found
- Each resolver gives an answer to the next part of the domain
- Answers are cached



Network basics

When you connect to a website:

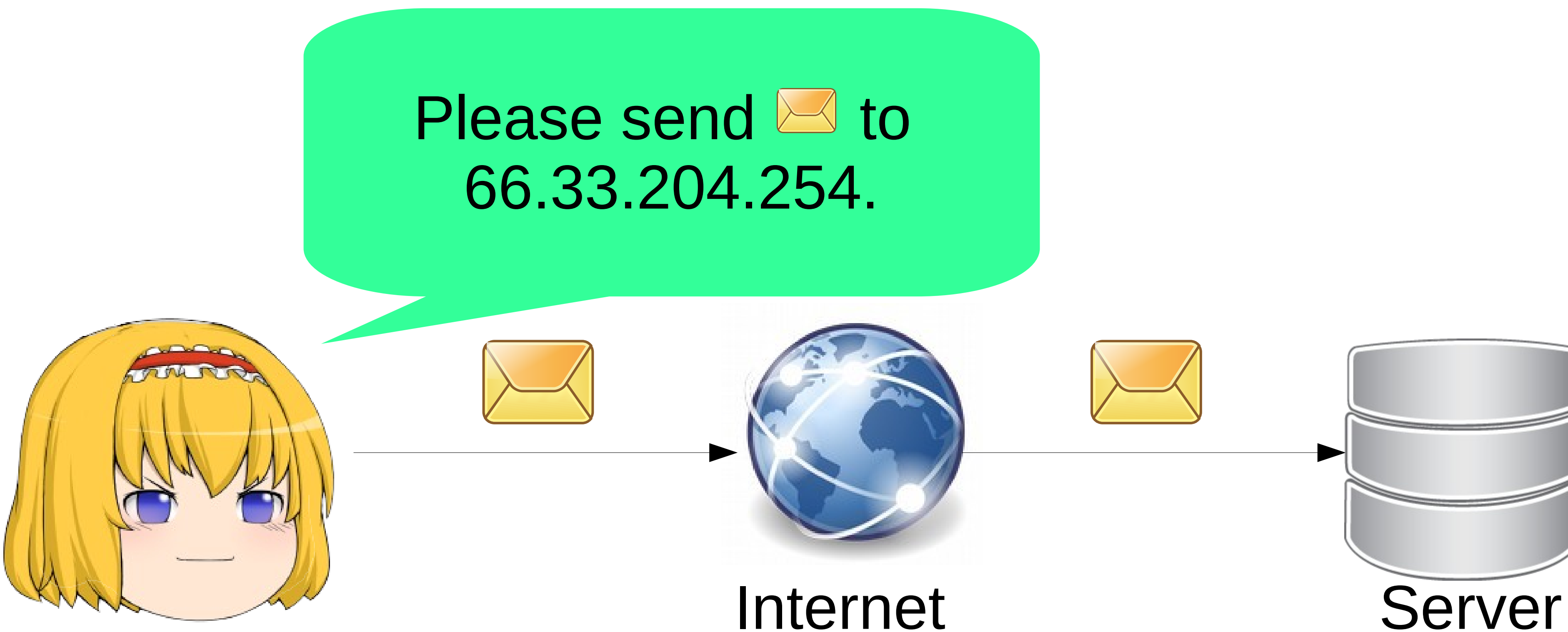
2. Create packet with your IP/port and website IP/port



Network basics

When you connect to a website:

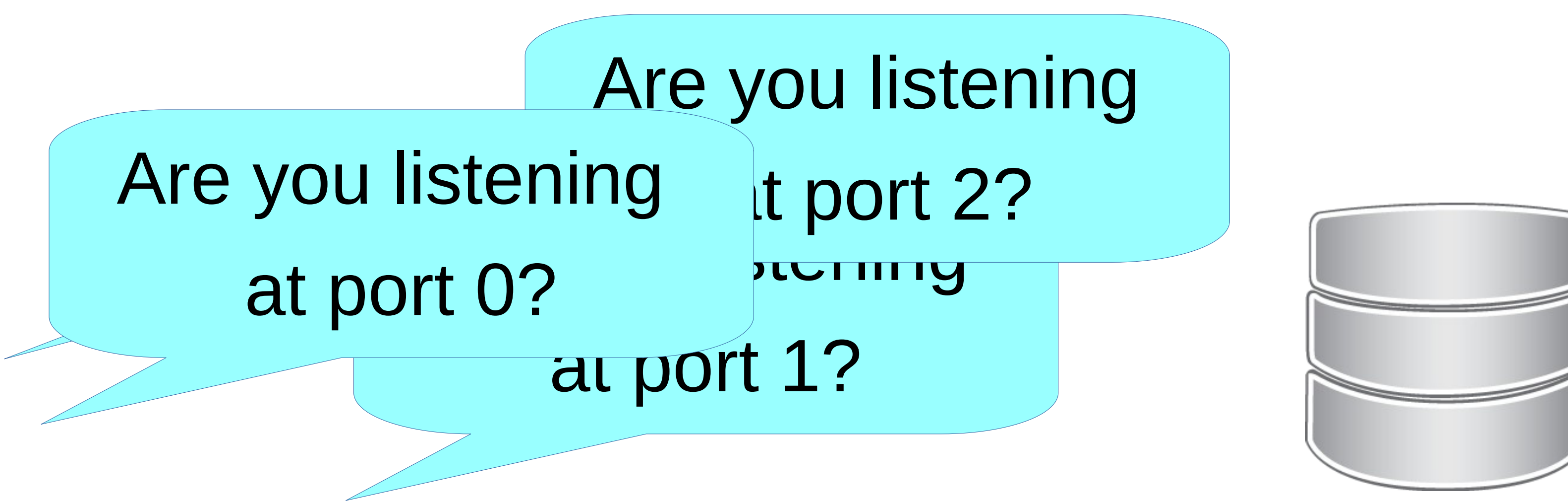
3. Packet is sent across the Internet towards the target server, carried by ISPs through TCP/IP



Discovering Victims

Port scanning

- Send packet to port, reply: open, closed, no reply
- Detect vulnerable network-facing programs
- Can scan the entire Internet in minutes
- Important component of pentests



Attacks

We will discuss two types of attacks:

- Impersonation Attacks (“Spoofing”)
  - Faking identity
  - Manipulates user trust of the Internet
- DoS Attacks
  - Take down services/targets
  - Redirection and Amplification

Impersonation Attacks

Fake host names:

- Register a web address similar to a real one
  - Typo: payapl.com
  - Visual: paypa1.com
  - Phonetic: paypel.com
  - Conceptual: paypalsecurity.com
- Link shorteners

Impersonation Attacks

<u>IP spoofing:</u>

- Write a fake address as your IP for your packets, pretending to be someone else
- Redirects return traffic to spoofed target
- Basis of many other attacks - unexpected behavior/denial of service

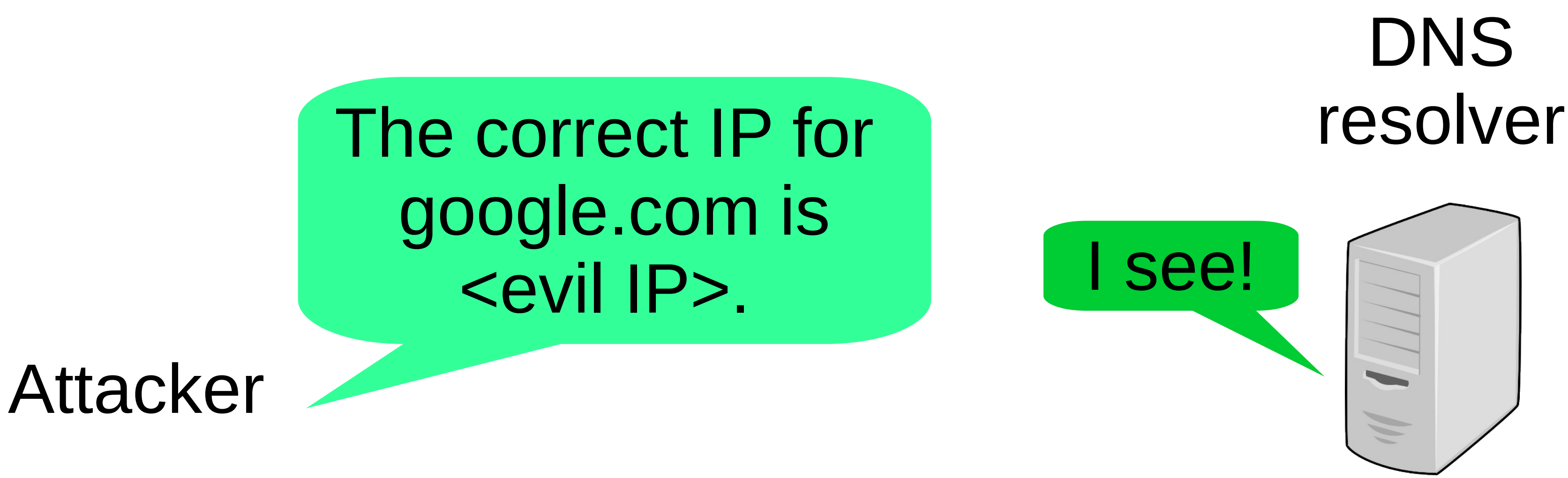
I am Alice!

I am attacking you!

Impersonation Attacks

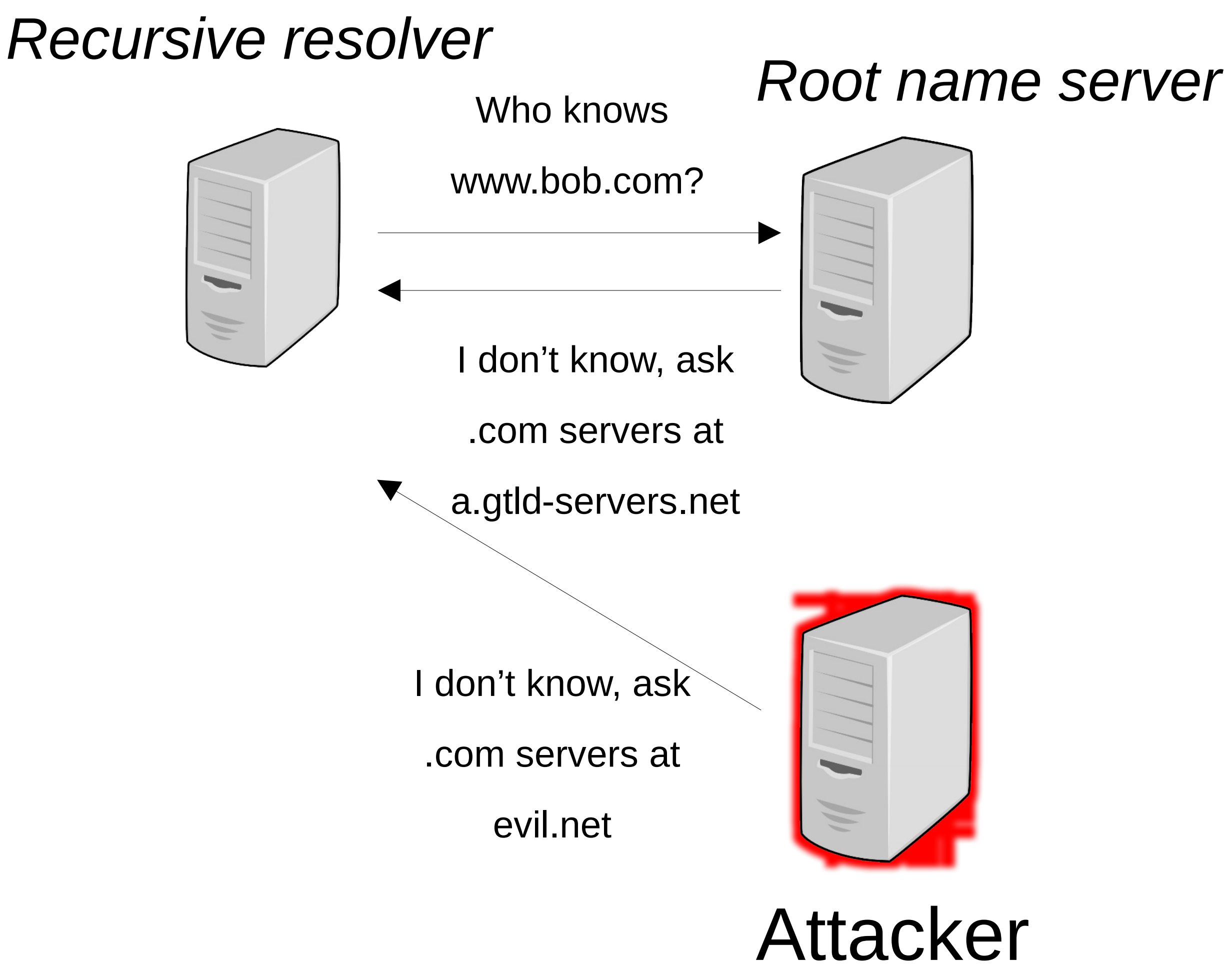
<u>DNS cache poisoning:</u>

- Makes a DNS server “remember” a wrong IP address for a human-readable address
- Redirects traffic to attacker's control
- Attacker can then compromise confidentiality, or feed fake files to the user



Impersonation Attacks

DNS cache poisoning:



Impersonation Attacks

DNS cache poisoning:

- Normally, recursive DNS servers ask *authoritative* servers for responses
- An attacker spoofs the IP of the authoritative server to reply, because there is no “proof” of an authoritative DNS server
- DNSSEC resolves this issue by using signature/verification schemes

Impersonation Attacks

Phishing:

- Tempts users (generally web-browsing or e-mail) to click a link or perform an action
- Spear-phishing: Highly targeted, personalized, uses data available on social media
  - Very hard to defend against! (e.g. cannot filter with firewall)

DoS Attacks

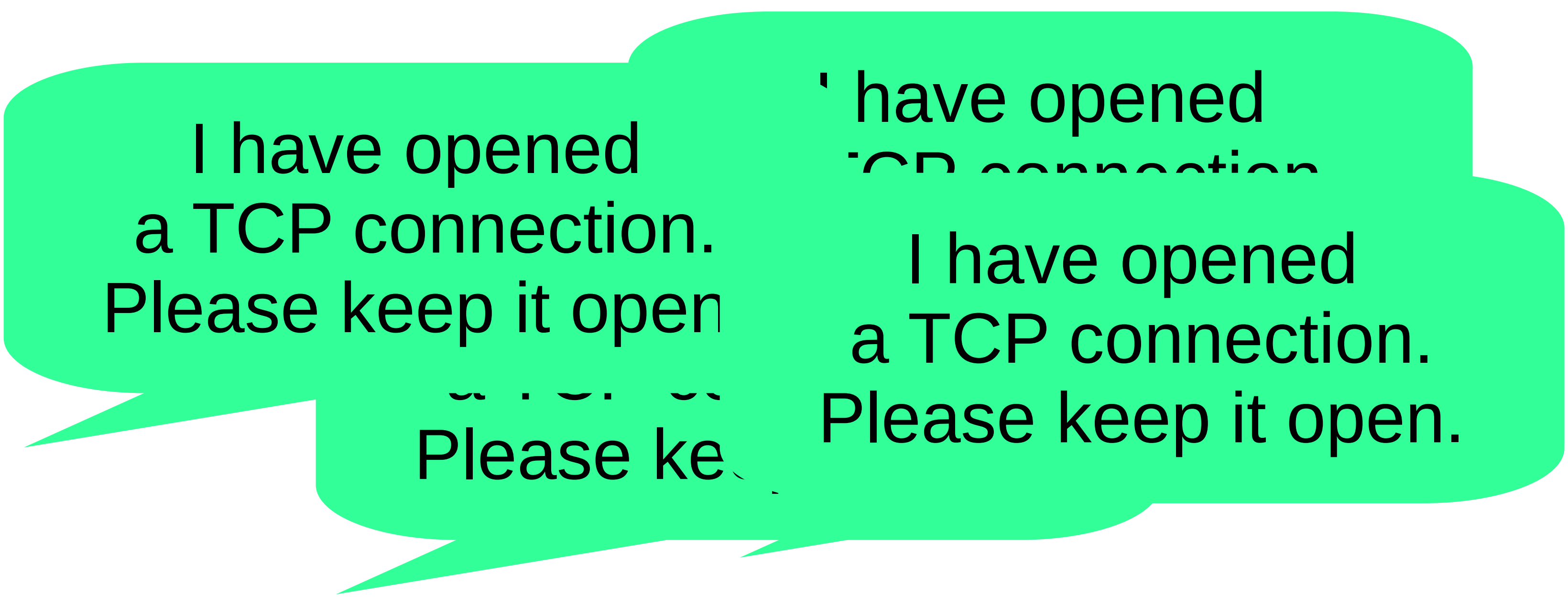
<u>Smurf attack:</u> (Amplification)

- Spoof victim's IP
- Send packet to network broadcast address
- Machines on the network will respond to this packet to the spoofed IP
- Nowadays: Packets to broadcast addresses are blocked

DoS Attacks

SYN flood:

- SYN is the start-up message of TCP
- Send many SYN messages, force server to open many connections
- Connections are memory-intensive



DoS Attacks

SYN flood:

- Solution: SYN cookies
- SYN cookies use a hash to enable completion of the TCP handshake without consuming any memory at the SYN-ACK stage

DoS Attacks

TCP/IP Fragmentation:

- Maximum IP datagram size is 65535 bytes, maximum ethernet frame size is usually 1500 bytes
- During transmission a single IP datagram can be fragmented into many packets
- Each IP datagram has an offset to aid in assembly
- Poor implementation of fragmentation caused vulnerabilities: ping of death, teardrop attack

DoS Attacks

Ping of death:

- Maximum offset field is a 13-bit number parsed as number of octets: 65528 bytes
- Maximum IP datagram size is 65535 bytes
- 65528 + 1500 > 65535
- If you implemented packets with an array:

```
> char packet_contents[65535];
```

You can now have a buffer overflow!

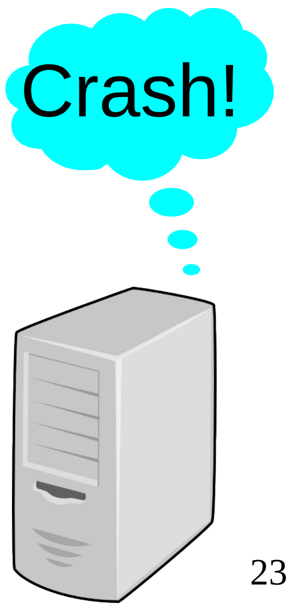
DoS Attacks

Teardrop attack:

- Since the attacker can set offset and contents herself, she can craft special IP fragments that have contradictions or problems

1) Send many tiny fragments to eat up CPU, never allowing reassembly

2) Send a fragment that would be completely contained in a previous fragment



Attacker

Bytes 1 to 6 are: HELLO!

Bytes 3 to 5 are: LLO

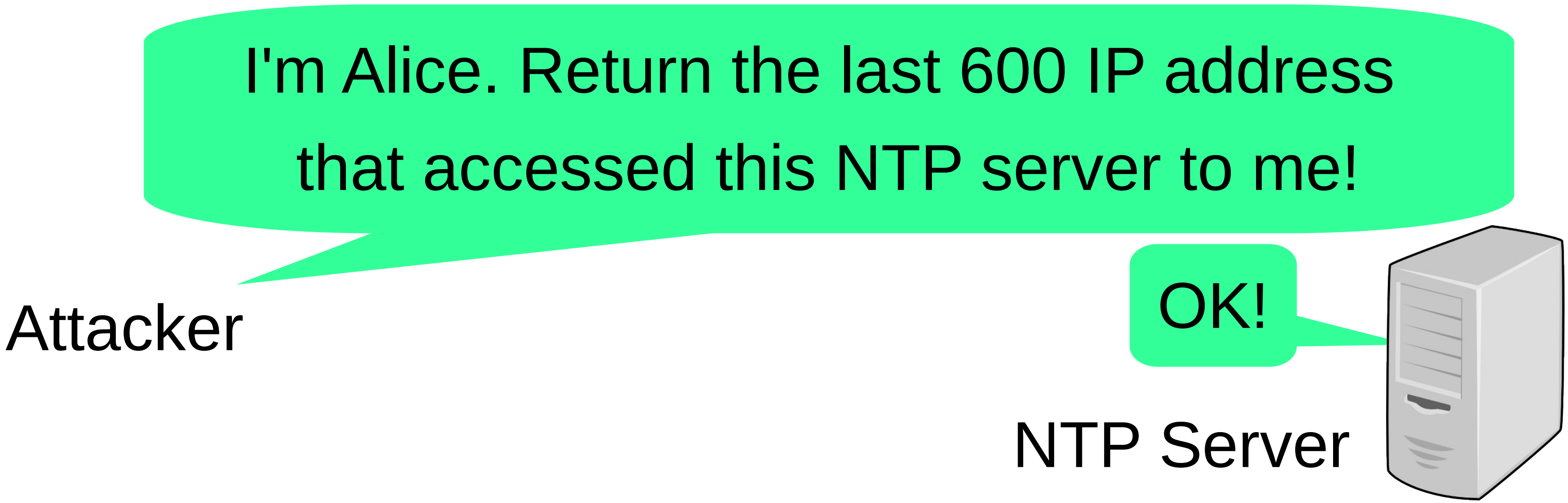
DoS Attacks

Distributed DoS:

- Control many machines, flood the victim
- Attacks are often short, hit critical moments
- Amplification:
  - Increase attack size
  - e.g. DNS response >> query size
- Reflection:
  - Redirect to target by spoofing target, then making query

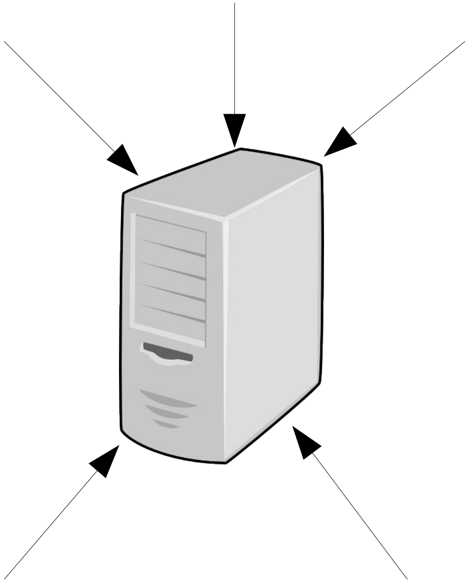
NTP amplification on CloudFlare

- NTP service allows online clock synchronization
- NTP allows 206-times amplification
- 400 Gbps amplified by 4500 NTP servers from possibly single server



DDoS on Dyn

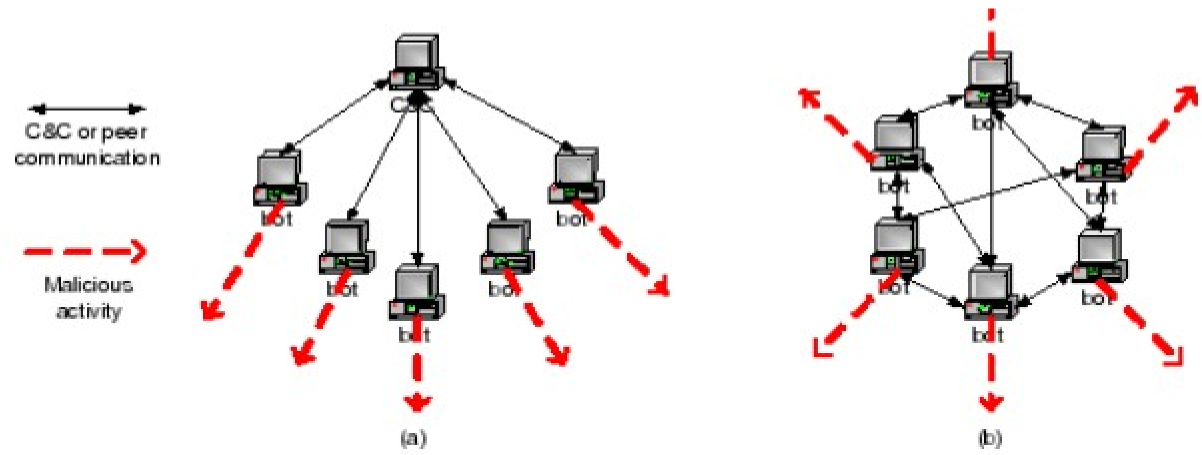
- Dyn was a DNS resolver
- 1200 Gbps requests to Dyn from massive IoT botnet called Mirai
- Many top sites taken down for a day
- Mirai simply logged in using factory-default passwords
- No amplification!



Botnets

- Consists of:
  - Many zombie computers (bots)

    Command and Control
  - A master
- Spam, DDoS, social media, bitcoin mining, loaders



Source: BotMiner (Gu et al.)

Botnets

- Command and Control systems:
  - Public channels (IRC, Twitter, etc.)
  - Peer-to-peer
  - Fast-flux: repeatedly registering different IP addresses for one domain at DNS
    - IP addresses are owned by bots
  - Domain Generation Algorithm: register many random domain names at DNS
    - Bots contact random domain names as well
    - e.g. Conficker (generates 50,000 domains and contacts 500 every day)
- Defended by: Sinkholing, usually with ISP cooperation or registrar cooperation

Botnet Takedown

- Usually involves cooperation of software company, law enforcement, and court
  - Company identifies domains to be taken down and gets court order to registrars
  - Physical takedown by seizing servers
  - Arresting individuals responsible
  - Takeover of botnet, send uninstall command

Defenses

Many attacks can be mitigated by packet filtering:

Packets can be filtered based on:
- headers (source, destination, size, etc.)
- contents (payload)

Firewall

Most computers have *personal firewalls*
But there are also *network-based firewalls*

Firewall

Firewall features:

- *Stateless packet filtering*: can block malicious IPs, attack code, “legitimate” but vulnerable protocols
- *Egress/ingress filtering*: Block IPs that don't make sense based on the network structure
- *Deep Packet Inspection*: Read packet content to decide what to block
- *Intrusion prevention*

Intrusion Detection System

- Detects intrusions, but does not block them
- Logs intrusions, raises alarm
- Two types, often used together:
  - Network-based
  - Host-based
- Subject to the base rate fallacy

Base rate fallacy

- True Positive Rate = Percentage of intrusions that raise alarms
- False Positive Rate = Percentage of non-intrusions that raise alarms
- Even if TPR ("accuracy") is very high, if the FPR is greater than the base rate of intrusions, then the majority of alarms are false (low precision)
- e.g. Boy who cries wolf (TPR = 100%, but low precision; eaten by wolf)

Intrusion Detection System

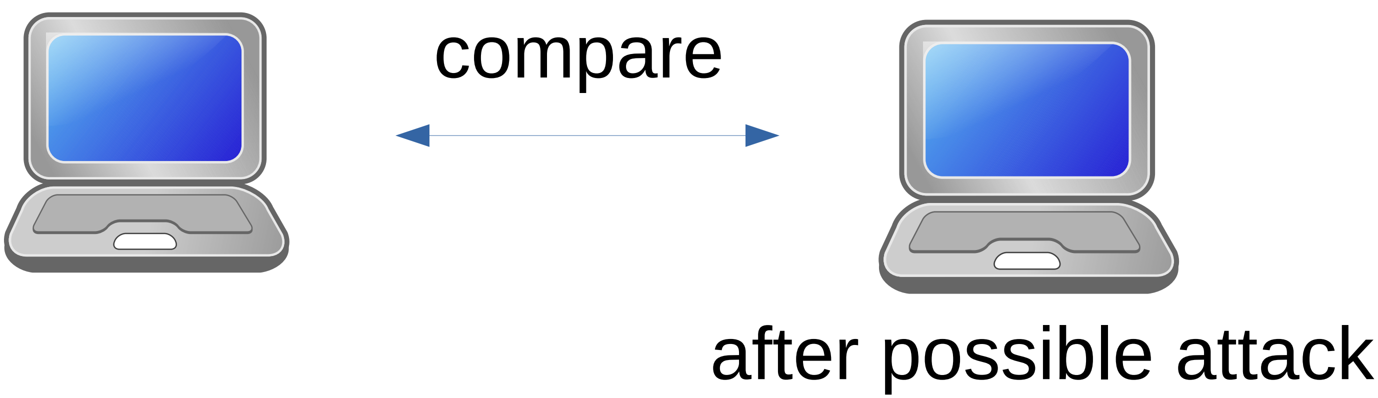
Network-based (NIDS):

- Detects malicious **packets**
- Monitors traffic after a firewall
- Behavior-based, or signature-based
  - Similar to malware scanning
- May be offline for efficiency reasons

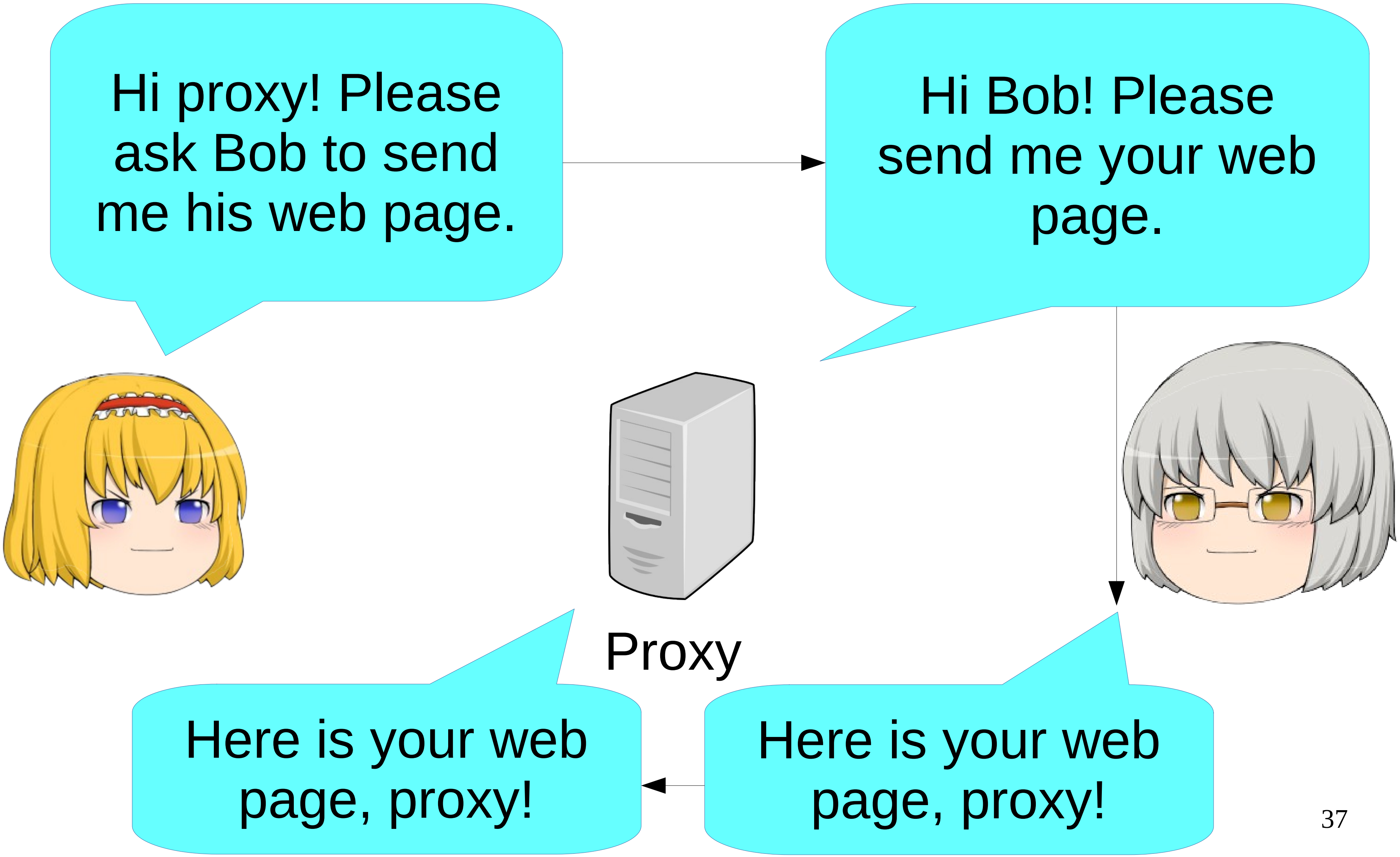
Intrusion Detection System

Host-based (HIDS):

- Detects malicious **system (filesystem)** changes
- Scans and saves clean system
- Later, scans new system after changes
- Detects malicious changes to critical aspects of system



Proxies (HTTP Request)



Proxies

Also useful for *anonymity*:

- Attackers can see source, destination IP (metadata)
- Sending packet from Alice to proxy:

  Source IP: <Alice>
  Destination IP: <Proxy>

- Sending packet from proxy to true destination:

  Source IP: <Proxy>
  Destination IP: <Destination>

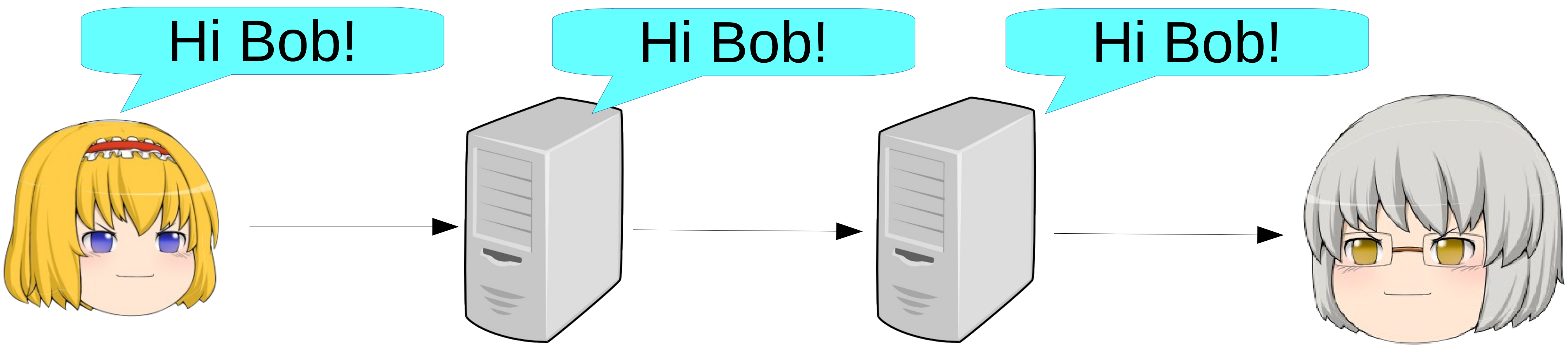
- Can also encrypt to improve confidentiality/integrity (but you have to trust the proxy)
- Application-specific

VPN

Similar to proxies, except:

- Always uses encryption
- Routes all network traffic
- May have many hops (not necessarily encrypted within the hops)

Anonymity Network



Proxy 1

(Knows Alice and Proxy 2)

Proxy 2

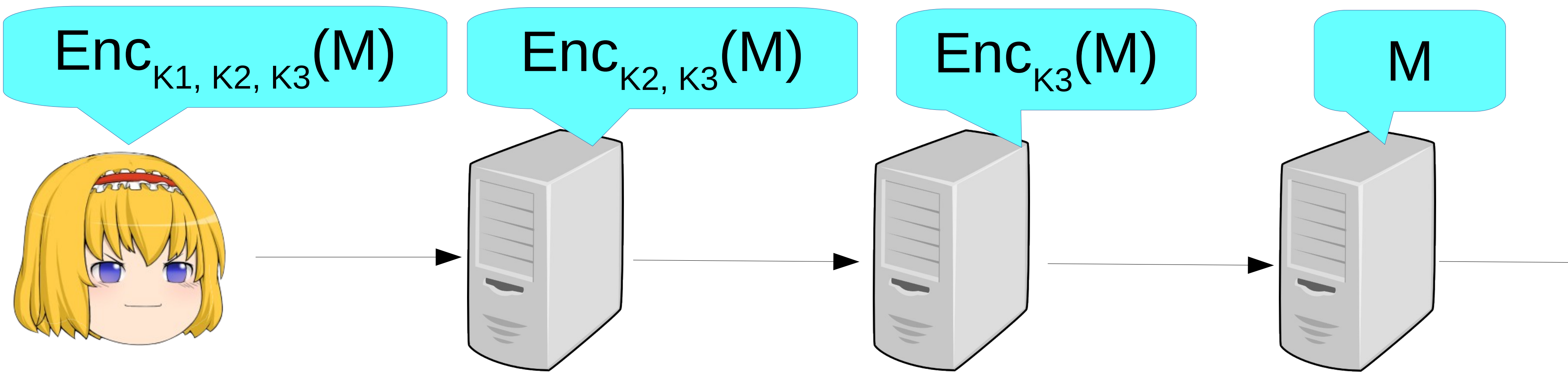
(Knows Proxy 1 and Bob)

Anonymity Network

A chain of proxies with these properties:

- Only the first proxy knows who you are.
- Only the final proxy knows your destination.
- Encryption is layered such that only the final proxy can read your packets.
- e.g. Remailer systems, Tor

Tor



(Knows K1, K2, K3)

Proxy 1
(Knows K1)
"Guard"

Proxy 2
(Knows K2)

Proxy 3
(Knows K3)
"Exit"

$$Enc_{K1, K2, K3}(M) = Enc_{K1}(Enc_{K2}(Enc_{K3}(M)))$$

Onion Routing Circuit

Tor

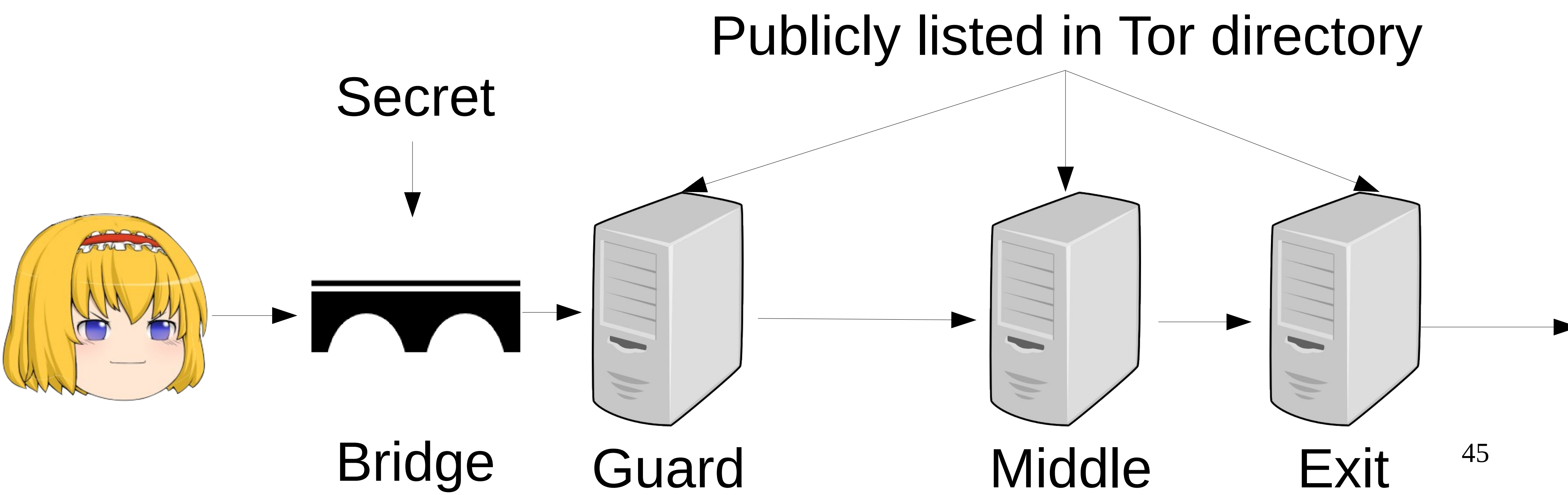
- Protects privacy, anonymity
- Usable directly with Tor Browser
- Some sites block it while others adjust for it
- Relies on volunteer nodes, which may misbehave
  - Peek into your packets
  - De-anonymization attacks

Tor Circuits

- Each circuit lasts for about 10 minutes
- Nodes cannot come from the same Autonomous System
- Exit nodes are the bottleneck in Tor's performance
- Users connect to public directory servers to obtain information about Tor nodes

Tor Bridges

- Tor is easily blocked
- For censorship resistance, Tor Bridges are used:
  - Secret instead of publicly known
  - Only carries traffic to Tor nodes



Tor

- Hidden services
  - Allows web servers and clients to communicate over Tor
  - Client accesses the server using a .onion address, never knows their real IP
  - Used to protect the server’s privacy
  - “Dark Web”, “Onionland”

Tor

- Hidden services
  - First, the hidden service establishes long-lasting circuits with **Introduction Points**. Introduction points are proxies chosen by the hidden service
    - Introduction Points are given to the directory servers
  - The user chooses another proxy as a **Rendezvous Point**
  - The user establishes a connection with an Introduction Point, and sends the Rendezvous Point through the Introduction Point to the hidden service
  - Both the user and the hidden service establish a circuit to the Rendezvous Point

https://community.torproject.org/onion-services/overview/

Tor attacks

- Controlling *f* proportion of nodes, the attacker has a f^2 chance of controlling each circuit
- To minimize time to first compromise, guards are long-term
- To increase this chance: Break the circuit
- Long-path attack:
  - Attacker is entry A, knows that next node is B
  - A builds such a circuit through B and DoSes B:
    - A -> B -> C -> B -> D -> B -> E -> ...

Tor attacks

- Browser fingerprinting: Server tracks client even across different connections
- Server asks client questions which have high-entropy answers:
  - Browser and OS version, timezone
  - Order of fonts installed
- Canvas fingerprinting: Server asks client to draw an image on a canvas, then retrieves the image

Tor attacks

- Website fingerprinting: Local attacker identifies what client is doing
- Different webpages have different packet traces
  - Classification
- Challenges: Huge open world, low base rate, training issues
- Much easier without Tor

DNS over TLS/HTTPS

- Implemented in some DNS servers
- Establishes encrypted channel between user and recursive resolver
  - Not the same as DNSSEC
- Defends against privacy threat
- DoT uses port 853, DoH uses port 443
- Some backlash from network administrators

QUIC

- Proposed replacement of TCP with UDP-like transport
- TCP + TLS handshake is 3 round trips
- QUIC + TLS handshake is 1 round trip
- Avoids TCP-based attacks
- Current: Some percentage of websites

Encrypted Client Hello

- Currently, the TLS handshake itself tells observers what website you are visiting
- ECH changes the handshake so that all handshakes to the same IP look the same
- Added privacy benefit to CDNs